Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational Physics 220 (2007) 654–677

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A new compact scheme for parallel computing using domain decomposition

T.K. Sengupta *, A. Dipankar, A. Kameswara Rao

Department of Aerospace Engineering, I.I.T. Kanpur, U.P. 208016, India

Received 16 January 2006; received in revised form 17 May 2006; accepted 22 May 2006
Available online 18 July 2006

Abstract

Direct numerical simulation (DNS) of complex flows require solving the problem on parallel machines using high accuracy schemes. Compact schemes provide very high spectral resolution, while satisfying the physical dispersion relation numerically. However, as shown here, compact schemes also display bias in the direction of convection – often producing numerical instability near the inflow and severely damping the solution, always near the outflow. This does not allow its use for parallel computing using domain decomposition and solving the problem in parallel in different sub-domains. To avoid this, in all reported parallel computations with compact schemes the full domain is treated integrally, while using parallel Thomas algorithm (PTA) or parallel diagonal dominant (PDD) algorithm in different processors with resultant latencies and inefficiencies. For domain decomposition methods using compact scheme in each sub-domain independently, a new class of compact schemes is proposed and specific strategies are developed to remove remaining problems of parallel computing. This is calibrated here for parallel computing by solving one-dimensional wave equation by domain decomposition method. We also provide the error norm with respect to the wavelength of the propagated wave-packet. Next, the advantage of the new compact scheme, on a parallel framework, has been shown by solving three-dimensional unsteady Navier–Stokes equations for flow past a cone-cylinder configuration at a Mach number of 4.

Additionally, a test case is conducted on the advection of a vortex for a subsonic case to provide an estimate for the error and parallel efficiency of the method using the proposed compact scheme in multiple processors.

Keywords: Compact schemes; Domain decomposition method; Parallel computing; DNS

1. Introduction

For parallel computing, domain decomposition method of Schwartz [1,2] is preferred in either its multiplicative or additive variations [3] – where a complex large problem is solved in smaller sub-domains independently and exchanging overlap region data or information among conjoint sub-domains. While the original Schwartz method uses overlapping sub-domains – this is inefficient in parallel computing procedures due to various latencies of processes caused by interprocessor communications among sub-domains, specially for

* Corresponding author. Fax: +91 512 590007/597561.
E-mail address: tksen@iitk.ac.in (T.K. Sengupta).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.05.018

multiplicative Schwartz method (MSM) as compared to additive Schwartz method (ASM) [3] in parallel computing. Larger the size of overlap region, more problematic it becomes for parallel computing, as it involves larger data transfer among nodes. Thus, this overlapping region is removed in many variants where the non-overlapping sub-domains interact with their neighbors via interface boundary condition(s) only. The non-overlapping domains are also preferred when the grids are non-matching in contiguous sub-domains.

To remedy the slowness of convergence of the Schwartz's method, Lions [4] proposed to replace Dirichlet interface conditions by Robin interface conditions and the parameters of the artificial boundary conditions (ABC) were obtained via optimization for faster convergence rate. The resultant convergence rate is function of the parameters of ABC and on the amount of overlap among sub-domains. The latter actually increases convergence exponentially for CFD problems governed by convection and diffusion operators, when the flow is normal to the interface and the time step is small. For large time steps and when flow is tangential to the interface then this type of exponential convergence is absent. This information helps in the decision of decomposing the given flow domain and would be used in our examples. Also, the non-overlapping sub-domains suffer from poor convergence for high wave number components of the error [3]. It is to be noted that the convergence properties of Schwartz's method are critical for parabolic and elliptic PDEs and not so significant for hyperbolic or propagation problems.

Most parallel computations are performed using explicit formulations to avoid problems of passing volume of data among various sub-domains that becomes mandatory for implicit methods. Shang et al. [5] and Lockard and Morris [6] have reported efficient parallel explicit algorithms for multi-physics problems. While some efforts have been made in [7] towards developing parallel codes using high accuracy compact schemes, according to these authors *efficient implementation of compact schemes on parallel computers remains an open problem*. Basic compact schemes [8] have enormous advantages of large spectral resolution as compared to high order explicit schemes and hence preferred, whenever possible. It has also been shown in [9,10] that compact schemes with appropriate time integration schemes help preserve the physical dispersion relation in numerical sense – known as the dispersion relation preservation (DRP) property, that is mandatory for DNS. However, compact schemes with its associated one-sided boundary closure schemes (ABC) gives rise two inconvenient features of these methods. Firstly, a large number of often-used compact schemes have numerical instability problems near the inflow of the domain. Secondly, near the outflow, compact schemes give rise to massive attenuation of the function. In [9], a matrix-stability analysis method was introduced that quantitatively identified these shortcomings of compact schemes. The authors also provided newer compact schemes that avoid the above numerical instability problems. This peculiarity of compact schemes gives the solution a distinct bias to the procedure. When compact schemes are used in an integral domain, the ensuing instability is not persistent as the disturbance propagates downstream where the scheme has a stable behavior. However this will not be the case when the domain is segmented in the streamwise direction. Thus, this prohibits usage of compact schemes in individual sub-domains – where the solutions will display nonphysical growth and attenuation near the sub-domain interfaces for convection dominated phenomenon. This property will be demonstrated by a wave propagation problem solved in segmented overlapping domains. We will also suggest various means to overcome this problem. All earlier attempts where conventional compact schemes have been used, the flow domain is treated integrally, while various solution methods involved in compact schemes are computed in a distributed fashion.

This is done by performing various operations associated with obtaining derivatives, integrals and/or filtering operation via distributed computing. In most of these cases one is required to solve linear algebraic equations simultaneously. For the often-used compact schemes, this requires solving banded tridiagonal matrix equations by Thomas algorithm that requires O(N) operations for N unknowns. In [7] pipelined implementation of the Thomas algorithm (PTA) is discussed where various latencies of processors are reduced by performing non-local data-independent computations, solving for other spatial derivatives during forward and backward operations in PTA. Alternative to PTA is discussed in [11,12], where newer algorithms are proposed that replaces the forward and backward recursions of PTA by matrix-vector multiplications. However, this leads to significant increase in floating-point operations, defeating the rationale of faster computing by parallelization.

In another alternative a parallel diagonal dominant (PDD) algorithm was proposed in [13] for solving Toeplitz tridiagonal system arising from the usage of compact schemes. As the PDD algorithm is necessarily an approximation, the enhanced accuracy of compact schemes is compromised, in addition to incurring higher computational effort as compared to PTA.

In all parallel implementations referred to above, computations are done by solving the problem together by executing PTA or PDD algorithm in different processors, that imposes latencies and inefficiencies of computations. The above discussion raises the possibility to consider a strategy where one splits the full problem into multiple sub-domains and use Schwartz's method – eliminating all the problems of PTA and PDD algorithms, provided one can remove the unphysical bias of compact schemes. This can be attempted by a large overlap of contiguous sub-domains- since it has been established in [3] that the application of Schwartz's method via domain decomposition will have little or no problems of convergence if the overlap is sufficiently large. Such overlap of sub-domains are also required, as the analysis of [9] indicates that the problematic bias of traditional compact schemes exist over the first and last six to seven nodes in the domain. However, this would mean additional repetitive computations over twelve to fourteen nodes in every sub-domain. It would be preferable if the number of overlapping nodes can be reduced significantly. In a recent effort [14], a new compact scheme has been introduced that removes the asymmetry of basic compact schemes, in solving the sub-critical instability problem of plane Poiseuille flow that requires obtaining very high accuracy symmetric equilibrium solution. In the present research, a new class of symmetrized compact schemes is established furthermore by designing and using two such unbiased schemes to: (a) solve a wave propagation problem; (b) solve three-dimensional unsteady Navier–Stokes equation for high Mach number supersonic flow ($M_\infty = 4.0$) past a cone-cylinder configuration; and (c) convection of a vortex governed by inviscid equations for subsonic Mach number ($M_\infty = 0.4$). It should however, be pointed out that while the bias of compact schemes can be minimized – it cannot be completely removed. In such a situation, a Schwartzian domain decomposition may still give rise to numerical problems for parallel computing near the interfaces that requires detailed investigation. In the present work, we want to reduce the overlap to a minimum, so that an efficient parallelization is made possible. Apart from the usage of larger overlap of contiguous sub-domains to reduce numerical problems in parallel computing, we also propose here to use filtering in the physical space [15] or adding artificial numerical dissipation to remove the same problems.

The paper is formatted in the following manner. In the next section a new symmetrized compact scheme is developed. The developed scheme is tested for the propagation of a wave-packet following the one-dimensional wave equation. This will help establish the numerical properties of the developed symmetrized compact scheme for its stability and dispersion relation preservation property with respect to different parallelizing strategy using domain decomposition method. We also show the effects of physical plane filters in controlling some numerical problems of parallel computing. In Section 3, we use a symmetrized scheme to solve the three-dimensional supersonic flow at $M_\infty = 4.0$ past a cone-cylinder assembly by solving the unsteady three-dimensional Navier–Stokes equation. To test the ability of the proposed method for elliptic problems, we have studied the convection of a shielded vortex in inviscid flow at $M_\infty = 0.4$ in Section 4. The paper closes with some conclusions in Section 5.

2. Development of a symmetrized compact scheme for parallel computing

It is well known that compact difference schemes [8,9] used for solving PDEs possess very high spectral accuracy in resolving various spatial scales. For CFD and many other related activities, these schemes are used to evaluate various derivatives in an implicit manner. For example, to evaluate first derivatives u' of a vector $u(x_j)$, $j = 1, \ldots, N$ one can write it down as,

$$[A]\{u'\} = [B]\{u\}. \tag{1}$$

While this representation is valid for both explicit (with A as an identity matrix) and implicit methods, an equivalent explicit representation of this general form can be written down as,

$$\{u'\} = \frac{1}{h}[C]\{u\} \tag{2}$$

where h is the uniform grid spacing used for discretizing the domain. However, one does not work with C matrix while using compact schemes. Many practical problems require non-periodic boundary conditions (as provided from physical considerations or needed to close the system given in Eq. (1) – referred to as ABC in Section 1) that mandates one-sided stencils near and at the boundary nodes. This requirement makes A and B matrices non-symmetric (non-Hermitian). In the interior of the domain, symmetric entries of the B matrix correspond to non-dissipative central schemes, while non-symmetric entries of B matrix arise for upwinded compact schemes – see [8,10,14] for details. This shows that whether we choose a symmetric or non-symmetric interior stencils, one-sided boundary closure schemes for non-periodic problems, make A and B matrices always asymmetric. Effects of such asymmetric stencils near boundaries percolate in the interior of the computing domain. In the following we report a very low-bias compact scheme based on the optimal scheme OUCS4 introduced in [10]. This scheme was optimized to minimize truncation error of a central compact scheme whose stencils are already given in [10].

If the first derivatives at different nodes are represented by prime, following central stencil is used for interior points,

$$\alpha u'_{j-1} + u'_j + \alpha u'_{j+1} = \frac{1}{h}\sum_{k=-2}^{2} a_k u_{j+k} \tag{3}$$

with $a_0 = 0$, and $a_{-2} = -a_2$; $a_{-1} = -a_1$. While this stencil alone is sufficient to evaluate first derivatives for periodic problems, for non-periodic problems we have to supplement the relation given by above – as it cannot be used directly near and at the boundary due to the stencil size on the right-hand side. This is circumvented by using the following one-sided explicit boundary closure schemes for $j = 1$ and $j = 2$, respectively [9],

$$u'_1 = \frac{1}{2h}(-3u_1 + 4u_2 - u_3) \tag{4}$$

$$u'_2 = \frac{1}{h}\left[\left(\frac{2\gamma_2}{3} - \frac{1}{3}\right)u_1 - \left(\frac{8\gamma_2}{3} + \frac{1}{2}\right)u_2 + (4\gamma_2 + 1)u_3 - \left(\frac{8\gamma_2}{3} + \frac{1}{6}\right)u_4 + \frac{2\gamma_2}{3}u_5\right] \tag{5}$$

Similarly, one can write down the boundary closure schemes for $j = N$ and $j = (N - 1)$ using γ_{N-1}. Note that the sign of the coefficients in boundary closure schemes are reversed on the right hand side on the opposite boundaries and this lead to large difference of added numerical dissipation near the boundaries causing asymmetry. With the help of relations given in Eqs. (3)–(5), one can construct A, B and C matrices in Eqs. (1) and (2) readily. The interior stencil, given by Eq. (3) applies at $j = 3$ to $N - 2$ and its structure makes A a tridiagonal matrix, while B is a penta-diagonal matrix. We introduce briefly the analysis method [9,10] below that explains the above mentioned asymmetry clearly.

The analysis is performed in the wave number plane, with the unknown expressed in terms of original and bi-lateral Laplace transform pair,

$$u(x_j) = \int_{k_{\min}}^{k_{\max}} U(k)e^{ikx_j}\,dk \tag{6}$$

where $k_{\min}$ and $k_{\max}$ denote the minimum and maximum wave numbers that has been resolved by the discrete computing method. Usage of the bilateral Laplace transform instead of Fourier series removes the restriction of periodicity for the unknowns. In spectral methods, $k_{\max} = -k_{\min} = \pi$, as one works with complex variables with its complex conjugate, so that the represented function is purely real. Other discrete computing methods work in the physical plane with $k_{\min} = 0$ and all the variables are 2π-periodic. The derivative of the function at x_j can be expressed for an exact method as $u'(x_j) = \int ikU(k)e^{ikx_j}\,dk$ and corresponding expression for the numerical derivative using other discrete computing methods can be written as,

$$u'(x_j) = \int ik_{\text{eq}}U(k)e^{ikx_j}\,dk$$

Using Eq. (6) in Eq. (2) and comparing with the above, it is readily seen that

$$ik_{\text{eq}}(x_j) = \sum_{l=1}^{N} C_{lj}e^{ik(x_l - x_j)} \tag{7}$$

Thus, the jth row of C matrix determines the derivative at the jth node – as given by k_{eq} in Eq. (7). In [10] the following values of the parameters were obtained and termed as OUCS4: $a_1 = 1.546277$, $a_2 = 0.329678$, $\gamma_2 = -0.025$ and $\gamma_{N-1} = 0.09$.

Asymmetric behavior of compact schemes can be clearly noted from Eq. (7) due to the asymmetric C matrix. The reason for the asymmetry of C is due to one-sided boundary closure schemes given by Eqs. (4) and (5). Symmetrization of C matrix can be brought about by either of two methods. In the first method we simply take an arithmetic average of the entries of C matrix in Eq. (2) about the center row (that represents the middle of the domain), thereby making the derivative symmetric and removing the bias of the compact scheme. This was suggested and followed in [14]. There is the second method where we evaluate the derivatives using Eq. (1) at all nodes by going from $j = 1$ to N and going from $j = N$ to 1 in each sub-domain. The symmetrized derivative are taken as the average of the derivatives obtained in following these two directions. While these two alternative procedure of symmetrizing the derivatives are identical, in the first procedure one performs N^2 multiplications and in the second process this involves solving Thomas algorithm twice, thereby performing $10N$ operations. Thus, if one chooses $N > 10$, the second method is faster, and such a choice of N is always made to avoid the negative effects of boundary closure schemes. We will refer to this method of evaluating derivatives as the S-OUCS4 scheme in the rest of the paper. It is important to investigate that apart from removing the bias of compact schemes, whether the new scheme retains or improves the numerical properties, the amplification rate and DRP. These properties are studied here with respect to the model one-dimensional linear convection equation that represents many convective flows and wave phenomena,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad c > 0 \tag{8}$$

Eq. (8) is non-dispersive and convects the initial solution to the right with speed c i.e. the group velocity is equal to the phase speed c. This equation helps study numerical instability and most importantly the dispersion error and DRP property [9,14], using physical and numerical group velocity. It is necessary to ensure that the numerical group velocity is as close to the physical group velocity as possible for the resolved space-time scales. This is discussed next along with the numerical stability of space-time discretization schemes for Eq. (8).

Consider the numerical solution of Eq. (8) subject to the following initial condition,

$$u_m^0 = u(x_m, t = 0) = \int A_0(k)\mathrm{e}^{\mathrm{i}kx_m}\,\mathrm{d}k \tag{9}$$

The exact solution of Eq. (8) can be written down in terms of the initial solution as,

$$u_{\text{exact}}(x, t) = \int A_0(k)\mathrm{e}^{\mathrm{i}k(x-ct)}\,\mathrm{d}k \tag{10}$$

The numerical solution can be represented at $x = x_m$ and $t = t^n$ by $u_m^n = \int \widehat{U}(k, t^n)\mathrm{e}^{\mathrm{i}kx_m}\,\mathrm{d}k$, so that a numerical amplification factor can be introduced by $G(k) = \frac{\widehat{U}(k,t^{n+1})}{\widehat{U}(k,t^n)}$.

One can compare the exact solution with that obtained numerically by using Eq. (9) in discretized Eq. (8) to obtain general numerical solution [10] as,

$$u_m^n = u(x_m, t^n) = \int A_0(k)(G_r^2 + G_i^2)^{n/2}\mathrm{e}^{\mathrm{i}k(x_m - n\beta)}\,\mathrm{d}k \tag{11}$$

where the numerical amplification factor is given by $G(k) = G_r(k) + \mathrm{i}G_i(k)$ and $\tan\beta = -G_i/G_r$. The numerical group velocity is found from the numerical dispersion relation as given by [10],

$$\frac{V_{gN}(k)}{c} = \frac{1}{N_c h}\frac{\mathrm{d}\beta}{\mathrm{d}k} \tag{12}$$

where $N_c = c\Delta t/h$ is the CFL number.

For non-periodic problems, the corresponding quantities will vary from node to node. If for a discrete computing scheme the spatial derivative at the jth node is evaluated using Eq. (7), then it is given by,

$$\frac{\partial u_j}{\partial x} = \frac{1}{h}\sum_{l=1}^{N} C_{lj}\mathrm{e}^{\mathrm{i}k(x_l - x_j)}u_j \tag{13}$$

Utilizing this in Eq. (8) one gets

$$\frac{\partial u_j}{\partial t} + \frac{cu_j}{h} \sum C_{lj}[\cos k(x_l - x_j) + \mathrm{i} \sin k(x_l - x_j)] = 0 \tag{14}$$

This equation can be used to obtain wave-number dependent amplification factor at the *j*th node, $G_j(k)$, once the time integration method is fixed. We have used the four-stage Runge–Kutta (RK_4) time integration strategy here, as it was noted to have good numerical stability and DRP property [14], when used with traditional compact schemes. The amplification factor for the RK_4 time integration scheme, for *j*th node can be shown to be given by [14]:

$$G_j(kh, N_\mathrm{c}) = 1 - A_j + \frac{A_j^2}{2} - \frac{A_j^3}{6} + \frac{A_j^4}{24} \tag{15}$$

where

$$A_j = N_\mathrm{c} \sum_{l=1}^{N} C_{lj} \mathrm{e}^{\mathrm{i}k(x_l - x_j)} \tag{16}$$

Using Eqs. (15) and (16), one obtains the amplification factors at all spatial nodes for any combinations of kh and N_c for RK_4 time integration scheme. In the above, we can use the symmetrized C matrix by replacing the entries of the *j*th and $(N - j + 1)$th rows of the C matrix by their arithmetic average, where N is the number of points. Such symmetrizing operation will also alter the above discussed numerical properties. In Fig. 1, $|G_j|$ at different representative nodes are shown by taking $N = 101$. On the left column of Fig. 1, shown are the amplification rate contours for the *j*th node and on the right corresponding results for $(N - j + 1)$th nodes for the original OUCS4 method. In the middle column of Fig. 1, corresponding results are shown for the S-OUCS4 scheme at the same nodes. Results for $j = 2$ exhibit unstable nature, even for very small values of N_c, while for $j = 100$ we note excessive damping for the OUCS4 scheme. In contrast, S-OUCS4 method has significantly improved amplification rates for $j = 2$ and $(N - 1)$. Similar improvements have been brought about at all near-boundary points. It is seen that there is negligible asymmetry of $|G|$ for $j \geqslant 8$ onwards for the basic OUCS4 scheme that is removed by symmetrization.

In Fig. 2, the effects of symmetrizing is shown on V_{gN}/c, by plotting the properties of OUCS4 scheme on either side, with the S-OUCS4 value plotted in the middle column. Once again, an overall improvement of DRP property by symmetrizing is noted. It is also seen that one has to take very small values of N_c to avoid effects of spurious dispersion.

Informations contained in Figs. 1 and 2 can be effectively used for solving problems using parallel computing via domain decomposition. To calibrate the developed procedure, we solve Eq. (8) for the propagation of a wave-packet whose initial profile and location is given by,

$$u_0(x) = \mathrm{e}^{-\phi(x - x_0)^2} \sin(k_0 x) \tag{17}$$

where $\phi = 50$ and $x_0 = 1.5$, is the center of the packet in the physical plane and $k_0 = 0.838242/h$ is the center of the packet in the spectral plane. The problem is solved in the domain $0 \leqslant x \leqslant 10$ using four equal length sub-domains for parallel computing. Initially, the center of the packet lies in the interior of the first sub-domain. Implementation of the S-OUCS4 scheme for this problem is made following the overlap of the sub-domains shown in Fig. 3, i.e. the *q*th processor receives information from $(q \pm 1)$th processors as indicated by the arrows in the figure. Thus, the evaluation of derivatives by compact scheme is performed in the *q*th processor from $i = -1$ to $M + 2$, while the problem is solved from $i = 1$ to M.

The specific wave packet given by Eq. (17) is considered to convect to the right at the phase speed, $c = 0.5$. Initially, this problem is solved sequentially by using a single grid with 664 points and $N_\mathrm{c} = 0.02$, so that $\Delta t = 0.0006$ and for this parameter combinations one gets stable solution that is shown in Fig. 4(a) at the indicated time instants. Next, the same problem is solved using four nodes of a parallel computer with 12 points overlapping – six for each of the neighboring sub-domains. Thus, the number of points per processor in this case is 178 and the results are as shown in Fig. 4(b) at the same time instants, as that is shown for the sequentially computed results in Fig. 4(a). A direct comparison reveals that the two computed packets are exactly identical. The only differences that one notices are the appearance of negligibly small spurious wave-packets

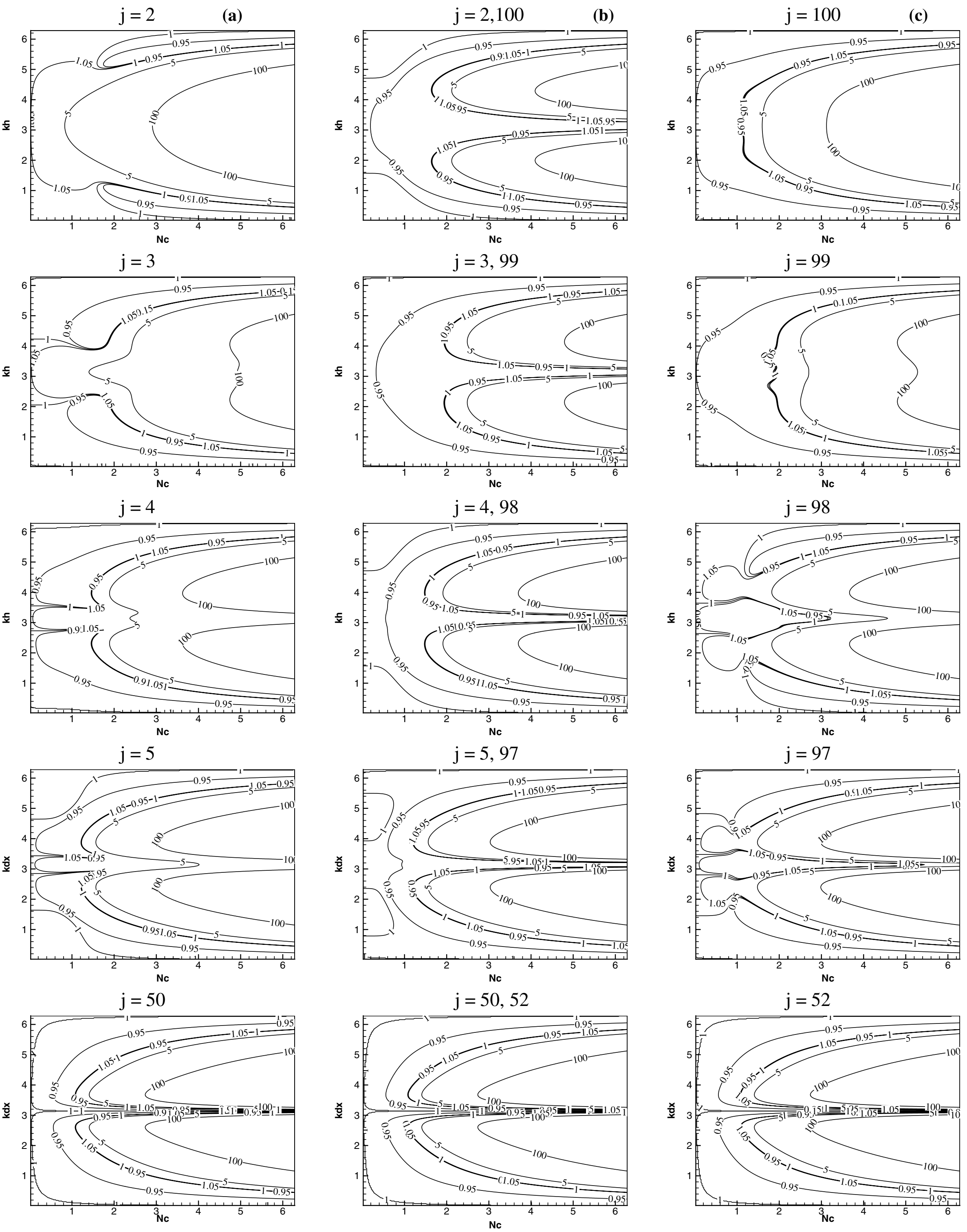

Fig. 1. Amplification factor ($|G|$) contours in (kh-N_c)-plane for (a), (c) OUCS4 and (b) S-OUCS4 at the indicated nodes.

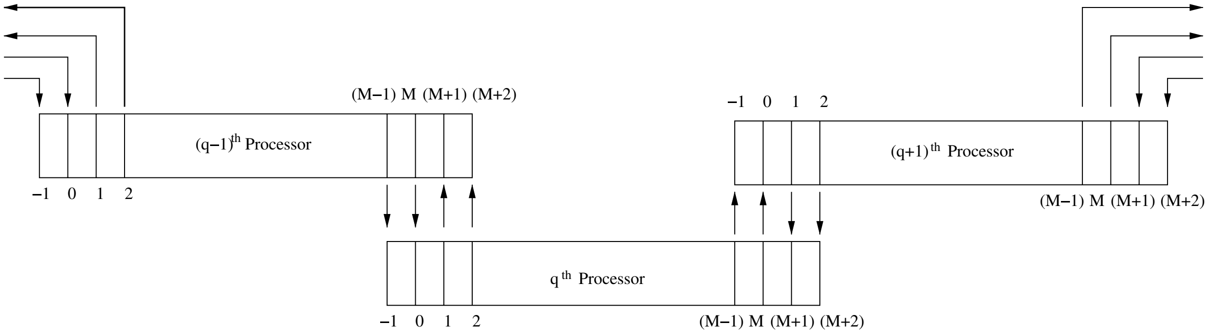

Fig. 3. Schematic representation of the overlap region between the processors.

in Fig. 4(b) near the interface boundaries visible in the frames at $t = 2.4$, 4.2, 9 and 15. These small spurious packets are due to the properties of near-boundary stencils of S-OUCS4 and can be removed by having larger overlap regions. Also, it has been pointed out [3] that with larger overlap the convergence rate of the Schwartz method improves further. In the present computations, there was no need to iterate the solutions in different sub-domains as the correct results were obtained in single pass of computations.

As noted in the case of Fig. 4, to obtain the solutions by parallel computing, we required to add extra 12 points per processor and that also has some negligibly small spuriously dispersing errors near the sub-domain interfaces. A complete removal of which would require larger overlap region among contiguous sub-domains and hence extra data communications. This naturally raises the question, as to whether one can obtain accurate results via parallel computing using lesser number of overlapping nodes while using other methods of error control near the interfaces. For this purpose, we decided to take only 2 overlap layers i.e. the number of points per processor in this case reduces to 170. The results of this simulation, without any error control, is shown in Fig. 5(a) at the indicated time frames, that show large amplitude (but decaying) error-packets that move downstream after initial upstream traveling error-packets are created that is reflected from the inflow of the domain. When the wave-packet exit the first sub-domain at around $t = 2.0$, small error-packets are generated locally, wherever the signal changes sign, that travel upstream. At $t = 2.4$ this process creates four such packets moving upstream at $V_g \approx -2c$. Fourier transform of the solution at this time indicates the presence of an additional peak at $kh = 3.05$. From Fig. 2, it is seen that this matches with the above propagation speed of the error that does neither grow or decay significantly as from Fig. 1, one notes the value of $|G|$ at the same kh to be close to one. This process of creating error packets continues as the signal encounters every interface. Such upstream propagation of error packet is also noted in Fig. 4(b), when six point overlap was considered. However, the amplitude of error was comparatively smaller, that could be further reduced by even larger overlap.

However, from Fig. 5(a), one also notices large reflected downstream propagating packets from the inflow of the domain. These packets are created due to the reflection of those small upstream propagating error-packets from the inflow. FFT of the solutions at $t = 0.0$ and 4.2 revealed identical spectrum, implying these reflected packets to have the same kh as that of the input signal. It is not known, at this stage, the reason for such scale selection of these reflected error-packets from inflow.

To remedy the situation, we have filtered the solution in the physical space in each sub-domain separately following the sixth order filters given in [15] as,

$$\alpha_f \hat{u}_{i-1} + \hat{u}_i + \alpha_f \hat{u}_{i+1} = \sum_{n=0}^{3} \frac{a_n}{2} (u_{i+n} + u_{i-n}) \tag{18}$$

where $-0.5 < \alpha_f \leqslant 0.5$, with higher the value of α_f lesser the effect of filtering and attenuation at high wave numbers. In this equation, variables with hats indicate filtered quantities and the coefficients of the filter are as given in Table IV of [15]. This filter is used from $i = 2$ to $(M - 1)$ for the nodes as shown in Fig. 3. For $i = 0,1,M$ and $(M + 1)$th nodes, the filters given by Eq. (16) of [15] are used – that are also sixth order accurate. In Fig. 5(b) solutions are shown when we used the above filters after every fifty time-steps with $\alpha_f = 0.499$. The high value of α_f and the filtering frequency is determined by matching with the exact solution.

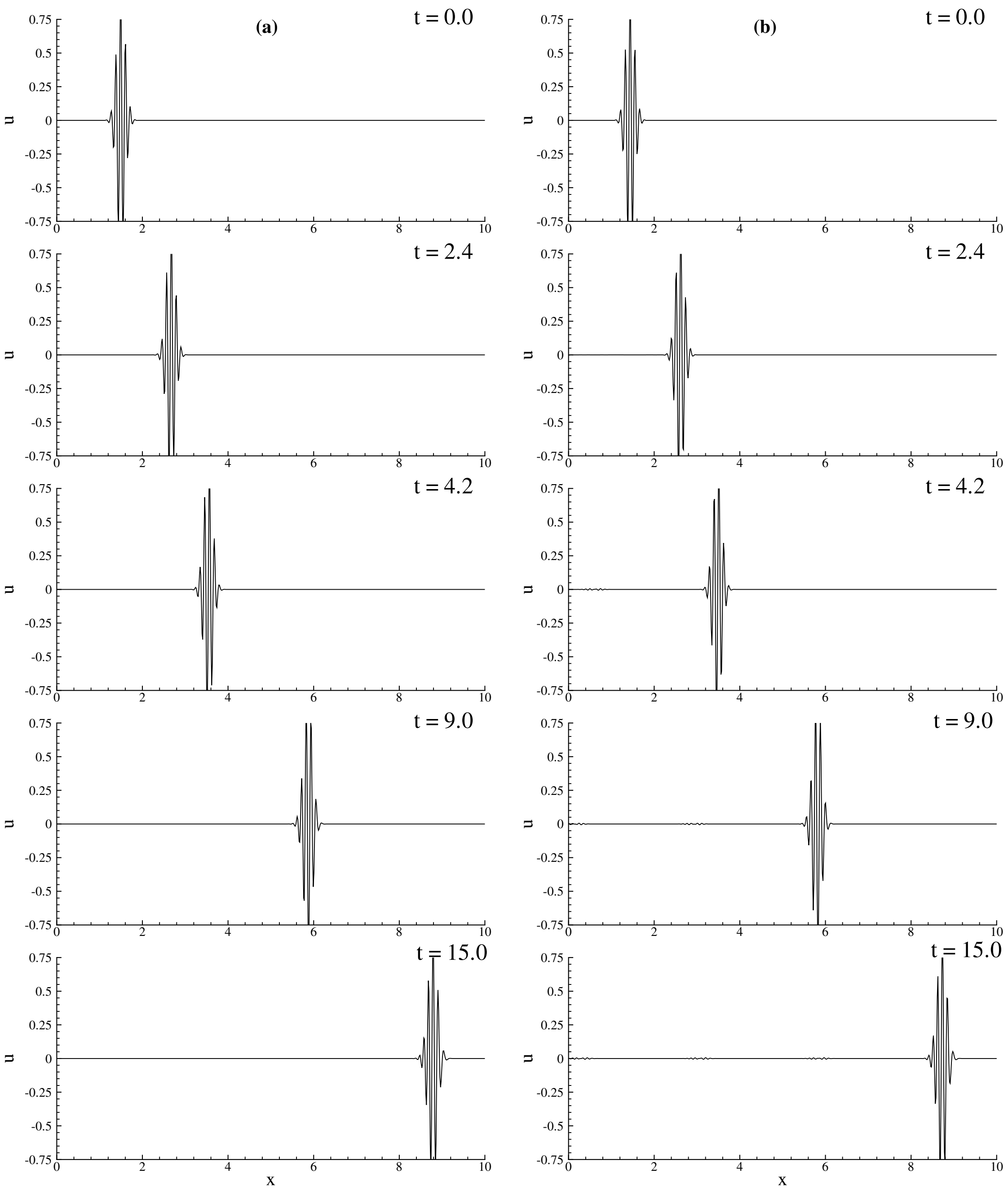

Fig. 4. Computed solution of Eq. (8) obtained using S-OUCS4 scheme for a wave packet with $k_0h = 0.838242$, using (a) sequential method (b) parallel method with 6-point overlap.

For example, higher frequency of filtering would lead to significant attenuation of the signal itself. In the same way, lower values of α_f also lead to the same problem of attenuation. Thus, it is seen that high accuracy filtering allows one to obtain correct solution with minimum overlap of sub-domains and at the same time avoiding the numerical problems due to ABC of compact schemes for parallel computing. In the following sections, we use this parallel computing procedure to solve practical problems of interest, the results of which are essential for flow transition studies.

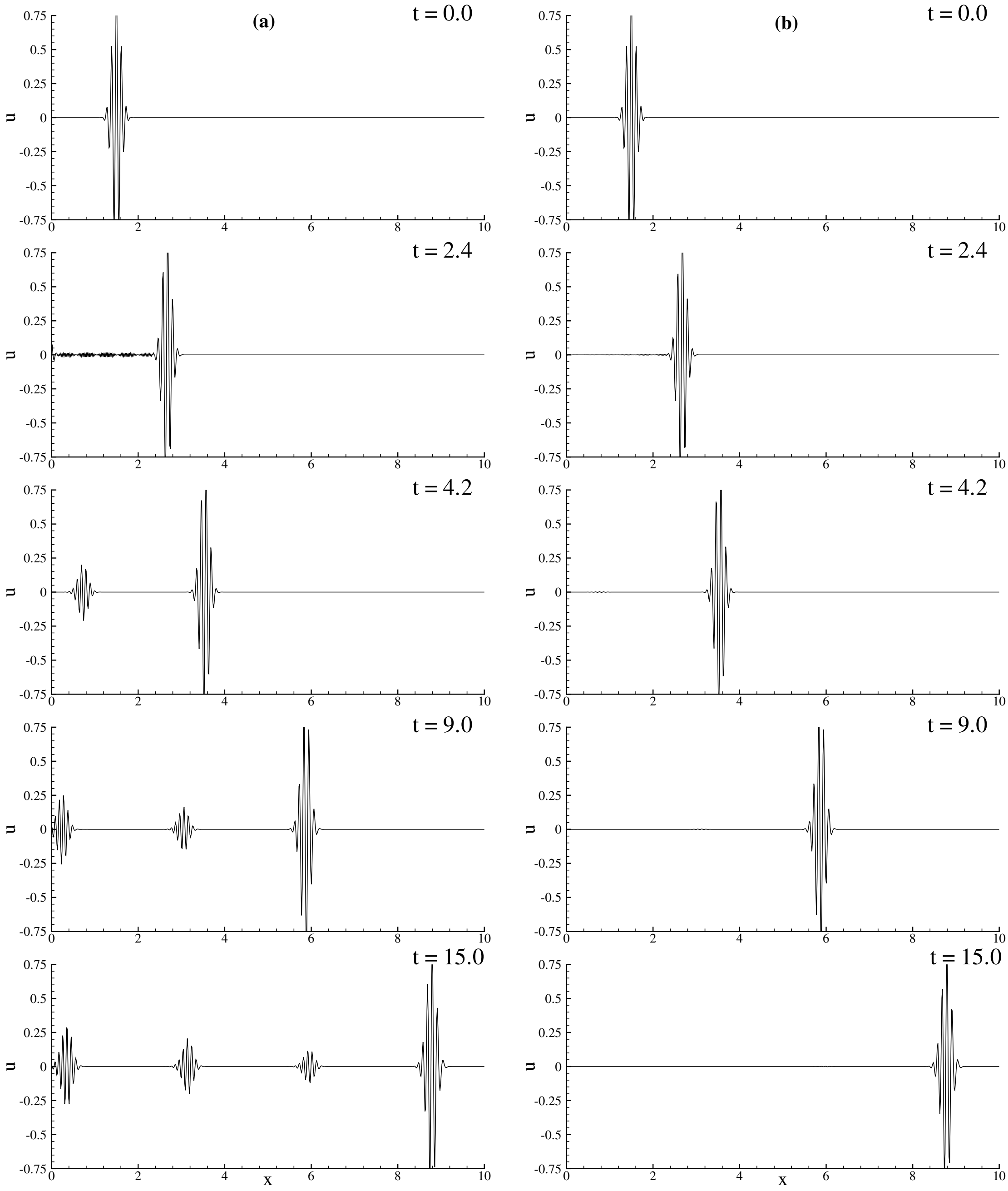

Fig. 5. Computed solution of Eq. (8) obtained using S-OUCS4 scheme in a parallel framework with 2 point overlap for, (a) without filtering and (b) with filtering cases, for the wave packet used in Fig. 4.

Next, we perform some additional cases of wave-packet propagation to show the behaviour of the error with respect to the wavelength of the propagated signal for different overlap and filter strategies. The propagating wave-packet is given by Eq. (17), with $\phi = 16$ and the initial location is such that the numerical properties displayed in Figs. 1 and 2 for the interior stencil applies in these cases. For these cases, the domain, the number of points and spacing is the same, as in the case of Figs. 4 and 5. We have varied k_0h from 0.4 to 1.6 for the following: Case-A: where we use two-point overlap in the domain decomposition method followed by filtering the solution periodically; Case-B: where we use six-point overlap without any filtering and Case-C: where the solution is obtained using six-point overlapped domains followed by periodic filtering.

The error in all these cases is calculated from, $L_\infty(\text{Error}) = \sum_{i=1}^{M}|(u_i)_{\text{exact}} - (u_i)_{\text{computed}}|/M$. We note that the error is caused primarily by: (i) attenuation of the actual signal; and (ii) by the dispersion of the computed signal. For the former source of error, the magnitude can be at most equal to the area under the wave-packet, in the absence of phase error. While the maximum error that can be created by dispersion alone- in the absence of attenuation – is equal to twice the area created under the wave packet. Therefore, in the presence of dispersion and dissipation, the maximum error can at most be equal to twice the area under the curve.

In Fig. 6(a), L_∞(Error) is plotted for the above-mentioned cases for different values of k_0h, that shows the error to be of same order of magnitude, up to $k_0h \cong 1$. For $k_0h = 1.2$, it is noted that Case-A has the least error. Case-B, like Case-A produces reflection at the domain interface that travels upstream and is left uncontrolled that causes this case to have largest error in Fig. 6(a), for all k_0h except for 1.4. This reflection of Case-B is

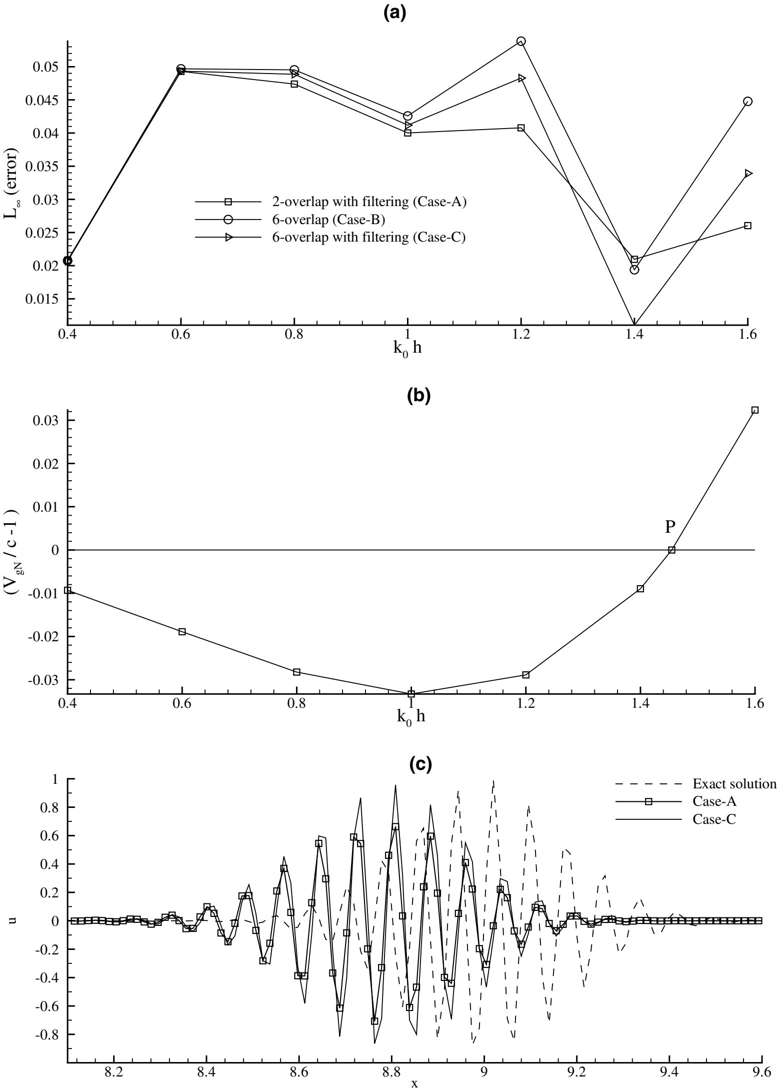

Fig. 6. (a) L_∞ norm of the error with respect to the exact solution is plotted against k_0h for the indicated cases. (b) Departure of the scaled numerical group velocity from unity (dispersion error) is shown for different values of k_0h. (c) Computed solution of Eq. (8) has been compared with the exact solution for a wave packet with $k_0h = 1.2$.

removed by filtering in Case-C. We have noted, that the two-point overlap case has larger reflected signal from the domain interface as compared to the six-point overlap cases. Thus, by the removal of the reflected signal by filtering causes the main signal to attenuate more for Case-A as compared to Case-C. We also note the error decreasing rapidly for $k_0h = 1.4$, for all the cases, with Case-C showing the least amount of error. To understand this behavior for the error, we note that in all these cases, G is equal to one, identically, as can be verified from Fig. 1 for the combinations of k_0h and N_c. However, the dispersion error, exemplified by $[V_{gN}/c - 1]$ (obtained from Fig. 2), shows its variation with k_0h for the fixed N_c, as shown in Fig. 6(b). This shows a rapid decrease in dispersion error, as k_0h increases from 1.2 to 1.4. Also, from the same figure, we note that the dispersion error increases rapidly above 1.45465 (where the dispersion error is identically zero), that is marked as point P in this figure. This feature explains the rapid increase of error at $k_0h = 1.6$ in Fig. 6(a).

Case-A show less error as compared to Case-B and -C for the following reason. Both the two- and six-point overlap cases, have identical dispersion error i.e. the computed signal is shifted with respect to the actual signal- as shown in Fig. 6(c) for $k_0h = 1.2$. But, Case-A also experience larger attenuation of the main signal as discussed above, as compared to Case-C results shown in Fig. 6(c). This damping has an equivalent effect of producing lesser departure of the computed signal from the actual one for Case-A. Despite this apparent contradiction, it is seen that more the overlap, one gets higher accuracy of the computed signal.

The feature of the dispersion error, as suggested in Fig. 6(b), is also tested by computing the wave-packet propagation cases with $k_0h = 1.2$, 1.4 and 1.45465. The last case is purposely chosen, as Fig. 6(b) indicates this to produce zero dispersion error. The computed results, using the strategy of Case-C, are shown in Fig. 7 along with the exact signal. It is noted that for $k_0h = 1.2$, $V_{gN} = 0.97112c$ – with largest dispersion error among these three cases. For $k_0h = 1.4$, the error reduces as $V_{gN} = 0.991034c$, while for $k_0h = 1.451465$, $V_{gN} = 1$, implying zero dispersion error. However, the results in Fig. 7, show for $k_0h = 1.45465$ some dispersion near the tails of the signal. This is clearly understood by noting that the wave-packet of Eq. (17) represents a spectral distribution about the above central values of k_0h. For the Hermitian property of the signal, the tails correspond to the farthest points from the central wave numbers. For $k_0h = 1.45465$, the dispersion is exactly zero at the center, while it is not identically zero on either side that is responsible for the residual departure from the exact solution.

3. Parallel computation of supersonic flow past a cone-cylinder configuration

The direct numerical simulation (DNS) of the high supersonic flow past a cone-cylinder configuration is sought here to be developed as an essential research tool for studies on Laminar-turbulent transition at hypersonic speed. Transition has a dramatic effect on heat transfer, skin friction and separation and is currently not predicted satisfactorily with available research tools. This effect is critical to reentry vehicles and air-breathing cruise vehicles, yet the physics of the transition process in not well understood. Since, the computational size of such problems to be solved with DNS is very large, it is mandatory to have a parallel DNS code that is highly efficient in terms of parallel computation. Here, we are going to report the solution of full three-dimensional NS equations in generalized coordinates, to show the superiority of the developed compact scheme (in parallel framework) over the other existing compact schemes in the literature. The essential idea here is to obtain an accurate equilibrium laminar flow past the cone-cylinder at high Mach number and Reynolds number. Thus, the presented results are by no means to be interpreted as DNS results. They essentially represent an ability of the symmetrized compact scheme to handle large problems in parallel computing. Having said that, one also notes the feasibility of stable computations with coarse grid that ensures the success of any computations with more refined grids and smaller time steps, as the information carried in Figs. 1 and 2 would clearly indicate.

3.1. Governing equations

In this section we have used the symmetrized scheme (S-OUCS4), for the solution of the flow past a three-dimensional cone-cylinder configuration at a free stream Mach number of 4. Governing equations are the full three-dimensional unsteady Navier–Stokes equations in strong conservation form as given below:

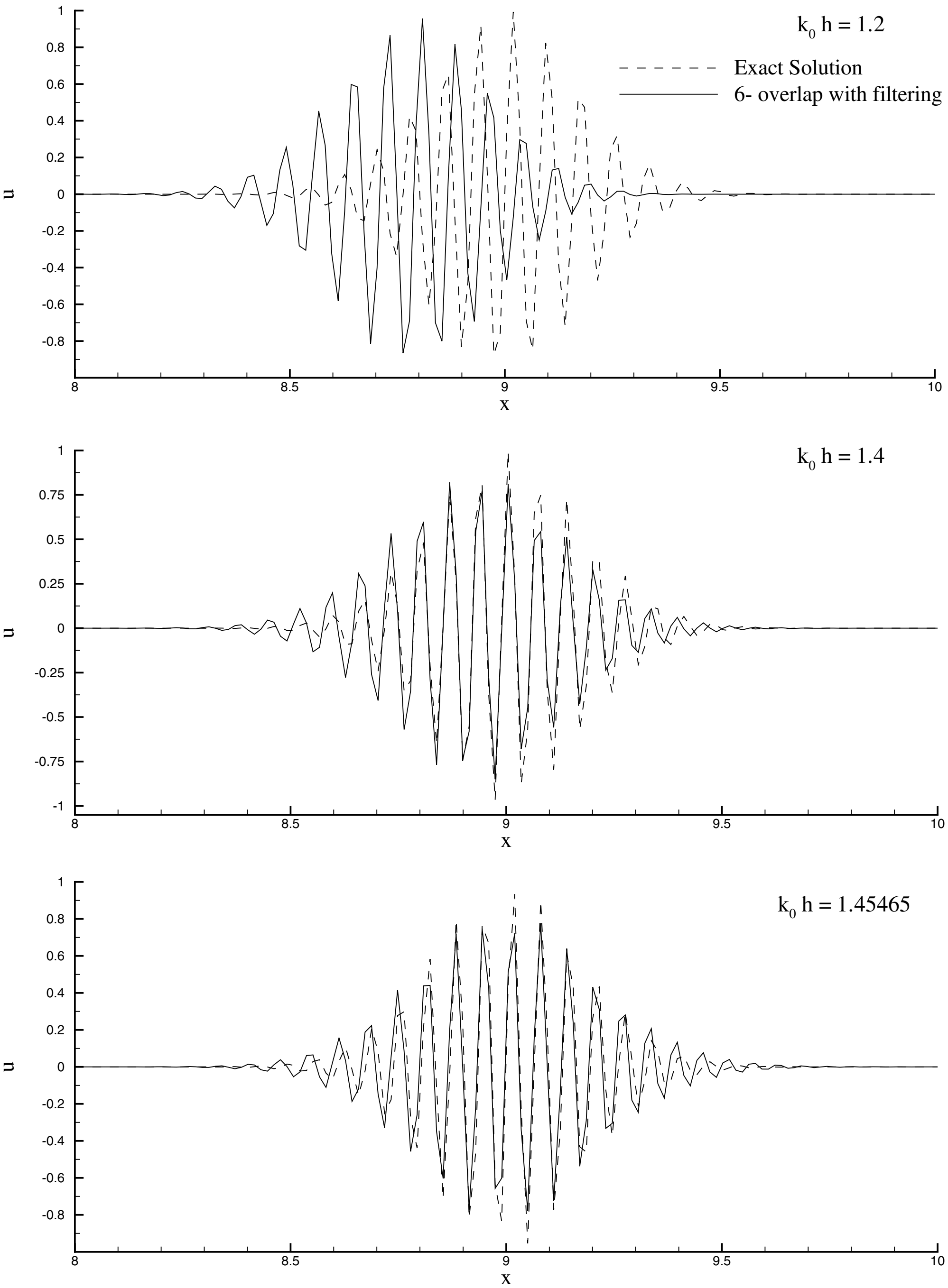

Fig. 7. Dispersion error due to the present parallel computing method is shown for the signals of decreasing wavelength at $t = 15$.

$$\frac{\partial \widehat{\mathbf{Q}}}{\partial t} + \frac{\partial \widehat{\mathbf{E}}}{\partial \xi} + \frac{\partial \widehat{\mathbf{F}}}{\partial \eta} + \frac{\partial \widehat{\mathbf{G}}}{\partial \zeta} = (1/Re)\left[\frac{\partial \widehat{\mathbf{E_v}}}{\partial \xi} + \frac{\partial \widehat{\mathbf{F_v}}}{\partial \eta} + \frac{\partial \widehat{\mathbf{G_v}}}{\partial \zeta}\right] \tag{19}$$

for the unknown state vectors, $\widehat{\mathbf{Q}} = \mathbf{Q}/J$, the inviscid and viscous flux vectors in generalized coordinates are represented as,

$$\begin{aligned}
\widehat{\mathbf{E}} &= (1/J)(\xi_x\mathbf{E} + \xi_y\mathbf{F} + \xi_z\mathbf{G})\\
\widehat{\mathbf{F}} &= (1/J)(\eta_x\mathbf{E} + \eta_y\mathbf{F} + \eta_z\mathbf{G})\\
\widehat{\mathbf{G}} &= (1/J)(\zeta_x\mathbf{E} + \zeta_y\mathbf{F} + \zeta_z\mathbf{G})\\
\widehat{\mathbf{E_v}} &= (1/J)(\xi_x\mathbf{E_v} + \xi_y\mathbf{F_v} + \xi_z\mathbf{G_v})\\
\widehat{\mathbf{F_v}} &= (1/J)(\eta_x\mathbf{E_v} + \eta_y\mathbf{F_v} + \eta_z\mathbf{G_v})\\
\widehat{\mathbf{G_v}} &= (1/J)(\zeta_x\mathbf{E_v} + \zeta_y\mathbf{F_v} + \zeta_z\mathbf{G_v})
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{Q} &= [\rho, \rho u, \rho v, \rho w, \rho e]^{\mathrm{T}}\\
\mathbf{E} &= [\rho u, \rho u^2 + p, \rho vu, \rho wu, (\rho e + p)u]^{\mathrm{T}}\\
\mathbf{F} &= [\rho v, \rho uv, \rho v^2 + p, \rho wv, (\rho e + p)v]^{\mathrm{T}}\\
\mathbf{G} &= [\rho w, \rho uw, \rho vw, \rho w^2 + p, (\rho e + p)w]^{\mathrm{T}}\\
\mathbf{E_v} &= [0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x]^{\mathrm{T}}\\
\mathbf{F_v} &= [0, \tau_{yx}, \tau_{yy}, \tau_{yz}, u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y]^{\mathrm{T}}\\
\mathbf{G_v} &= [0, \tau_{zx}, \tau_{zy}, \tau_{zz}, u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z]^{\mathrm{T}}
\end{aligned}$$

The various stress tensor components (τ_{ij}) and the heat flux components (q_i) can be further written as:

$$\begin{aligned}
\tau_{xx} &= \left(2\mu\frac{\partial u}{\partial x} + \lambda\nabla.\mathbf{V}\right)\\
\tau_{yy} &= \left(2\mu\frac{\partial u}{\partial y} + \lambda\nabla.\mathbf{V}\right)\\
\tau_{zz} &= \left(2\mu\frac{\partial u}{\partial z} + \lambda\nabla.\mathbf{V}\right)\\
\tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\\
\tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\\
\tau_{zx} &= \tau_{xz} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\\
q_x &= \frac{\mu}{(\gamma - 1)PrM_\infty^2}\frac{\partial T}{\partial x}\\
q_y &= \frac{\mu}{(\gamma - 1)PrM_\infty^2}\frac{\partial T}{\partial y}\\
q_z &= \frac{\mu}{(\gamma - 1)PrM_\infty^2}\frac{\partial T}{\partial z}
\end{aligned}$$

In the expressions above, u, v, w are the Cartesian velocity components, ρ is the density, p is the pressure and e is the total energy per unit mass defined as, $e = p/[(\gamma - 1)\rho] + (u^2 + v^2 + w^2)/2$. The perfect gas relationship $p = \rho T/(\gamma M_\infty^2)$ is also assumed; $\gamma = c_p/c_v$ is the ratio of specific heats and is taken as 1.4 in the present case. Here, μ is the dynamic viscosity and λ from the Stokes' hypothesis is $-2\mu/3$. The Jacobian of grid transformation is represented as J in the expressions above. All the flow variables have been normalized by their respective free stream values except for pressure, which has been non-dimensionalized by $\rho_\infty c_\infty^2$. Reynolds number is indicated as Re in Eq. (19), which is equal to 1.12×10^6 for the present problem, based on c_∞ and the cylinder diameter (D). Also, Pr is the Prandtl number in the above expressions and is equal to 0.7 in the present computation.

3.2. Grid system and computation procedure

The computational domain for the present problem is shown in Fig. 8. Cone angle is 9.46° in the present case. All the lengths shown in the figure are non-dimensionalized by the cylinder diameter (D). Rotation of the shown two-dimensional grid about the center line produces an axisymmetric three-dimensional grid with a total of 181 points in axial direction (x), 85 points in radial direction (r) and 45 points in azimuthal direction (θ). In the radial direction we have solved the problem using the hyperbolic-tangent stretched grid, as used in [14]. The minimum grid size close to the solid surface, as shown in Fig. 8, is $\Delta r = 0.0004D$. The grid used in the problem is orthogonal everywhere except over the cone surface. Numerical errors due to this non-orthogonality has been successfully controlled in the present problem by using the scalar dissipation model given in [18]. The computational domain is divided into 20 sub-domains in the axial direction using two point overlap, as shown in Fig. 3.

Based on the results of the last section, we have used the symmetrized compact scheme in the axial direction and OUCS4 in the other two directions, for the discretization of spatial derivatives for the convection terms. In the radial direction we have profitably used the bias of OUCS4 to damp out reflections from the lateral boundaries as it provides a buffer domain like behavior at lateral boundaries. In the azimuthal direction, interior stencil of OUCS4 is used as this is the periodic direction. Viscous terms have been discretized using second order central difference scheme. Time integration is performed by using four stage Runge–Kutta method, as given in the last section. It is noted that in an axisymmetric structured grid the Jacobian approaches infinity at the centerline, marked as AB in Fig. 8. Therefore, all of the grid metrics become indeterminate at the centerline, making it impossible to integrate the solution in time. However, the solutions (conservative variables) at the centerline can be interpolated from the neighboring values. In the present work a fourth order interpolation of [17] is used along AB. It can be written as,

$$f_0 = \frac{13(f_1 + f_{-1}) + 8(f_2 + f_{-2}) - 5(f_3 + f_{-3})}{32} \tag{20}$$

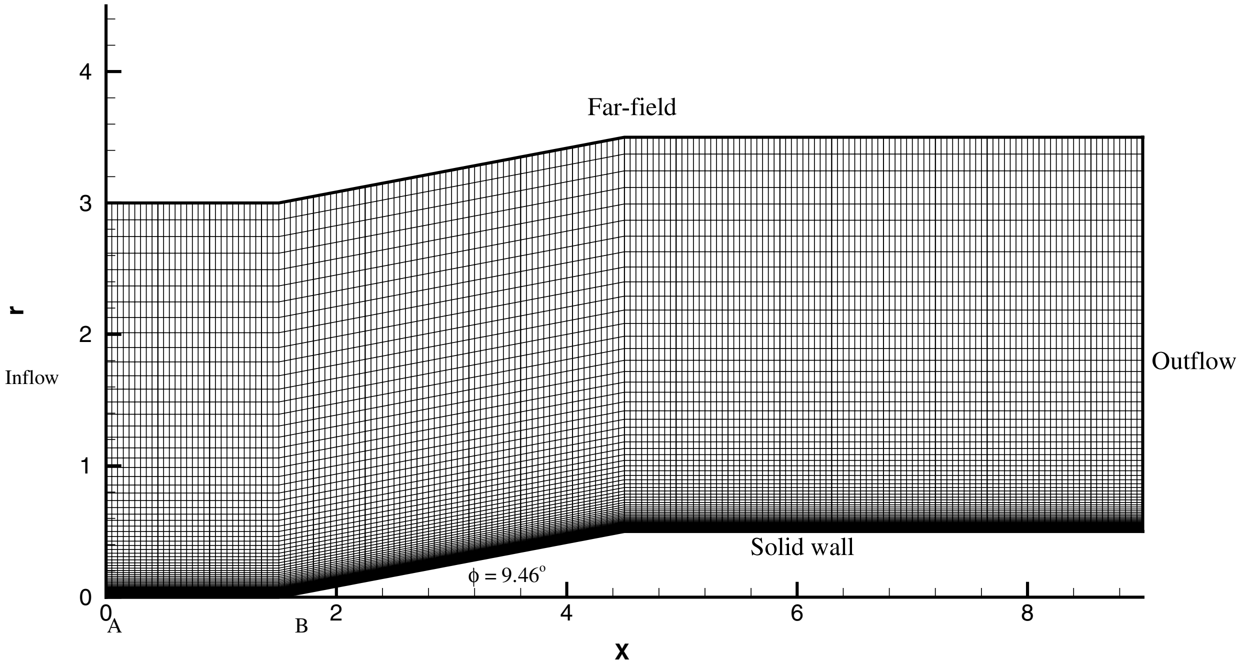

Fig. 8. Showing the grid arrangement for flow past a cone-cylinder configuration at $M_\infty = 4$. This arrangement was rotated about X-axis to generate the full 3D axi-symmetric grid. Number of points used are: $Nx = 181$, $Ny = 85$ and $N_\theta = 45$.

where the negative indices mean the values in the opposite direction across the centerline. This results in as many sets of interpolated values as half the number of grid points in the azimuthal direction. A unique value of the centerline solution is finally obtained by averaging these values.

High accuracy schemes in space and time resolve a wider range of wave number or frequency, as noted from Figs. 1 and 2. However, there are still some high wavenumber/frequencies which gives rise to spurious numerical oscillations due to dispersion – note the V_{gN}/c for $kh > 2.4$ in Fig. 2 for all interior nodes. This is true specially in a region where grid is non-orthogonal. To avoid such numerical errors from contaminating the result, a scalar dissipation model of [18] is used in the present computation, taking $\alpha_2 = 0.25$ and $\alpha_4 = 1/64$. This will also help us in getting rid of spurious reflections from the interface boundaries, as we experienced in the last section with the propagation of wave-packet. This artificial dissipation model is implemented only at the last stage of the Runge–Kutta time integration scheme in order to minimize the computational costs.

In addition to the reflection from the interfacial boundaries, there might be some reflection from the geometrical boundaries. Such reflection from the inflow was observed in Fig. 5(a), where the reflected wave had amplitude much higher than the incident one. Therefore, in addition to the requirement of these high accuracy schemes and artificial numerical dissipation, one must have a non-reflecting boundary condition to allow smooth passage to the incoming waves. To meet this requirement, generalized characteristic boundary condition of [16] is used in the present computation, for conditions at the solid wall. Free stream values have been used at all other boundaries of the computational domain, except at the outflow, where simple second order extrapolation [19] is used to generate a non-reflecting condition. Such simple conditions at all other boundaries are possible because of high free stream Mach number ($M_\infty = 4$) of the present problem. For $M_\infty > 1$ the characteristic convection speed of all the waves is unidirectional, i.e in the direction of the mean flow, whereas for $M_\infty < 1$, one is forced to use characteristic boundary condition at the inflow/outflow, in order to achieve a reflection-free condition. In the azimuthal direction periodic boundary condition is used for this axisymmetric geometry.

3.3. Results and discussion

Starting with a uniform flow condition everywhere, Eq. (19) is integrated in time following the computation procedure discussed in the previous subsection. The time step for the present computation is small ($\Delta t = 3 \times 10^{-5}$) so as to maintain a near-neutral behavior of the numerical schemes for long time integration [14]. The Mach number contours, hence obtained are shown in Fig. 9(a), at the indicated times. Contours are shown in the (r–x)-plane for $\theta = \pi/2$. No reflection is observed either at the interfacial boundaries or the outflow/inflow boundaries – as this is effectively controlled by selective addition of second and fourth order dissipation [18]. One notes the formation of an oblique shock at the vertex of the cone and an expansion fan at the cone-cylinder junction. Cross-sectional view of the contours are plotted in Fig. 9(b), at $x/D = 4.45$, marked by dotted lines in Fig. 9(a). One notes a perfectly symmetrical result at these cross-sections and no visible change in the contours, with time but there are changes in the contours very close to the surface that have not been plotted here.

To show the superiority of the symmetrized compact scheme (S-OUCS4) over the OUCS4 scheme in the axial direction, another case is run using OUCS4 in the axial direction as well, using two point overlap at the interface boundaries. Results obtained are very revealing in showing the difference between these two sets of calculations as shown in Fig. 10. In Fig. 10(a), difference in Mach number contours obtained by S-OUCS4 and OUCS4 schemes is shown at $t = 0.5$. Significant differences are noted near the cone vertex and the cone-cylinder junction. The magnitude of maximum difference is noted in all the frames of Fig. 10 for different quantities. The values differ in first decimal place, very close to the surface and keeps on reducing as we move up in radial direction, as one would expect from such high Mach number flow. Similar trends are seen in Fig. 10(b) and (c), where difference in pressure and density contours are shown at the same time. It is to be noted that same code is used for the two cases, just that S-OUCS4 is replaced by OUCS4 in the latter in axial direction. This leads us to conclude that these differences are only due to the near-boundary behavior of OUCS4, which develops a directional bias in the derivatives, causing more reflection from the interfacial boundaries. One can note this directionality of OUCS4 from the $|G|$ and V_{gN}/c contours shown in Figs. 1

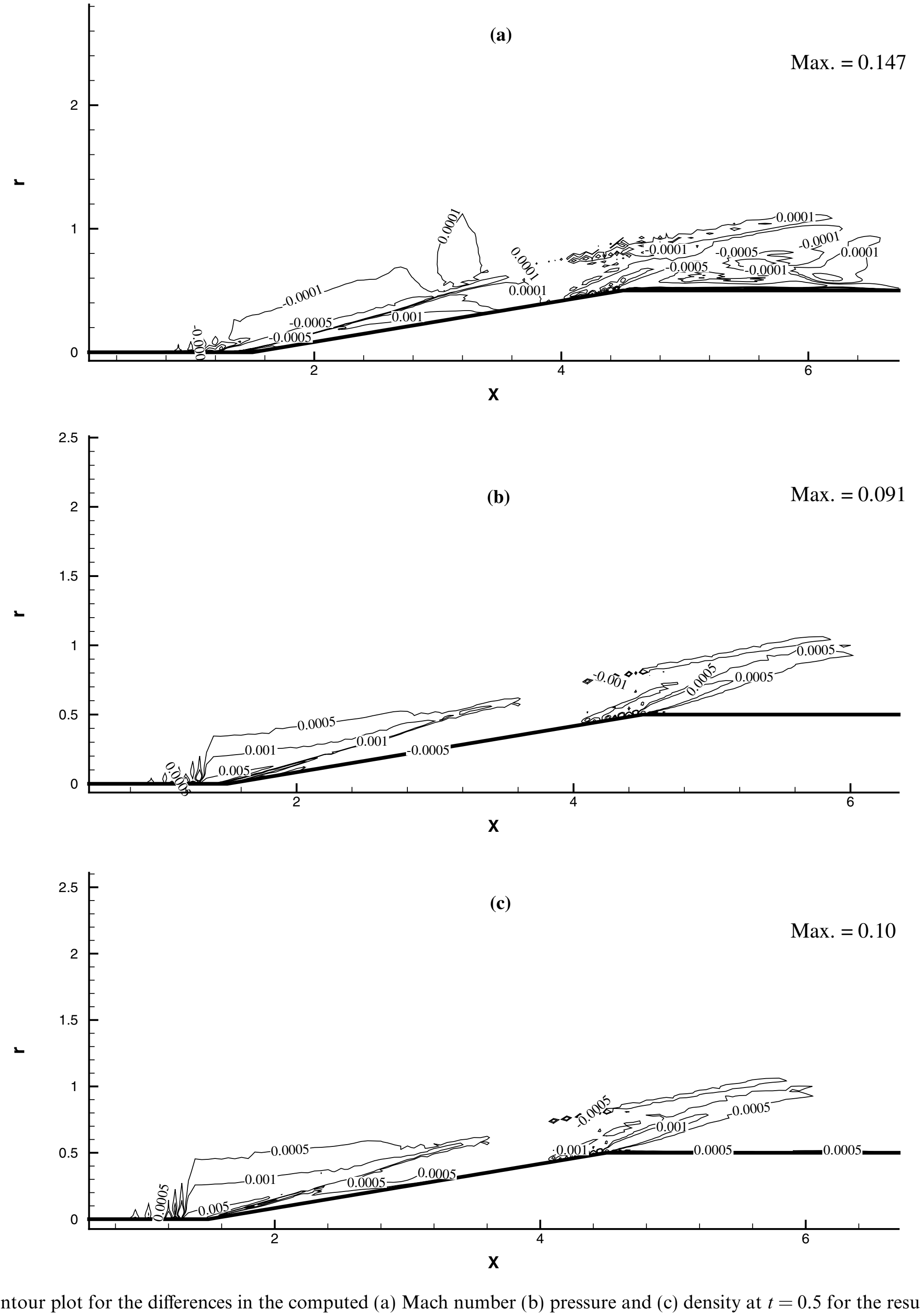

Fig. 10. Contour plot for the differences in the computed (a) Mach number (b) pressure and (c) density at $t = 0.5$ for the results obtained using OUCS4 and S-OUCS4 schemes. The maximum value of the differences, in magnitude, is shown in each frame.

everywhere, except at few locations. These locations are, at the cone vertex and the junction of the cone and cylinder. Reason for jumps in $\Delta c_p/c_p$ values at these two junction is well understood as, one of them correspond to the shock location and the another one correspond to the expansion fan. But there are few more jumps near $x = 6.5$–7.0, reason for which is not clear. This could be because of the interaction of the oblique

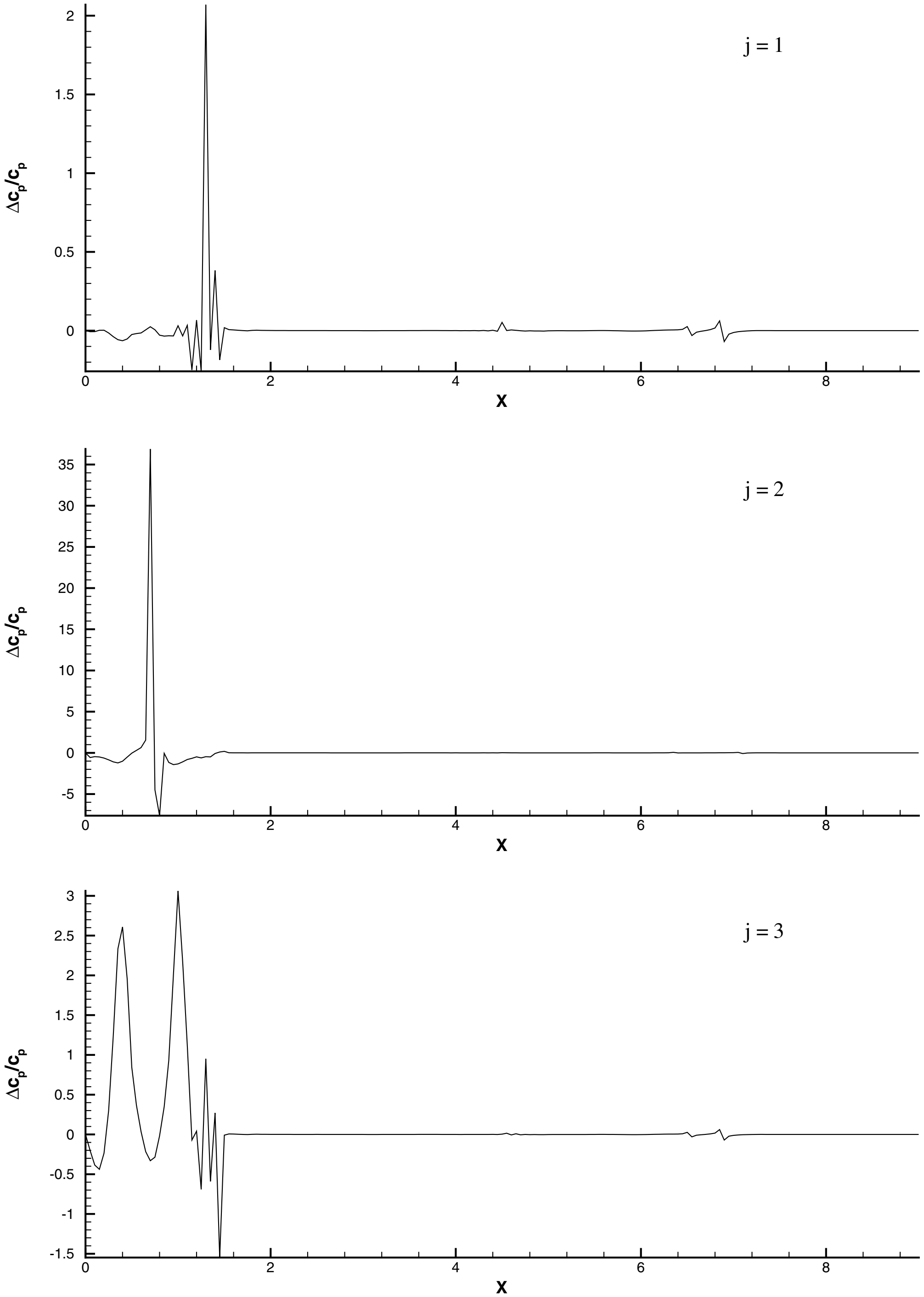

Fig. 11. Relative difference in pressure coefficient (c_p) between the OUCS4 and S-OUCS4 scheme computations is shown along the streamwise direction, at the indicated j level from the solid surface, with $j = 1$ as the wall itself.

shock with the shear layer, that requires further analysis. Similar patterns are also seen in the other two frames of Fig. 11, where $\Delta c_p/c_p$ is plotted for $j = 2$ and $j = 3$. One notices a much larger difference at $j = 2$, while at $j = 3$ the values are of the same order as that observed at $j = 1$. At $j = 3$, we have large values of $\Delta c_p/c_p$ from $x = 0$ to $x = 1.8$, which is not there in other two figures.

Overall, the calculated results using S-OUCS4 scheme are seen to be adequate in capturing the equilibrium laminar flow, whose receptivity to imposed acoustic, vortical and entropic disturbances will be studied in future.

4. Convection of a shielded vortex in subsonic flows

Finally, the case for the convection of a shielded vortex is studied at low subsonic speed to check whether the present method of domain decomposition extends to elliptic cases or not. This allows us to estimate the error committed in the multiprocessor computing with respect the serial calculation of the same. In a shielded vortex, the core is surrounded by an annular ring with vorticity of opposite sign. The shielded vortex studied here correspond to the two-parameter inviscid distributed vortex given by,

$$\omega = K(1-\alpha_1 r^2)e^{-\alpha_2 r^2} \tag{21}$$

This vortex can have non-zero circulation, depending upon the values of α_1 and α_2, as given by,

$$\Gamma = \frac{\pi K}{\alpha_2}(1-\alpha_1/\alpha_2) \tag{22}$$

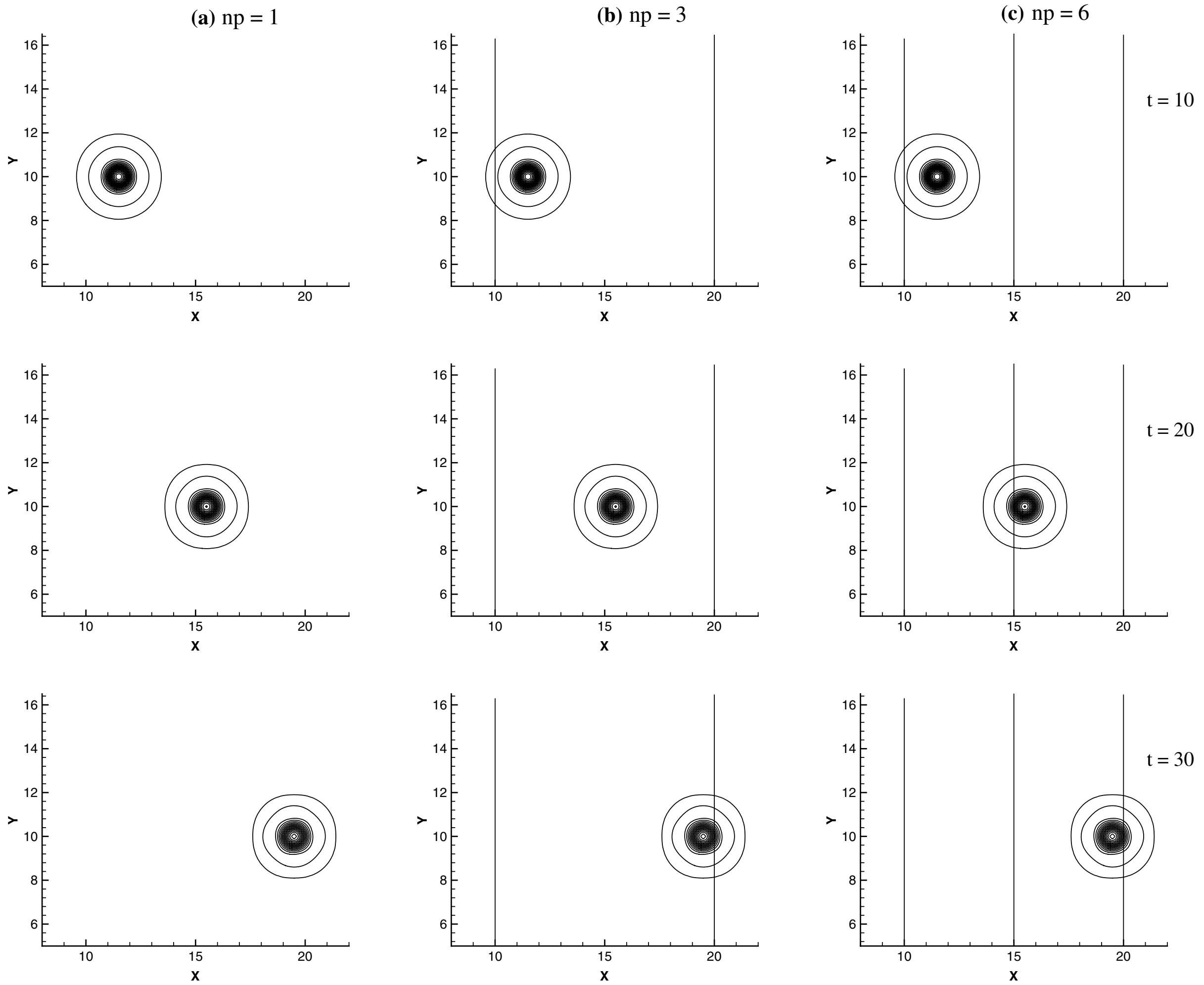

Fig. 12. Euler solution obtained for the convection of a shielded-vortex at the indicated times using (a) sequential; (b) 3 nodes parallel and (c) 6 nodes parallel arrangement of grids. The vertical lines indicate the location of the inter-processor boundary.

In the present work, we have used $\alpha_1 = \alpha_2 = 1/2$, that gives rise to a vortex with zero circulation and that creates the velocity and pressure distribution given in Section 4.1.1. of Ref. [15]. The vortex of Eq. (21) also is characterized by the presence of an inflexion point for the velocity distribution i.e. $\frac{\partial\omega}{\partial r} = 0$. It is well known that such velocity profiles are very susceptible to inviscid instabilities – as given in [20] as a necessary condition for the centrifugal instability for two-dimensional non-axisymmetric disturbances. This instability is physical in nature, as it was studied in [21] where the mono-polar vortex gave rise to a tripolar structure upon the application of small amplitude disturbances, that could also be triggered by numerical error.

In the following, we have performed calculations for a case, where a vortex of non-dimensional strength equal to 0.02 convects in a free stream at the Mach number $M_\infty = 0.4$. Subsonic cases are more sensitive to the boundary closures than the supersonic case because for $M_\infty < 1$ the flow is elliptic in nature and the waves are not unidirectional, as in the case of $M_\infty > 1$. For this reason, the flow in this case is more susceptible

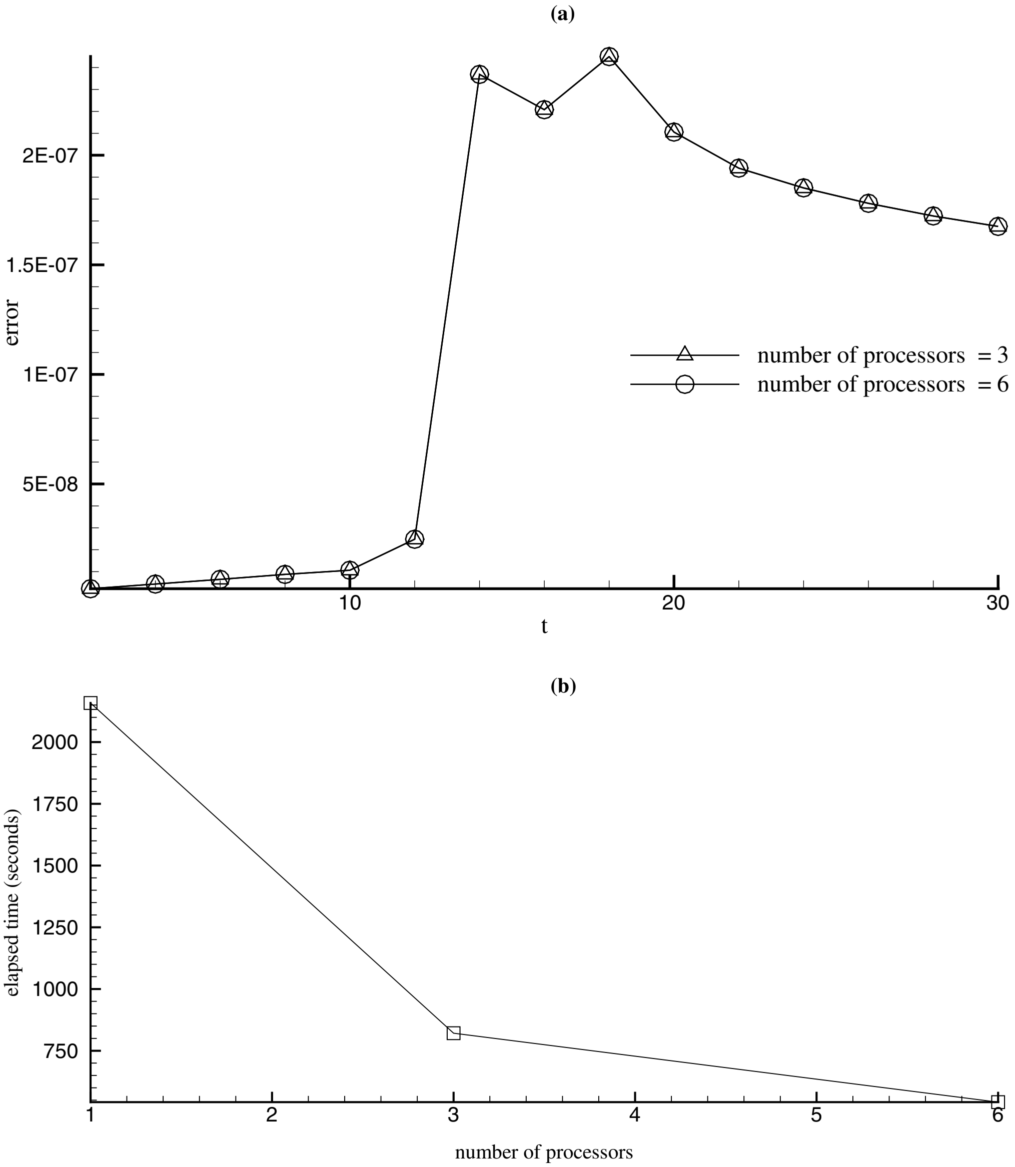

Fig. 13. (a) Time variation of the L_2 norm of the error due to parallel computing using the present domain decomposition technique is shown for the test case of Section 4. (b) Shows the efficiency of the present parallel computing method with respect to the elapsed time (in seconds) and number of processors.

to reflection at the interface boundaries. The computational domain for the present case varies from $0 \leqslant x \leqslant 30$ and $0 \leqslant y \leqslant 20$ with 301 points in the x direction and 201 points in the y direction such that $\Delta x = \Delta y = 0.1$. At $t = 0$, the vortex is centered at $x = 7.5$ and the computation is carried out till the non-dimensional time of $t = 30$. It is to be noted that the velocity scale for this problem is the speed of sound and hence the non-dimensional convection speed of the vortex is $U_\infty = 0.4$. In this section, we have used the symmetrized scheme (S-OUCS3) developed in [14] to show the identical beneficial effects of symmetrization of any compact scheme. Also, explicit fourth order numerical dissipation is used to suppress high frequency reflection from the inter-domain boundaries. Slight use of numerical damping is necessary to see the best performance of the symmetrized compact schemes using the present parallel computing method, as there will always be some high frequency non-zero reflections from the inter-domain boundaries, that when left uncontrolled, can contaminate the solution everywhere [22]. Time dependent characteristics based boundary conditions are used on density, at all the geometrical boundaries to avoid reflection from them.

Three different cases have been performed using 1, 3 and 6 processors splitting the problem in the streamwise (x) direction with six-point overlap strategy. The number of grid points are divided equally among the processors to make a perfect load balance. The first case with single processor, corresponds to serial computing and is undertaken to measure the error of parallel computing with respect to this serial benchmark. The results in Fig. 12, show the vorticity contours at different time instants for the three cases. The vertical lines in these figures, indicate the locations of the inter-domain/inter-processor boundaries. Comparison of Figs. 12(b) and (c) with Fig. 12(a) clearly shows that the results of parallel computations match perfectly with the sequential results and the error induced by reflections from the inter-domain boundaries are negligible and can not be observed visually. A quantitative measure of the error ($u_{\text{sequential}} - u_{\text{parallel}}$), in parallel computation framework, is shown by the time variation of the L_2 norm of the error in Fig. 13(a). The errors are negligibly small, of the order of 10^{-7}. It is also noted that the error remains the same when the number of processors increases.

In Fig. 13(b), elapsed time (in seconds) of these cases are plotted against the number of processors in order to show the parallel efficiency of the present method. The figure shows that the parallel efficiency keeps coming down with increase in number of processors. This is due to the larger number of information transfer across the inter-domain boundaries, associated with six-point overlap method used in this test case.

5. Summary

Compact schemes have been specifically designed for parallel computing using domain decomposition method. While retaining the excellent numerical properties of compact schemes for DNS, the streamwise bias of these schemes have been reduced significantly by symmetrization so that the resultant scheme can be used for parallel domain decomposition methods with minimal overlap. The remaining problems of parallel computing using compact schemes are removed by adopting any one of the following: (a) larger overlap of sub-domains, with and without filtering [15] of the solution; (b) reducing the overlap, with mandatory filtering of the solution in sub-domains and (c) selective addition of artificial second and fourth order dissipation [18]. The developed procedures for (a) and (b) have been calibrated for the propagation of a wave-packet following linear convection equation. We have performed detailed error analysis identifying the contributing sources and showing that the six-point overlap is more accurate than the two-point overlap domain-decomposition method. Procedures adopted for (c) has been shown by solving 3D unsteady Navier–Stokes equations for $Re = 1.12 \times 10^6$, $M_\infty = 4$ flow past a cone-cylinder.

References

[1] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, UK, 1999.

[2] B.F. Smith, P. Bjorstad, W. Gropp, Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, New York, 1996.

[3] V. Dolean, S. Lanteri, F.C. Nataf, Int. J. Num. Meth. Fluids 40 (2002) 1539.

[4] P.L. Lions, in: T.F. Chan, R. Glowinski, J. Periaux, O. Widlund (Eds.), Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1990.

[5] J.S. Shang, J.A. Camberos, M.D. White, Advances in time-domain computational electromagnetics, AIAA Paper no., 99-3731, 1999..

[6] D.P. Lockard, P.J. Morris, J. Comput. Acoust. 5 (1997) 337.

[7] A. Povitsky, P.J. Morris, ICASE Report No. 99-34, 1999.

[8] S.K. Lele, J. Comput. Phys. 103 (1992) 16.

[9] T.K. Sengupta, G. Ganeriwal, S. De, J. Comput. Phys. 192 (2) (2003) 677.

[10] T.K. Sengupta, S.K. Sircar, A. Dipankar, J. Sci. Comput. 26 (2) (2006) 151.

[11] J.J. Dongarra, I.S. Duff, D.C. Sorenson, H.A. Van der Vorst, Numerical Linear Algebra for High Performance Computers, SIAM, Philadelphia, 1998.

[12] J. Hofhaus, E.F. Van de Velde, SIAM J. Sci. Comput. 17 (2) (1996) 454.

[13] X.-H. Sun, Parallel Comput. 21 (8) (1995) 1241.

[14] A. Dipankar, T.K. Sengupta, J. Comput. Phys. 215 (2006) 245.

[15] M.R. Visval, D. Gaitonde, J. Comput. Phys. 181 (2002) 155.

[16] J.W. Kim, J.D. Lee, AIAA J. 42 (1) (2004) 47.

[17] J.W. Kim, P.J. Morris, AIAA J. 40 (10) (2002) 1961.

[18] A. Jameson, W. Schmidt, E. Turkel, AIAA Paper 81 (1981) 1259.

[19] T.H. Pulliam, J.L. Steger, AIAA J. 130 (2) (1980) 159.

[20] P.G. Drazin, W.H. Reid, Hydrodynamic Stability, Cambridge University Press, Cambridge, UK, 1981.

[21] X.J. Carton, G.R. Flierl, L.M. Polvani, Europhys. Lett. 9 (4) (1989) 339.

[22] R. VichnevetskyMathematics and Computers in Simulation, vol. XXIII, North Holland Publishing Co., 1981.