Tenth MSU Conference on Differential Equations and Computational Simulations.
Electronic Journal of Differential Equations, Conference 23 (2016), pp. 139–154.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu
ftp ejde.math.txstate.edu

A GENERALIZATION OF PREFACTORED COMPACT SCHEMES FOR ADVECTION EQUATIONS

ADRIAN SESCU

Abstract. A generalized prefactorization of compact schemes aimed at reducing the stencil and improving the computational efficiency is proposed here in the framework of transport equations. By the prefactorization introduced here, the computational load associated with inverting multi-diagonal matrices is avoided, while the order of accuracy is preserved. The prefactorization can be applied to any centered compact difference scheme with arbitrary order of accuracy (results for compact schemes of up to sixteenth order of accuracy are included in the study). One notable restriction is that the proposed schemes can be applied in a predictor-corrector type marching scheme framework. Two test cases, associated with linear and nonlinear advection equations, respectively, are included to show the preservation of the order of accuracy and the increase of the computational efficiency of the prefactored compact schemes.

1. Introduction

Compact difference schemes, as opposed to explicit schemes, possess the advantage of attaining higher-order of accuracy with fewer grid points per stencil. They are preferred in applications where high accurate results are desired, such as direct numerical simulations, large eddy simulations, computational aeroacoustics or electromagnetism, to enumerate few, and in some instances they feature accuracy comparable to spectral methods. One of the disadvantages of compact schemes is that an implicit approach is required to determine the derivatives, where a matrix (usually multi-diagonal) has to be inverted.

A comprehensive study of high-order compact schemes approximating both first and second derivatives on a uniform grid was performed by Lele [15]. A wavenumber based optimization was introduced wherein the dispersion error was reduced significantly (assuming an exact temporal integration). Over the next years, compact schemes have been studied by many research groups, and applied to various engineering problems (see for example, Li et al. [16], Adams and Shariff [1], Liu [17], Deng and Maekawa [6], Fu and Ma [9, 10], Meitz and Fasel [21], Shen et al [25], Shah et al [24]). Other examples include Kim and Lee [14] who performed an analytic optimization of compact finite difference schemes, Mahesh [20] who derived a family of compact finite difference schemes for the spatial derivatives in the

2010 *Mathematics Subject Classification.* 65M06, 65M15, 35L02.
Key words and phrases. Compact difference scheme; Transport equation; discretization error.

Published March 21, 2016.

Navier-Stokes equations based on Hermite interpolations (see also, Chu and Fan [5] for a similar prior analysis), or Deng and Zhang [7] who developed compact high-order nonlinear schemes which are equivalent to fifth-order upwind biased explicit schemes in smooth regions. Hixon and Turkel [12, 13] derived prefactored high-order compact schemes that use three-point stencils and returns up to eighth-order of accuracy. These schemes combine the tridiagonal compact formulation with the optimized split derivative operators of an explicit MacCormack type scheme. The optimization of Hixon's [12, 13] schemes in terms of reducing the dispersion error was performed by Ashcroft and Zhang [2] who used Fourier analysis to select the coefficients of the biased operators such that the dispersion characteristics match those of the original centered compact scheme and their numerical wavenumbers have equal and opposite imaginary components. Sengupta et al [23] derived a new compact schemes for parallel computing. Today, compact schemes are widely used in numerical simulations of turbulent flows (e.g., direct numerical simulations), computational aeroacoustics, or computational electromagnetics. In order to increase the speed of such numerical simulations it is desirable to derive more computational efficient compact schemes without affecting the order of accuracy and the wavenumber characteristics.

In this work, we propose a generalized prefactorization of existing compact schemes aimed at reducing the stencil and increasing the computational efficiency. It is based on the type of prefactorization introduced previously by Hixon and Turkel [12, 13], but here the order of accuracy can be increased indefinitely, and there are no specific requirements for the original compact schemes to be suitable to prefactorization, other than they must fall in the class of centered scheme. A similar optimization was recently performed by Bose and Sengupta [3]. They developed an alternate direction bidiagonal (ADB) scheme, which showed neutral stability and good dispersion characteristics. The analysis included here can be viewed as a generalization of their work, except the focus is on the order of accuracy rather the dispersion characteristics. One of the restrictions of the schemes derived here is that the proposed prefactored schemes can be combined with a predictor-corrector type time marching scheme only. This allows the determination of the derivatives by sweeping from one boundary to the other, thus avoiding the inversion of matrices which can make the computational algorithm cumbersome and the execution time-consuming. It is shown that the original order of accuracy of the classical compact schemes is preserved, while the computational efficiency can be almost doubled (numerical tests pertaining to fourth and sixth order accurate schemes show over 40% decrease in the processing time, while higher order schemes are expected to be more efficient).

In Section 2, we discuss the classical compact difference schemes including wavenumber characteristics. In section 3, we introduce and analyze the prefactored compact schemes. In section 4, we consider two test cases to verify the efficiency and to check the preservation of the order of accuracy of the proposed schemes. The two text cases correspond to a linear and a nonlinear problem. In the last section we have some concluding remarks.

2. Compact difference schemes

Consider the general compact centered difference approximation for the first derivative:

$$\sum_{k=1}^{N_c} \alpha_k (u'_{j+k} + u'_{j-k}) + u'_j = \frac{1}{h} \sum_{k=1}^{N_e} a_k (u_{j+k} - u_{j-k}) + O(h^n), \tag{2.1}$$

where $1 \le j \le N$ (with N being the number of grid points), the gridfunctions at the nodes are $u_j = u(x_j)$, the values of the derivatives with respect to x are u'_j, and h is the spatial step. If $\alpha_k = 0$ then the scheme is termed explicit. Compact schemes (also known as implicit or Pade schemes), by contrast, have $\alpha_k \neq 0$ and require the solution of a matrix equation to determine the derivatives of the grid function. Conventionally, the coefficients α_k and a_k are chosen to give the largest possible exponent, n, in the truncation error, for a given stencil width. Table 1 gives several weights for various compact difference schemes that are considered in this study: fourth (C4), sixth (C6), eight (C8), tenth (C10), twelfth (C12), fourteenth (C14) and sixteenth (C16) order accurate. Schemes C4 and C6 require a tri-diagonal matrix inversion, C8 and C10 a penta-diagonal matrix inversion, C12 and C14 a seven-diagonal matrix inversion, while C16 a nine-diagonal matrix inversion. The inversion of three- and five-diagonal matrices for determining the weights of compact differencing schemes received increased attention, while seven- and nine-diagonal matrices are less popular due to the computational inefficiency. The prefactored compact scheme of Hixon [12] is also included here in the form

$$\begin{aligned} a u^{F'}_{j+1} + c u^{F'}_{j-1} + (1-a-c) u^{F'}_j &= \frac{1}{h}\left[b u_{j+1} - (2b-1) u_j - (1-b) u_{j-1} \right], \\ c u^{B'}_{j+1} + a u^{B'}_{j-1} + (1-a-c) u^{B'}_j &= \frac{1}{h}\left[(1-b) u_{j+1} - (2b-1) u_j - b u_{j-1} \right], \end{aligned} \tag{2.2}$$

where F and B stand for 'forward' and 'backward', respectively. For sixth order accuracy, $a = 1/2 - 1/(2\sqrt{5})$, $b = 1 - 1/(30a)$ and $c = 0$.

Table 1. Weights of several compact difference schemes

Scheme	α_1	α_2	α_3	α_4	a_1	a_2	a_3	a_4
$C4$	1/4	0	0	0	3/4	0	0	0
$C6$	1/3	0	0	0	7/9	1/36	0	0
$C8$	4/9	1/36	0	0	20/27	25/216	0	0
$C10$	1/2	1/20	0	0	17/24	101/600	1/600	0
$C12$	9/16	9/100	1/400	0	21/32	231/1000	49/4000	0
$C14$	3/5	3/25	1/175	0	31/50	67/250	283/12250	1/9800
$C16$	16/25	4/25	16/1225	1/4900	72/125	38/125	1784/42875	761/686000

The leading order term in the truncation error of a finite difference scheme depends on the choice of the coefficients and the $(n+1)$st derivative of the function u. To obtain the wavenumber characteristics of compact schemes, consider a periodic domain with N uniformly spaced points on $x \in [0, L]$ (with $h = L/N$), and the discrete Fourier transform of u as

$$u_j = \sum_{m=-N/2}^{N/2-1} \hat{u}_m e^{i k_m x_j}; \quad j = 1, \dots, N. \tag{2.3}$$

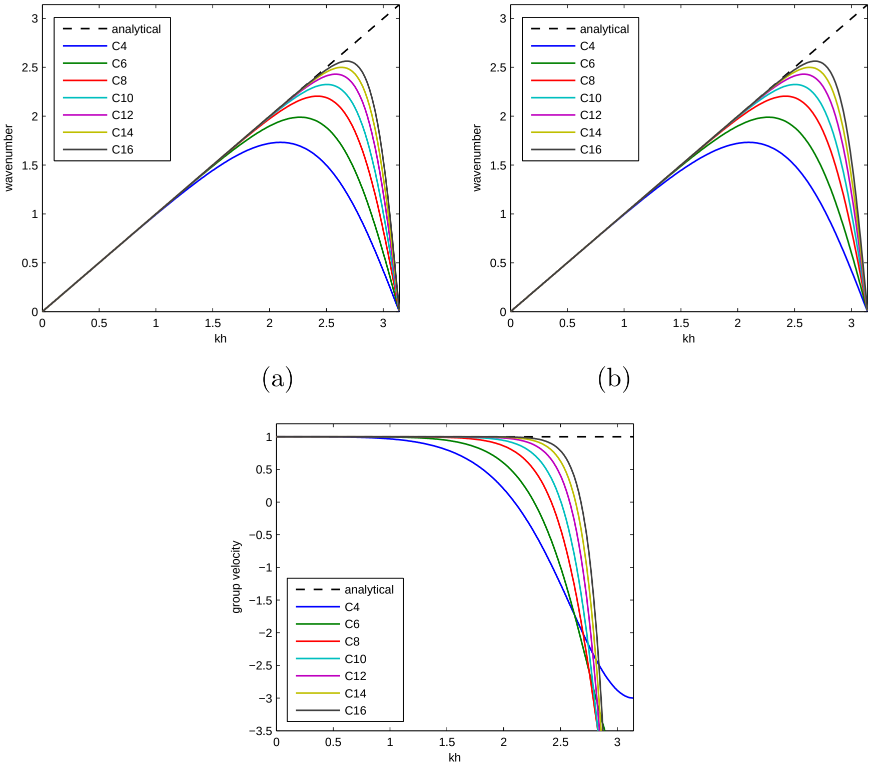
where the wavenumber is $k_m = 2\pi m/L$. The mth component of the discrete Fourier transform of u' denoted $\hat{u}'_m$ is simply $i k_m \hat{u}_m$. Taking the discrete Fourier transform of equation (2.1) provides the approximate value of $\hat{u}'_m$ in the form

$$(\hat{u}'_m)_{num} = iK(k_m h)\hat{u}_m, \tag{2.4}$$

where the numerical wavenumber is given by:

$$K(z) = \frac{\sum_{n=1}^{N_e} 2a_n \sin(nz)}{1 + \sum_{n=1}^{N_c} 2\alpha_n \cos(nz)}. \tag{2.5}$$

where z is the wavenumber. Figure 1 shows the numerical wavenumber, the phase velocity, and the group velocity corresponding to the schemes given in table 1 (the exact wavenumber, phase, and group velocity are also included for comparison). One can notice that the dispersion error decreases as the order of accuracy is increased.



(a) (b)

(c)

Figure 1. (a) Numerical wavenumber for various compact schemes compared to the exact wavenumber. (b) Numerical phase velocity compared to the exact phase velocity. (c) Numerical group velocity compared to the exact group velocity.

3. Generalized profactored schemes

The classical compact difference scheme (2.1) is written (conveniently) in the following form:

$$\sum_{k=1}^{N_c}\gamma_k(u'_{j+k}+u'_{j-k})+\Big(1-2\sum_{k=1}^{N_c}\gamma_k\Big)u'_j=\frac{1}{h}\sum_{k=1}^{N_e}\eta_k(u_{j+k}-u_{j-k})+O(h^n), \quad (3.1)$$

where the coefficients γ_k and η_k are functions of the original coefficients α_k and a_k given in table 1. The prefactored compact schemes proposed in this work are given in the form

$$\Big(1-\sum_{k=1}^{N_c}\beta_k\Big){u'_j}^F+\sum_{k=1}^{N_c}\beta_k {u'_{j+k}}^F=\frac{1}{h}\sum_{k=1}^{N_e}b_k\Big(u_{j+k}-u_j\Big)+O(h^n), \quad (3.2)$$

for the forward operator, and

$$\Big(1-\sum_{k=1}^{N_c}\beta_k\Big){u'_j}^B+\sum_{k=1}^{N_c}\beta_k {u'_{j-k}}^B=\frac{1}{h}\sum_{k=1}^{N_e}b_k\left(u_j-u_{j-k}\right)+O(h^n), \quad (3.3)$$

for the backward operator ('forward' and 'backward' correspond to the predictor and corrector steps, respectively). Notice that the two new schemes are of downwind and upwind types, respectively, so they may feature dissipation errors. However, when combining the predictor and corrector operators the dissipation errors equate to zero, while the dispersion error is the same as the one corresponding to the original centered compact scheme. If we consider a spatial discretization on a one-dimensional grid (consisting of N grid points), the schemes (3.2) and (3.3) can be written in matricial form (Hixon and Turkel [12, 13]) as

$$B^F_{mn}{U'_n}^F=\frac{1}{h}C^F_{mn}U_n, \quad (3.4)$$

$$B^B_{mn}{U'_n}^B=\frac{1}{h}C^B_{mn}U_n \quad (3.5)$$

where $\{U'\}$ is the vector of derivatives, $\{U\}$ is the vector of grid functions, and $m,n\in\{1,2,\dots,N\}$. Equations (3.2) and (3.3) imply

$$B^F_{mn}=B^B_{nm}, \quad (3.6)$$

$$C^F_{mn}=-C^B_{nm} \quad (3.7)$$

The coefficients in (3.2) and (3.3) are determined by requiring that the original classical compact scheme is recovered by performing an average between the predictor and corrector operators, formally written as

$$\langle\cdot\rangle'=\frac{1}{2}\Big(\langle\cdot\rangle'^F+\langle\cdot\rangle'^B\Big) \quad (3.8)$$

Multiplying (3.8) by $B^B_{mn}B^B_{nm}$ and using (3.4), (3.5), (3.6) and (3.7), as well as the relation $B^B_{mn}B^B_{nm}=B^B_{nm}B^B_{mn}$, which is true for matrices of the type considered here, we obtain

$$B^B_{mn}B^B_{nm}\langle\cdot\rangle'=\frac{1}{2h}\left(B^B_{nm}C^B_{mn}-B^B_{mn}C^B_{nm}\right)\langle\cdot\rangle \quad (3.9)$$

The coefficients β_k and b_k can now be determined by matching equation (3.9) with equation (3.1) (in the appendix, a Mathematica [28] code used to determine the coefficients for the new 12th order accurate scheme is included). Tables 2 and 3 include these coefficients for several schemes of different orders of accuracy, fourth (PC4), sixth (PC6), eigth (PC8), tenth (PC10), twelfth (PC12), fourteenth (PC14) and sixteenth (PC16), that correspond to the classical compact schemes given in table 1.

Table 2. Weights of several prefactored compact difference schemes (left-hand side).

Scheme	β_1	β_2	β_3	β_4
$PC4$	0.211324870586	0	0	0
$PC6$	0.276393202250	0	0	0
$PC8$	0.353614989057	0.022913166676	0	0
$PC10$	0.390891054882	0.041982762456	0	0
$PC12$	0.424261339307	0.076528671307	0.002177424900	0
$PC14$	0.440844836186	0.103628733678	0.005175974177	0
$PC16$	0.450833811211	0.139274137394	0.012291382216	0.000195518547

Table 3. Weights of several prefactored compact difference schemes (right-hand-side).

Scheme	b_1	b_2	b_3	b_4
$PC4$	1.000000000000	0	0	0
$PC6$	0.907868932583	0.046065533708	0	0
$PC8$	0.679849926548	0.160075036725	0	0
$PC10$	0.544199349631	0.223702048938	0.002798850830	0
$PC12$	0.377436479527	0.283739040458	0.018361813185	0
$PC14$	0.270368050633	0.312589794656	0.034570978390	0.000184856220
$PC16$	0.157403729700	0.326389389050	0.060796869771	0.001856720721

The stencil count has been reduced as follows: from 3- to 2-point stencil for PC4 (this is similar to the scheme proposed by Hixon [12]), from 5- to 3-point stencil for PC6 and PC8, from 7- to 4-point stencil for PC10 and PC12, and from 9- to 5-point stencil for PC14 and PC16. The advantage of these prefactored schemes is that there is no need to invert matrices because the derivatives can be obtained explicitly by sweeping from one boundary to the other (assuming the grid functions and the derivatives are available at the boundaries). This simplifies the computational cost significantly, without affecting the performance of the schemes since by averaging the predictor and corrector operators the original classical compact schemes are obtained. Thus, the wavenumber characteristics manifested through zero-dissipation and low dispersion of the original compact scheme (2.1) is retained.

4. Test cases

4.1. **Preliminaries.** We consider the initial-value problem in $\mathbb{R} \times [0, \infty)$:

$$\frac{\partial u}{\partial t} + c(u)\frac{\partial u}{\partial x} = 0, \tag{4.1}$$

$$u(x, 0) = u_0(x), \tag{4.2}$$

and appropriate boundary conditions, where $u(x, t)$ is a scalar function, c is the convective velocity which may depend on u, and $u_0(x)$ is a given function of space.

Let $\Omega = \{x, l_1 < x < l_2\}$ be a finite domain in the real set $\mathbb{R}$ with l_1 and l_2 chosen such that there exist a real non-negative number $h = (l_1 - l_2)/N$ called spatial step (N is an integer representing the number of grid points).

Because the accuracy of the time marching scheme is not the focus of this study, we use a second order MacCormack [19] scheme which is a two-step predictor-corrector time advancement scheme, and a second order TVD Runge-Kutta scheme [18]. The first time marching schemes is applied in the framework of prefactored compact schemes, while the second scheme is applied in the framework of classical compact schemes. To increase the accuracy of the time marching scheme, a very small time step is set, the emphasis being on the accuracy of the spatial discretization. For the advection equation in one dimension, the MacCormack scheme can be written as

$$\begin{aligned} u_j^F &= u_j^n - \sigma \Delta_x^F u_j^n \\ u_j^B &= u_j^F - \sigma \Delta_x^B u_j^F \\ u_j^{n+1} &= \frac{1}{2}(u_j^F + u_j^B), \end{aligned} \tag{4.3}$$

where $\sigma = c(u)k/h$, k is the time step, Δ_x^F is the 'forward', and Δ_x^B is the 'backward' difference operators. The second order TVD Runge-Kutta method [18] is

$$\begin{aligned} u_j^{(0)} &= u_j^n \\ u_j^{(1)} &= u_j^{(0)} + \Delta t L(u_j^{(0)}) \\ u_j^{n+1} &= \frac{1}{2}u_j^{(0)} + \frac{1}{2}u_j^{(1)} + \frac{1}{2}\Delta t L(u_j^{(2)}) \end{aligned} \tag{4.4}$$

where $L(u_j) = -c(u_j)(\partial u/\partial x)_j$. These two time marching schemes are essentially the same, except the first is applied in the framework of predictor-corrector type marching procedure, while the second is applied with 2 stages per time step.

4.2. **Linear advection equation.** For the linear advection equation, the convective velocity c is a positive constant (equal to 1 here), which renders the transport of the initial condition $u_0(x)$ in the positive direction. The initial condition is

$$u_0(x) = u(x,0) = \frac{1}{2}\exp\big[-(\ln 2)\frac{x^2}{9}\big] \tag{4.5}$$

(Hardin et al. [11]). The domain boundaries are $l_1 = -20$ and $l_2 = 450$, and the final time is $t_f = 200$. The equation (4.1) is solved numerically using all prefactored compact difference schemes considered in the previous section. Figure 2 shows the numerical solution using the sixth order accurate prefactored and classical compact schemes; the numerical solutions are compared to the analytical solution (the initial solution is also included). One can notice that there is no difference among the solutions. This is stressed in table 4 which lists the L_2-errors for the fourth, sixth, eight and tenth order accurate schemes (both prefactored and classical); the differences between the errors are very small, which is expected since the schemes have similar behavior (the differences in the errors come from the time marching algorithms).

From Taylor series expansions corresponding to two different steps, h_1 and h_2, the order of accuracy of the compact scheme can be estimated as

$$p = \frac{\ln(\epsilon_1/\epsilon_2)}{\ln(h_1/h_2)} \tag{4.6}$$

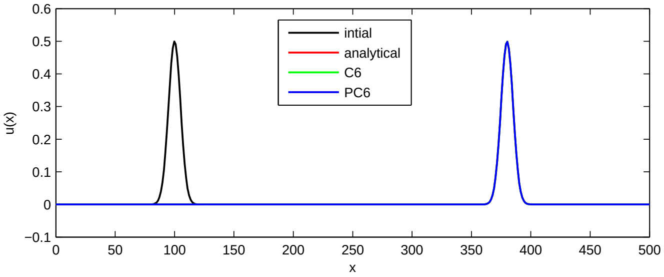


Figure 2. Plot of the linear solution at $t = 280$ for the sixth order accurate schemes.

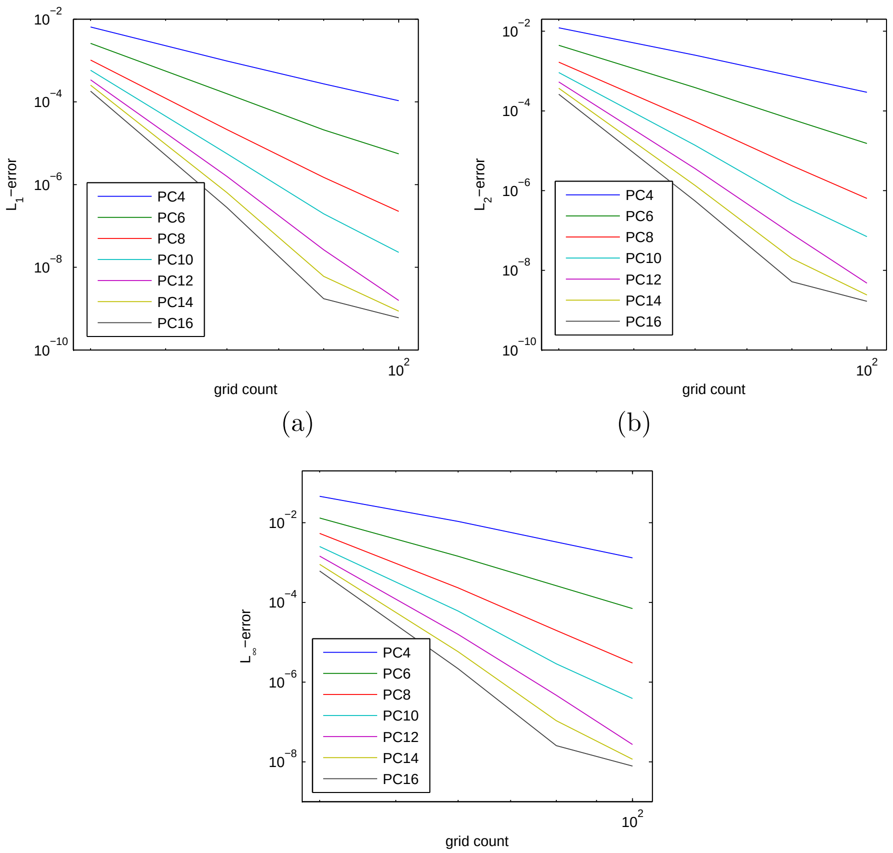
Table 4. L_2 error comparison between prefactored and conventional compact schemes.

Order of accuracy	L_2 error PC4	L_2 error C4
4th	4.4090580981E-003	4.4101995527E-003
6th	6.0846046295E-004	6.0895634631E-004
8th	7.2394864471E-005	7.2375087716E-005
10th	1.5989324725E-005	1.5470197664E-005

where ϵ_1 and ϵ_2 are errors associated with the spatial steps h_1 and h_2, respectively. The smallest time step in the MacCormack time marching scheme was $k = 1E - 6$ corresponding the the sixteenth order accurate scheme. Different grid point counts are considered to study the behavior of the numerical errors. Figure 3 shows L_1, L_2 and L_∞ errors (calculated by taking a summation over all points in the grid) plotted against grid point count, corresponding to all prefactored schemes, while table 5 lists these errors and the evaluated order of accuracy (calculated via equation (4.6)). From the plots in figure 3, a linear decrease of all errors with respect to the grid count can be noticed, except slight deviations for the last grid count in the behavior of PC14 and PC16. At this accuracy level the errors approach the machine precision, but in addition the time marching may no longer provide the required accuracy (a further decrease of the time step was not pursued). The evaluated orders of accuracy included in table 5 are close to the theoretical ones (indicated by the suffixes, e.g. 'PC4' has 4th order of accuracy) suggesting that the prefactored schemes are indeed able to maintain the desired order.

The computational efficiency increase is tested for PC4, PC6, PC8 and PC10, by comparing the computation time to the case when corresponding classical compact schemes of fourth (C4), sixth (C6), eight (C8) and tenth (C10) order of accuracy, respectively, are employed. C4 and C6 schemes necessitate the inversion of a three-diagonal matrix which is performed here via the Thomas algorithm ([27]), while for the C8 and C10 schemes a pentadiagonal matrix inversion is required; this is done here using a LU-factorization, where the matrix elements are calculated only once and stored to be re-used. For classical compact schemes, a second order TVD Runge-Kutta scheme - which is equivalent to the second-order MacCormack schemes in terms of the number of operations - is used to march the solution in time. The number of grid points is significantly large - in the order of 10,000 - such that the spatial discretization consumes most of the computational time. Table 6 shows the results in terms of the computational time. By using PC4 as opposed to C4, a 41.7% decrease in the computational time is achieved; a 40.2% decrease in the computational time is achieved when employing PC6 scheme versus C6; a 33.1% decrease is achieved when applying PC6 versus C6; and a 32.3% decrease is achieved when applying PC10 versus C10.

The percentages shown in table 6 can be explained by a comparison in terms of the number of operations that are necessary when the two types of schemes are applied. It was found that fewer operations are necessary in the case of PCn schemes in all cases (for example, 4 additions and 5 multiplications per step are needed when applying the PC8 scheme, as opposed to 7 additions and 8 multiplications per stage that are needed when applying the C8 scheme)



(a) (b)

(c)

Figure 3. L_1, L_2 and L_∞ errors plotted against grid point count.

4.3. **Nonlinear advection equation.** In the context of the nonlinear advection equation, the convective velocity is $c(u) = u$. The domain boundaries are $l_1 = -1/2$

TABLE 5. L_1, L_2 and L_∞ errors for different grid point counts - linear case.

PC4 scheme			
Grid	L_1-error	L_2-error	L_∞-error
40	6.4309202E-003	1.2147772E-002	4.5647113E-002
60	9.7376776E-004	2.5019901E-003	1.0738430E-002
80	2.7148411E-004	7.4323530E-004	3.2702186E-003
100	1.0678123E-004	2.9341239E-004	1.3099043E-003
estimated order of accuracy $p = 3.9058$			
PC6 scheme			
40	2.5853741E-003	4.4073207E-003	1.3058909E-002
60	1.5631156E-004	3.8415139E-004	1.4335771E-003
80	2.1072380E-005	6.1152769E-005	2.6060378E-004
100	5.5345770E-006	1.5190237E-005	6.9914708E-005
estimated order of accuracy $p = 5.8700$			
PC8 scheme			
40	1.0269107E-003	1.6601837E-003	5.3893200E-003
60	2.1429783E-005	5.3936887E-005	2.3061682E-004
80	1.4878953E-006	4.2014630E-006	1.9661071E-005
100	2.2419349E-007	6.3432171E-007	3.0175207E-006
estimated order of accuracy $p = 8.3155$			
PC10 scheme			
40	5.7962247E-004	9.1838126E-004	2.5118507E-003
60	5.5528859E-006	1.3631097E-005	6.0412057E-005
80	1.9686060E-007	5.5050250E-007	2.8709444E-006
100	2.2838208E-008	6.9681216E-008	3.8666113E-007
estimated order of accuracy $p = 10.1140$			
PC12 scheme			
40	3.3956164E-004	5.3364060E-004	1.4421609E-003
60	1.5771299E-006	3.5518382E-006	1.5715408E-005
80	2.6383987E-008	8.0943016E-008	4.6658106E-007
100	8.7747738E-009	2.4115746E-008	1.1512857E-007
estimated order of accuracy $p = 11.9781$			
PC14 scheme			
40	2.5287271E-004	3.6913391E-004	9.0116478E-004
60	6.3975306E-007	1.3428221E-006	5.8091922E-006
80	6.0074747E-009	1.9783115E-008	1.0758384E-007
100	2.8940807E-009	7.8821369E-009	3.7617685E-008
estimated order of accuracy $p = 13.4793$			
PC16 scheme			
40	1.8127733E-004	2.6222753E-004	6.1433898E-004
60	2.7869625E-007	5.4486475E-007	2.1557661E-006
80	5.6610923E-009	1.5792774E-008	7.7338168E-008
100	2.8717976E-009	7.8132634E-009	3.7871620E-008
estimated order of accuracy $p = 15.2329$			

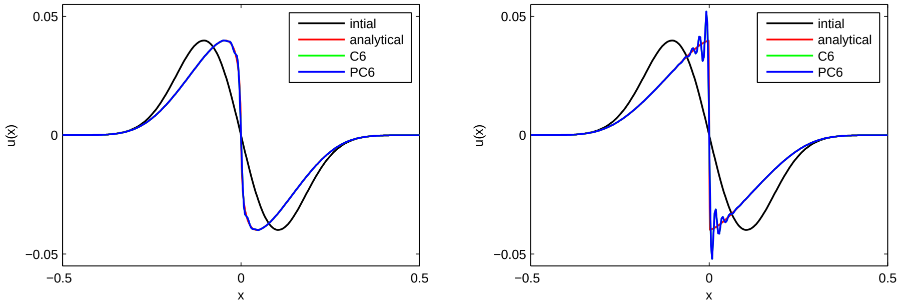
Table 6. Computational time decrease by using prefactored schemes.

Schemes	comp. time decrease (%)
PC4 vs. C4	41.7%
PC6 vs. C6	40.2%
PC8 vs. C8	33.1%
PC10 vs. C10	32.3%

and $l_2 = 1/2$, while the initial condition is

$$u_0(x) = u(x,0) = 0.1 \exp\Big(-\frac{x^2}{0.16^2}\Big)\sin(2\pi x) \tag{4.7}$$

where the Gaussian function was introduced to drive the initial waveform exponentially to zero, at both boundaries (in this way, errors generated by boundary conditions are minimized). The final time is $t_f = 1.5$. As in the linear case, the equation (4.1) is solved numerically using all prefactored compact difference schemes for different grid point counts. Figure 4 shows the numerical solution at two different time instances, using the sixth order accurate prefactored and classical compact schemes; a comparison to the analytical solution is also included along with the initial condition. A discontinuity is forming in $x = 0$, which is stronger in figure 4(a), corresponding to $t = 2.2$; here both compact schemes develop spurious oscillations around the discontinuity, but the important conclusion is that both prefactored and conventional schemes behave similarly, as expected. Table 4 lists the L_2-errors for the fourth, sixth, eight and tenth order accurate schemes (both prefactored and classical schemes); the differences between the errors are very small as in the linear case.



(a) (b)

Figure 4. Plots of the nonlinear solution for the sixth order accurate schemes: (a) $t = 1.6$, (b) $t = 2.2$.

Figure 5 shows L_1, L_2 and L_∞ errors plotted against grid point count, corresponding to all prefactored schemes. Up to tenth order of accuracy, the trends are the same as in the linear case, but for order of accuracy greater than twelfth there are some discrepancies that may be the result of the accuracy of calculating the exact solution. Table 8 lists L_1, L_2 and L_∞ errors and the evaluated order of accuracy (calculated via equation (4.6)). The orders of accuracy included in table

Table 7. L_2 error comparison between prefactored and conventional compact schemes.

Order of accuracy	L_2 error PC4	L_2 error C4
4th	1.9978793015E-007	2.0023910853E-007
6th	7.9774807530E-008	7.8986651180E-008
8th	3.1242097731E-008	2.8410542257E-008
10th	1.4655545221E-008	5.7531058004E-009

8 are close to the theoretical ones for PC4, PC6, PC8 and PC10, but there are deviations for PC12, PC14 and PC16 (due to the same reasons that are mentioned above). However, the trend of increasing the evaluated order of accuracy as the theoretical one is increased is still right.

In terms of the computational efficiency gain, by using PC4 as opposed to C4 a 40.6% decrease in the computational time is achieved; a 39.8% decrease in the computational time is achieved when employing PC6 scheme versus C6; a 31.6% decrease is achieved when applying PC8 scheme versus C8; and a 30.4% decrease is achieved when applying PC8 scheme versus C8 (see table 9).

Conclusions. In this work, a prefactorization of classical compact schemes was proposed adn studied, targeting the increase of computational efficiency and the reduction of the stencil width. The new prefactored compact schemes can be applied in the context of predictor-corrector type time marching schemes, where sweeping from one boundary to the other is possible. The wavenumber characteristics of the original schemes are preserved since by averaging the predictor and corrector operators the dissipation error vanishes, while the dispersion error matches exactly the dispersion error of the original scheme. Here, up to sixteenth order accurate compact schemes were analyzed, but theoretically any order of accuracy can be considered and tested, including existing optimized compact schemes. As an example, the coefficients of the prefactored spectral-like compact scheme of Lele [15] were calculated (given in table 10) and tested, and the results showed similar behavior as the original scheme.

L_1, L_2 and L_∞ errors for different grid point counts, both for linear and nonlinear cases, showed that the order of accuracy of the new schemes is preserved (with some deviations in the nonlinear case attributed to other external sources of errors). As expected, the computational efficiency increased by using the prefactored schemes as opposed to corresponding classical compact schemes. This was demonstrated for fourth, sixth, eighth and tenth order accurate schemes.

As mentioned above, one of the disadvantages of the prefactored schemes is the restriction to predictor-corrector type time marching schemes. Another disadvantage may be the difficulty in applying the schemes in parallel solvers and at the boundaries of the domain, although similar issues may also be encountered when employing classical high order compact schemes.

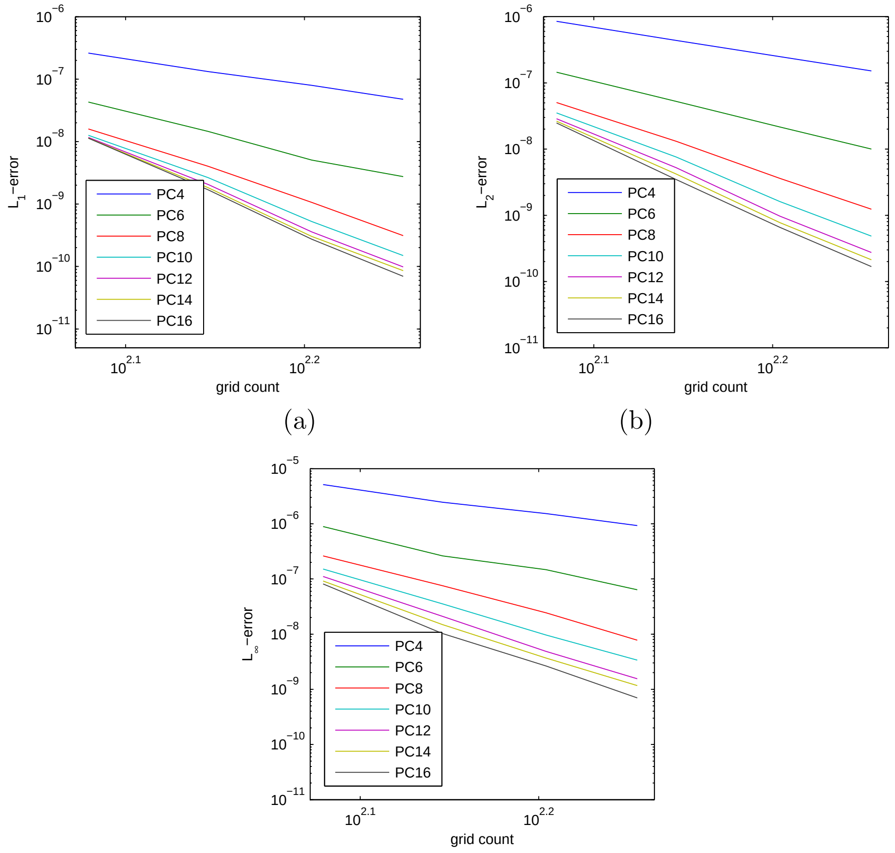
Appendix

In this appendix, a Mathematica code used to determine the weights of the optimized 12th order accurate scheme is included.

Next matrix A is from Taylor series

Table 8. L_1, L_2 and L_∞ errors for different grid point counts - nonlinear case.

PC4 scheme			
Grid	L_1-error	L_2-error	L_∞-error
120	2.6057545E-007	8.5267800E-007	5.1480039E-006
140	1.3189979E-007	4.3851923E-007	2.4635989E-006
160	7.9331220E-008	2.4853181E-007	1.5294059E-006
180	4.7629938E-008	1.5153784E-007	9.2540099E-007
estimated order of accuracy $p = 4.2243$			
PC6 scheme			
120	4.2923461E-008	1.4477527E-007	8.8853969E-007
140	1.4589376E-008	5.2417825E-008	2.6191106E-007
160	5.0682594E-009	2.1581203E-008	1.4697422E-007
180	2.7495252E-009	1.0043626E-008	6.3695264E-008
estimated order of accuracy $p = 6.3415$			
PC8 scheme			
120	1.5900337E-008	5.0506253E-008	2.6150391E-007
140	4.0357681E-009	1.3133096E-008	7.5480377E-008
160	1.0600384E-009	3.6076941E-009	2.4301185E-008
180	3.1257899E-010	1.2368444E-009	7.7387924E-009
estimated order of accuracy $p = 8.4347$			
PC10 scheme			
120	1.2630357E-008	3.5129113E-008	1.5106339E-007
140	2.6550133E-009	7.5831916E-009	3.5363402E-008
160	5.2494587E-010	1.6113340E-009	9.6153273E-009
180	1.4983937E-010	4.8509852E-010	3.3684929E-009
estimated order of accuracy $p = 9.8317$			
PC12 scheme			
120	1.1642690E-008	2.8799170E-008	1.1068219E-007
140	2.0583881E-009	5.2232619E-009	2.1042827E-008
160	3.5757610E-010	9.6967151E-010	4.8609743E-009
180	9.8160882E-011	2.7427907E-010	1.5507093E-009
estimated order of accuracy $p = 11.1075$			
PC14 scheme			
120	1.1386239E-008	2.6357760E-008	9.1586511E-008
140	1.8347290E-009	4.2047321E-009	1.4882790E-008
160	3.0278882E-010	7.7245435E-010	3.6903868E-009
180	8.5577391E-011	2.1319123E-010	1.1708572E-009
estimated order of accuracy $p = 11.4742$			
PC16 scheme			
120	1.1315127E-008	2.4784162E-008	8.0909695E-008
140	1.6970188E-009	3.4771499E-009	1.0244846E-008
160	2.7276821E-010	6.5903831E-010	2.6510809E-009
180	6.9435657E-011	1.6853644E-010	6.9679461E-010
estimated order of accuracy $p = 12.1178$			



(a)

(b)

(c)

FIGURE 5. L_1, L_2 and L_∞ errors plotted against grid point count.

TABLE 9. Computational time decrease by using prefactored schemes.

Schemes	comp. time decrease (%)
PC4 vs. C4	40.6%
PC6 vs. C6	39.8%
PC8 vs. C8	31.6%
PC10 vs. C10	30.4%

```
A = {{ 1, 1, 1, -2, -2, -2 },
{1, 2^2, 3^2, -2*3! /2!, -2*3!/2!*2^2, -2*3!/2!*3^2},
{1, 2^4, 3^4, -2*5! /4!, -2*5!/4!*2^4, -2*5!/4!*3^4},
{1, 2^6, 3^6, -2*7! /6!, -2*7!/6!*2^6, -2*7!/6!*3^6},
{1, 2^8, 3^8, -2*9! /8!, -2*9!/8!*2^8, -2*9!/8!*3^8},
{1, 2^10, 3^10, -2*11!/10!, -2*11!/10!*2^10, -2*11!/10!*3^10}};
R = {1,0,0,0,0,0};
XF = Inverse[A].R;
```

Table 10. Weights of the prefactored spectral-like compact scheme of Lele [15].

β_1	0.4482545282296
β_2	0.0817278256497
b_1	0.3069790178973
b_2	0.3294144889364
b_3	0.0113973418854

```
XF[[1]]=XF[[1]]/2; XF[[2]]=XF[[2]]/4; XF[[3]]=XF[[3]]/6;
Simplify[XF]
```

```
Solve[{a3/(1-2*a1-2*a2-2*a3)=XF[[6]],
a2/(1-2*a1-2*a2-2*a3)=XF[[5]],
a1/(1-2*a1-2*a2-2*a3)=XF[[4]]}, {a1,a2,a3}]
```

```
b1=XF[[1]]*(1-2*a1-2*a2-2*a3)
b2=XF[[2]]*(1-2*a1-2*a2-2*a3)
b3=XF[[3]]*(1-2*a1-2*a2-2*a3)
```

```
NSolve[{ap3*(1-ap1-ap2-ap3)=s3,
ap2*(1-ap1-ap2-ap3)+ap1*ap3=s2,
ap1*(1-ap1-ap2-ap3)+ap1*ap2+ap2*ap3=s1},
{ap1,ap2,ap3}, WorkingPrecision->25]
par = (1-ap1-ap2-ap3);
```

The next 1/2 is from averaging the backward and forward operators;

```
NSolve[{1/2(-ae3*par-ap3*(ae1+ae2+ae3))=-b3,
1/2(-ae2*par-ae3*ap1+ap3*ae1-ap2*(ae1+ae2+ae3))=-b2,
1/2(-ae1*par-ae2*ap1-ae3*ap2+af3*ae2+ap2*ae1
 -ap1*(ae1+ae2+ae3))=-b1},
{ae1,ae2,ae3}, WorkingPrecision->20]
```

References

[1] Adams, N. A.; Shariff, K.; (1996) A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems, *J. Comput. Phys.*, Vol. 127, pp. 27.

[2] Ashcroft, G.; Zhang, X.; Optimized prefactored compact schemes, *J. Comput. Phys.*, Vol. 190 (2003), pp. 459-477.

[3] Bose, R.; Sengupta, T. K.; Analysis and Design of a New Dispersion Relation Preserving Alternate Direction Bidiagonal Compact Scheme, Journal of Scientific Computing, Vol. 61 (2014), pp. 1-28

[4] Bruger, A.; Gustafsson, B.; Lotstedt, P.; Nilsson, J.; High-order accurate solution of the incompressible Navier-Stokes equations, *J. Comput. Phys.*, Vol. 203 (2005), pp. 49.

[5] Chu, P. C.; Fan, C.; A three-point combined compact difference scheme, *J. Comput. Phys.*, Vol. 140 (1998), pp. 370-399.

[6] Deng, X.; Maekawa, H.; Compact high-order accurate nonlinear schemes, *J. Comput. Phys.*, Vol. 130 (1997) , pp. 77-91.

[7] Deng, X.; Zhang, H.; Developing high-order weighted compact nonlinear schemes, *J. Comput. Phys.*, Vol. 165 (2000), pp. 22-44.

[8] Eswaran, De A.K.; Analysis of a new high resolution upwind compact scheme, *J. Comput. Phys.*, Vol. 218 (2006), pp. 398-416.

[9] Fu, D. X.; Ma, Y. W.; A high-order accurate finite difference scheme for complex flow fields, *J. Comput. Phys.*, Vol. 134 (1997), pp. 1-15.

[10] Fu, D. X.; Ma, Y. W,; Analysis of Super Compact Finite Difference Method and Application to Simulation of Vortex-Shock Interaction, *Int. J. Numer. Meth. Fluids*, Vol. 36 (2001), pp. 773-805.

[11] Hardin, J. C.; Ristorcelli, J. R.; Tam, C. K. W.; ICASE/LaRC Workshop on benchmark problems in Computational Aeroacoustics, NASA Conference Publication 3300 (1995).

[12] Hixon, R.; Prefactored small-stencil compact schemes, *J. Comput. Phys.*, Vol. 165 (2000), pp. 522-541.

[13] Hixon, R.; Turkel, E.; Compact implicit MacCormack-type schemes with high accuracy, *Journal of Computational Physics*, Vol. 158 (2000), pp. 51-70.

[14] Kim, J. W.; Lee, D.J.; Optimized compact finite difference schemes with maximum resolution, *AIAA Journal*, Vol. 34 (1996), pp. 887-893.

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

[16] Li, M.; Tang, T.; Fornberg, B.; A compact fourth-order finite difference scheme for the steady incompressible NavierStokes equations, *Int. J. Numer. Meth. Fluids*, Vol. 20 (1995), pp. 1137-1151.

[17] Liu, W. E. J.-G.; Essentially compact schemes for unsteady viscous incompressible flows, *J. Comput. Phys.*, Vol. 126 (1996), pp. 122-138.

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

[19] MacCormack, Robert W.; The Effect of Viscosity in Hypervelocity Impact Cratering. *AIAA Paper AIAA-69-354* (1969).

[20] Mahesh, K.; A family of high order finite difference schemes with good spectral resolution, *J. Comput. Phys.*, Vol. 145 (1998), pp. 332-358.

[21] Meitz, H. L.; Fasel, H. F.; A compact difference scheme for the NavierStokes equations in velocity-vorticity formulation, *J. Comput. Phys.*, Vol. 157 (2000), pp. 371.

[22] Sescu, A.; Hixon, R.; Numerical anisotropy study of a class of compact schemes, *Journal of Scientific Computing*, (2013), DOI 10.1007/s10915-014-9826-0.

[23] Sengupta, T. K.; Dipankara, A.; Kameswara Rao, A.; A new compact scheme for parallel computing using domain decomposition, *J. Comput. Phys.*, Vol. 220 (2007), pp. 654-677.

[24] Shah, A.; Yuan, L.; Khan, A.; Upwind compact finite difference scheme for time-accurate solution of the incompressible NavierStokes equations, *Applied Mathematics and Computation*, Vol. 215 (2010), pp. 3201-3213.

[25] Shen, Y. Q.; Yang, G. W.; Gao, Z.; High-resolution finite compact differences for hyperbolic conservation laws, *J. Comput. Phys.*, Vol. 216 (2006), pp. 114-137.

[26] Shukla, R. K.; Tatineni, M.; Zhong, X.; Very high-order compact finite difference schemes on non-uniform grids for incompressible NavierStokes equations, *J. Comput. Phys.*, Vol. 224 (2007), pp. 1064-1094.

[27] Thomas, L. H.; Elliptic Problems in Linear Differential Equations over a Network, Watson Sci. Comput. Lab Report, Columbia University, New York (1949).

[28] Wolfram Research, Inc.; Mathematica, Version 10.0, Wolfram Research, Inc., Champaign, Illinois (2014).

Adrian Sescu
Mississippi State University, Mississippi State, MS 39762, USA
E-mail address: sescu@ae.msstate.edu, phone 1-662-325-7484, fax 1-662-325-7730