Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 210 (2005) 459–496

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

High-order compact finite-difference methods on general overset grids

Scott E. Sherer [a,*], James N. Scott [b]

[a] Computational Sciences Branch, Air Vehicles Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, United States

[b] Department of Aerospace Engineering and Aviation, The Ohio State University, Columbus, OH 43210, United States

Received 28 September 2004; received in revised form 25 April 2005; accepted 27 April 2005
Available online 22 June 2005

Abstract

This work investigates the coupling of a very high-order finite-difference algorithm for the solution of conservation laws on general curvilinear meshes with overset-grid techniques originally developed to address complex geometric configurations. The solver portion of the algorithm is based on Padé-type compact finite-differences of up to sixth-order, with up to 10th-order filters employed to remove spurious waves generated by grid non-uniformities, boundary conditions and flow non-linearities. The overset-grid approach is utilized as both a domain-decomposition paradigm for implementation of the algorithm on massively parallel machines and as a means for handling geometric complexity in the computational domain. Two key features have been implemented in the current work; the ability of the high-order algorithm to accommodate holes cut in grids by the overset-grid approach, and the use of high-order interpolation at non-coincident grid overlaps. Several high-order/high-accuracy interpolation methods were considered, and a high-order, explicit, non-optimized Lagrangian method was found to be the most accurate and robust for this application. Several two-dimensional benchmark problems were examined to validate the interpolation methods and the overall algorithm. These included grid-to-grid interpolation of analytic test functions, the inviscid convection of a vortex, laminar flow over single- and double-cylinder configurations, and the scattering of acoustic waves from one- and three-cylinder configurations. The employment of the overset-grid techniques, coupled with high-order interpolation at overset boundaries, was found to be an effective way of employing the high-order algorithm for more complex geometries than was previously possible.

PACS: 02.70.Bf; 47.11.+j

Keywords: High-order methods; Overset-grid methods; Interpolation; Computational fluid dynamics

[*] Corresponding author.
E-mail address: scott.sherer@wpafb.af.mil (S.E. Sherer).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.04.017

1. Introduction

The use of high-order numerical methods [1–3] is a recognized technique to circumvent the limitations of standard low-order algorithms for the simulation of various physical phenomena, including turbulence, aeroacoustics, electromagnetics, and fluid/structure interactions. Due to their superior ability to accurately represent waves at high-frequencies and/or with limited grid support, high-order methods can produce accurate results on coarser computational grids compared to low-order techniques, thus resulting in a decrease in the overall computational effort. However, most early efforts (see, for example, [2,4,5]) involving high-order methods have employed simple domains and Cartesian grids for a variety of reasons, such as the limited flexibility of spectral methods and the lack of developed curvilinear-grid techniques for other types of high-order approaches. Thus, situations where high-order methods could be applied were typically limited to fundamental research where canonical problems could be employed.

Overset-grid methods [6–8] have also developed into a powerful tool for complex numerical simulations. In the overset-grid approach, the computational domain is subdivided into overlapping structured-grid components, thus reducing a single, complex domain into a series of smaller, potentially simpler ones. The governing partial differential equations are solved separately on each component, and domain connectivity is obtained through the use of interpolation along the boundaries between overlapping grids. While originally developed as a means to address complex geometries (e.g., [9,10]), overset-grid methods have also been employed to simulate multiple bodies in relative motion (e.g., [11]) and provide flexible grid-adaptation through localized refinement/coarsening in a computational domain (e.g., [12,13]). The overset-grid paradigm may also serve as a domain-decomposition paradigm for exploiting massively parallel computing platforms [14]. Finally, the overset-grid approach is conducive to performing computational simulations in a production environment, where component grids for a complex system may be developed concurrently by different team members, libraries of common grid components may be developed for reusability, and small changes may be quickly incorporated into a system by modifying only the impacted component grids and not the entire grid system [15].

With their "spectral-like" resolution [2] and increased geometric flexibility compared to spectral methods, compact algorithms are an attractive approach to provide high-order accuracy. In particular, a high-order, compact, finite-difference algorithm was developed for use on three-dimensional, curvilinear, structured, single-grid topologies [16]. The algorithm has since been enhanced and validated for more complex grid topologies. The algorithm was extended to overset-grid topologies in the limited case where the overlaps between grids consisted exclusively of coincident grid points through the use of high-order one-sided formulations at block boundaries [17]. Because of the requirement of point-to-point matching in the overlap region, data could be directly injected between grid points, and thus no interpolation was required. The performance of the high-order approach for various complex topologies such as those that might be encountered in more realistic geometries has also been investigated. These include smoothly varying but significantly skewed and stretched grids [17], as well as discontinuous jumps in grid-spacing, abrupt grid skewing, extreme grid distortion, and moving/deforming grids [18].

In the current work, this high-order compact algorithm is extended to general overset-grid topologies. These topologies possess overlap regions between grids that contain non-coincident grid points. These topologies may also possess hole points, which are points that are "blanked out" from consideration in the solution [6]. These are common situations encountered when using overset-grid methods to simulate complex geometries, and their incorporation greatly enhances the range of problems that may be addressed with the algorithm. The use of overset-grid methods coupled with a high-order, structured-grid algorithm has been previously demonstrated by Yin and Delfs [19] and Delfs [20]. In these works, an explicit dispersion relation preserving (DRP) seven-point finite-difference scheme was utilized along with higher-order interpolation methods at the interfaces to solve benchmark problems in computational aeroacoustics. Because of the general, non-coincident nature of the overlap regions between grids, some form of interpolation will be necessary to allow the grids to communicate with each other. Delfs [20] demonstrated that the use of linear interpolation, which is typically used in second-order overset-grid solvers [21], eventually results in the corruption of the entire solution domain when used in conjunction with high-order algorithms despite the fact that the error is generated locally at the grid interfaces. Similarly, Chesshire and Henshaw [22] demonstrated that linear interpolation is unable to maintain the overall global accuracy of an overset-grid method at the same level as the higher-order discretization formulae. Thus, it is anticipated that some form of high-order interpolation will be required to fully realize the benefits of the high-order compact solver. The ability to manage the presence of hole points in grids is also considered, which leads to greater overall flexibility of the approach when dealing with solid bodies embedded in background grids.

Candidate interpolation methods are considered for inclusion into the high-order overset-grid algorithm and are introduced in Section 2, with detailed development of the approaches left for Appendices A and B. The methods considered here are a generalized Lagrangian interpolation approach that allows for both explicit and compact interpolation as well as non-optimized and optimized interpolation, and an interpolation method based on basis-splines (B-splines). A one-dimensional Fourier error analysis is performed on these methods in Section 3 to compare and contrast their accuracy and spectral behavior. The remainder of the work is focused on implementation of the interpolation and hole-cutting methods into the high-order algorithm (Section 4) and some of the validation studies that have been performed using the high-order overset-grid algorithm (Section 5). These cases include the interpolation of various analytic test functions from one grid to another, and the two-dimensional fluid dynamics problems of the convection of an inviscid vortex, the low-speed, laminar flow over one and two circular cylinders, and the scattering of acoustic waves from single- and three-cylinder configurations. While all of the fluid dynamics validation cases presented here are two-dimensional in nature, the method is completely extendable to three spatial dimensions as is demonstrated in [23].

2. Development of interpolation methods

A model one-dimensional interpolation stencil is shown in Fig. 1 and consists of N uniformly spaced donor points, $N-1$ subintervals defined as the regions between adjacent stencil points, and J

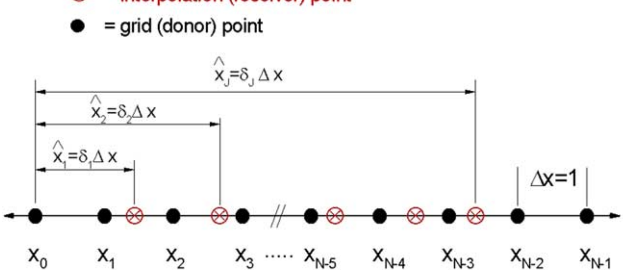
⊗ = interpolation (receiver) point
● = grid (donor) point
$\hat{x}_J=\delta_J\Delta x$
$\hat{x}_2=\delta_2\Delta x$
$\hat{x}_1=\delta_1\Delta x$
$\Delta x=1$
x_0
x_1
x_2
x_3 ·····
x_{N-5}
x_{N-4}
x_{N-3}
x_{N-2}
x_{N-1}

Fig. 1. Model one-dimensional interpolation stencil containing J interpolation points and N grid points.

arbitrarily distributed interpolation points. In an overset-grid application, the stencil points correspond to grid points from a donor grid where flowfield data is known, while the interpolation points correspond to grid points from a receiver grid where flowfield data is being updated. The location of the interpolation points within a stencil is completely arbitrary, and there is no a priori knowledge as to how the interpolation points are distributed. However, it is assumed without a loss of generality that the grid point spacing Δx is uniform and equal to unity ($\Delta x = 1$), which is justified by specifying the interpolation be performed in the transformed computational space of the donating grid rather than in physical space.

Using the notation of Fig. 1, a general, one-dimensional interpolation formula may be written as

$$\sum_{k=1}^{\mathrm{BW}} L_k \hat{f}(\hat{x}_k) = \sum_{j=0}^{N-1} R_j f(x_j), \tag{1}$$

where $\hat{f}(\hat{x}_k)$ is the unknown value of the discrete function being interpolated at the receiver point $\hat{x}_k = \delta_k \Delta x = \delta_k$ and $f(x_j)$ is the known value of the discrete function at the stencil point $x_j = j\Delta x = j$. Depending upon the amount of overlap between donor and receiver grids, it is possible for a point to be donating data to a grid while also receiving interpolated data from the same or a different grid. This situation is referred to in the overset-grid literature as *implicit* interpolation [22] due to the fact that a system of coupled equations must be solved in order to obtain the interpolated data. An interpolation method that does not possess this coupling between receiver points of different component grids is commonly called *explicit* interpolation. Here, it will be assumed that adequate overlap exists between component grids, and thus there is no coupling of donor and receiver points across grids. However, as discussed in the subsequent paragraph, the coupling of interpolation points from a single component grid will be considered. Further reference to an explicit or implicit (compact) interpolation method refers exclusively to this type of interpolation point coupling rather than the coupling of interpolation points across multiple component grids.

The coefficients L_k and R_j in Eq. (1) are interpolation coefficients, which may be found from a variety of methods as discussed subsequently. The quantities δ_k are referred to as the interpolation offsets, and are referenced from the stencil base point x_0 as shown in Fig. 1. The parameter BW dictates the level of coupling between interpolation points on the same component grid by controlling the number of unknown functional values present in each application of Eq. (1). A value of BW = 1 yields one unknown per equation, thus resulting in an explicit method where the interpolation points are decoupled from each other. Values of BW > 1 yield multiple unknowns per equation, and thus a system of linear algebraic equations is generated which couple the interpolation points from a single donor grid. A tridiagonal system of equations must be solved to obtain the interpolated functional values when BW = 3, while a pentadiagonal system results when BW = 5. Here, only tridiagonal systems will be considered, and thus subsequent references to compact methods refers exclusively to the BW = 3 case.

The error inherent in Eq. (1) may be quantified through the use of a one-dimensional Fourier error analysis [24]. Manipulation of the spectral nature of the error associated with this general interpolation formula also provides the foundation for the optimized interpolation methods being considered. The error analysis employed here follows the work of Tam and Kurbatskii [25], and the reader is referred there for more detail. By inserting a single Fourier component of the form $f(x) = \exp(\imath\alpha x)$ into the general interpolation formula of Eq. (1) and noting that $\Delta x = 1$, the local error as a function of the wavenumber α may be defined as

$$E_{\mathrm{local}} = \left| \sum_{k=1}^{\mathrm{BW}} L_k \mathrm{e}^{\imath\delta_k\alpha} - \sum_{j=0}^{N-1} R_j \mathrm{e}^{\imath j\alpha} \right|^2. \tag{2}$$

An integrated error may also be defined by integrating Eq. (2) over some fractional range R of the wavespace $0 \leqslant \alpha \leqslant \pi$ to yield

$$E_{\text{int}} = \int_0^{R\pi} \left| \sum_{k=1}^{\text{BW}} L_k \text{e}^{\imath \delta_k \alpha} - \sum_{j=0}^{N-1} R_j \text{e}^{\imath j \alpha} \right|^2 \text{d}\alpha \quad (0 \leqslant R \leqslant 1). \tag{3}$$

Because the local and integrated errors are functions of L_k and R_j, inserting the interpolation coefficients obtained with various interpolation methods into the above error expressions provides a means to quantitatively compare various interpolation methods.

2.1. Generalized Lagrangian methods

The first family of methods considered is a generalized Lagrangian approach. This is a generalization of the classic Lagrangian interpolation method, which is an explicit (BW = 1) method that generates a single interpolating polynomial of degree $N - 1$ valid anywhere within the N-point stencil. The formal order-of-accuracy, denoted by σ, of classic Lagrangian interpolation is given by $\sigma = N$, as its coefficients satisfy the first N order-of-accuracy constraint equations arising from the Taylor series expansion of Eq. (1). In the generalized Lagrangian approach, this classic method is extended by considering methods where BW > 1 as well as to the situation where alternative constraint equations are employed in place of the usual order-of-accuracy constraints. The maximum formal order-of-accuracy that may be obtained with Eq. (1) using a BW-wide method on an N-point stencil is given by

$$\sigma_{\max} = N + \text{BW} - 1. \tag{4}$$

However, if other types of constraints are used to determine the interpolation coefficients instead of constraints based solely on the formal order-of-accuracy of the method, then the spectral behavior of the method may be modified.

In Lele's seminal work on compact methods [2], high-order interpolation was considered, but was limited to the case of mid-point interpolation. In addition, a fourth-order, compact interpolation scheme was developed with "spectral-like" resolution by replacing formal order-of-accuracy requirements with ad hoc equations constraining the spectral response of the scheme. A more general approach was employed by Tam and Kurbatskii [25] for arbitrary interpolation offsets, and was used to generate interpolation coefficients that resulted in minimum error over a specified portion of Fourier wavespace for an explicit, formally first-order accurate method. Here, Tam's approach is used and extended to the more general cases of BW $\geqslant$ 1 and variable order-of-accuracy. Constraints may also be employed to force certain interpolation coefficients to take on specified values, but these cases will not be considered here (see [26] for details on the formulation of such cases).

In [25], optimized interpolation coefficients were generated by minimizing the integrated error defined in Eq. (3) over some portion of wavespace controlled by the optimization parameter r. Thus, an optimized error may be defined as

$$E_{\text{opt}} = \int_0^{r\pi} \left| \sum_{k=1}^{\text{BW}} L_k \text{e}^{\imath \delta_k \alpha} - \sum_{j=0}^{N-1} R_j \text{e}^{\imath j \alpha} \right|^2 \text{d}\alpha \tag{5}$$

and differs from the integrated error given by Eq. (3) only by the range of wavespace over which the integration is performed. The above equation is minimized using a Lagrange multiplier method subject to constraints on the formal order-of-accuracy of the resulting interpolation equations. This approach, detailed in Appendix A, results in a linear system of equations that when solved generates the interpolation coefficients R_j and L_k. When these coefficients are obtained using the maximum number of order-of-accuracy constraints ($\sigma = \sigma_{\max}$), no minimization of the integrated error will take place, and the method will be referred to as non-optimized. Similarly, a fully optimized method will refer to the situation where only the first-order-accurate constraint equation is applied ($\sigma = 1$, the minimum order-of-accuracy considered here). In this situation, the method will produce the smallest integrated error over the range of wavespace given by $0 \leqslant \alpha \leqslant r\pi$ for a given stencil and bandwidth. Partial optimization ($1 < \sigma < \sigma_{\max}$) also is a possibility, and while briefly discussed in [26], it is not considered here.

2.2. Basis-spline methods

The second general interpolation method under consideration is basis-spline, or B-spline, interpolation. The well-known cubic spline interpolation technique is a subset of B-spline interpolation. B-splines have been used as basis functions in the solution of systems of partial differential equations related to fluid dynamics [27–30] and recently for high-order interpolation and differentiation [31]. The theoretical underpinnings of B-splines are described in detail by de Boor [32], and a summary is presented in Appendix B.

A set of B-splines of order σ spans the vector space consisting of all piecewise-continuous polynomials of degree $n = \sigma - 1$ for some knot point sequence. As discussed in Appendix B, a knot point sequence may consist of repeated values of break points, which are the points where the piecewise interpolating polynomial changes form. The duplication of break points in the knot point sequence governs the number of continuous derivatives possessed by the piecewise polynomial across the repeated break point. Here, only B-splines possessing maximum continuity across the interior break points ($C^{\sigma-2}$ for a B-spline of order σ) will be considered. Thus, all interior knot points will have a multiplicity of one; i.e., there are no repeated interior break points. The reader is referred to [32] for further information regarding the relationship between knot multiplicity and the continuity of the piecewise polynomial.

For overset-grid applications, each break point corresponds to a single grid point from a donating grid. Even though one of the strengths of a B-spline approach is that it may be used directly on a non-uniform mesh [33], it will be assumed here that the interpolation is being performed in computational space with a uniform grid spacing of unity. Because of its piecewise nature, the degree of the interpolating function generated using B-splines is independent of the number of grid points that make up the donating stencil and instead depends on the order of the B-spline. This differs from the non-optimized Lagrangian method, where the degree of the single interpolating polynomial defined over the entire stencil depends directly on the size of the stencil. Thus, the stencil size N will in general not be an important parameter in the analysis of B-spline interpolation, except as it relates to boundary condition effects.

To generate the piecewise interpolation polynomial for the B-spline method, the collocation procedure detailed in Appendix B is employed to generate a piecewise interpolating polynomial that reproduces the known discrete functional value at each stencil point. The general interpolation equation can be written for a single interpolation point $\hat{x}$ using Eq. (1), with $L_1 = 1$ and R_j given by Eq. (B.19) from Appendix B,

$$R_j = \sum_{i=I}^{I+\sigma-1} [\beta]^*_{i,j+1} B_{i,\sigma}(\hat{x}). \tag{6}$$

As discussed in Appendix B, the matrix $[\beta]^*$ depends primarily on the order of the B-splines and the boundary conditions employed to complete the linear system of equations. The only boundary condition employed here is the natural-spline condition, where certain derivatives at each end of the stencil are set equal to zero. Other boundary conditions exist, including one-sided finite-difference boundary conditions, where the derivatives are approximated using one-sided finite-differences, and the not-a-knot condition [32], where knot points are removed from consideration in order to obtain a consistent system of equations. These alternate boundary conditions are not examined in the current work.

3. One-dimensional error analysis

Before discussing results from the one-dimensional Fourier error analysis, an additional parameter is defined as

$$\delta_C = \left|\delta_p - \left(\frac{x_{N-1} - x_0}{2}\right)\right|, \tag{7}$$

where δ_p is the offset associated with the lone interpolation point for an explicit or B-spline method, or the middle interpolation point of a BW-point grouping for a compact method. This metric indicates how centered the interpolation point(s) under consideration are within their respective stencil. Non-zero values of δ_C indicate some degree of one-sidedness to the stencil, while $\delta_C = 0$ represents an perfectly centered stencil.

3.1. Local error analysis

The spectral behavior of the interpolation error associated with a particular method may be examined by plotting the local error from Eq. 2 as a function of the wavenumber α. Example spectral profiles with $\delta_C = 0$ are shown in Fig. 2 for Lagrangian methods and Fig. 3 for B-spline methods. The improvement in the accuracy of the non-optimized explicit interpolation methods as the stencil size (and thus order-of-accuracy) increases is clearly observed from Fig. 2. In particular, second-order explicit interpolation, which is the typical approach used in low-order overset-grid algorithms, exhibits a much larger local error for $\alpha < 1$. The

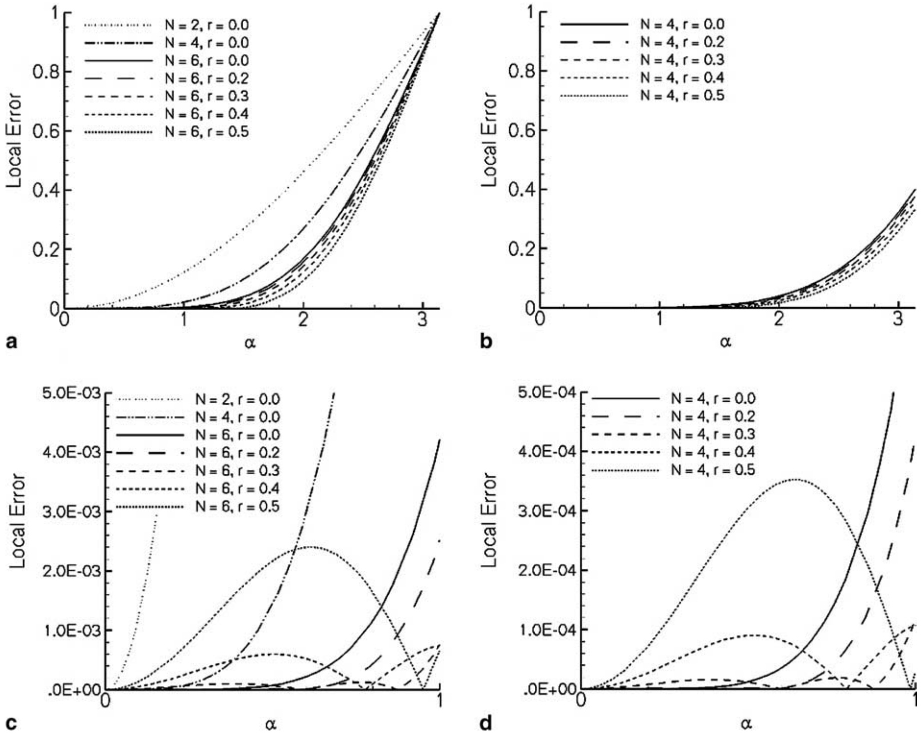

Fig. 2. Local error profiles for various generalized Lagrangian interpolation methods, $\delta_C = 0$: (a) explicit non-optimized ($r = 0.0$) and optimized ($r > 0.0$) methods, $0 \leqslant \alpha \leqslant \pi$; (b) compact non-optimized and optimized methods, $0 \leqslant \alpha \leqslant \pi$; (c) explicit methods in low wavenumber range $0 \leqslant \alpha \leqslant 1$; (d) compact methods in low wavenumber range $0 \leqslant \alpha \leqslant 1$.

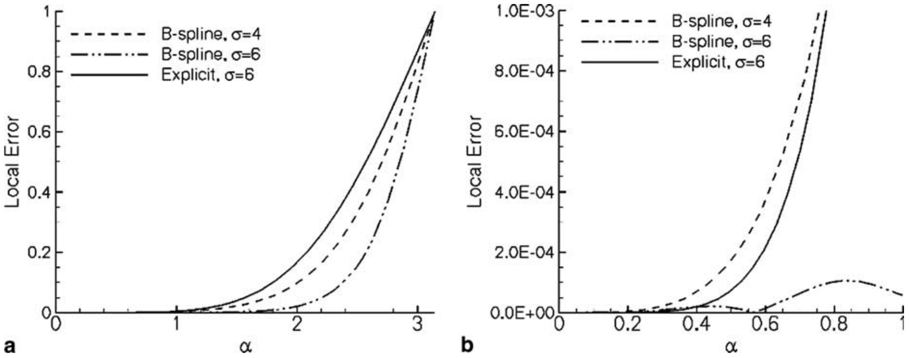

Fig. 3. Local error profiles for B-spline interpolation methods of various orders, $\delta_C = 0$, $N = 18$. An explicit, non-optimized, sixth-order Lagrangian interpolation method is included for comparative purposes: (a) $0 \leqslant \alpha \leqslant \pi$, (b) $0 \leqslant \alpha \leqslant 1$.

decrease in local error with the compact Lagrangian approach is also demonstrated, with over an order-of-magnitude improvement in error levels at lower wavenumbers compared to an explicit method with the same stencil size.

The effect of optimizing the explicit $N = 6$ case and the compact $N = 4$ case is also presented in Fig. 2 using values of the optimization parameter r defined in Eq. (5) ranging from 0.2 ($\alpha = \pi/5$) up to 0.5 ($\alpha = \pi/2$). Increasing values of r results in the steady improvement in the local error profiles for $\alpha > 1$. However, upon close examination of the lower wavenumber range $\alpha \leqslant 1$, non-monotonic increases in the local error become visible in the optimized approach. This behavior is observed for the explicit method in [25] and is seen here for the compact method as well. While similar in a qualitative sense, the compact method is observed to lower the local error by an order of magnitude compared to the explicit methods. Increasing the value of the optimization parameter r is seen to shift the wavenumber at which the local error ultimately begins its steep monotonic increase to the right, although it also enlarges both the width and the height of these pockets of increased error at the lower end of wavespace. This conflicting behavior points to the question of whether the decrease in the local error at the higher wavenumbers is worth the increase in error at the lower wavenumbers. Also, it is clear from Fig. 2 that for some part of the wavespace, the integrated error of the non-optimized method will be less than that of the equivalent optimized method for the same stencil size. Both of these issues will be examined further in subsequent sections.

The spectral error behavior associated with the B-spline method is considered in Fig. 3, which shows the local error as a function of wavenumber for the fourth- and sixth-order B-spline methods with $\delta_C = 0$ and $N = 18$. The local error profile for an explicit non-optimized method on a six-point stencil is also shown for reference purposes. For $\alpha \geqslant 1$, both B-spline methods possess lower local errors than the sixth-order explicit method, with the sixth-order B-spline having the lowest local error in this range. However, non-monotonic behavior similar to the optimized Lagrangian methods is also observed for the sixth-order B-spline method for $\alpha < 1$. This behavior is not observed at this value of N with the fourth-order B-spline. Although not shown here, the non-monotonic regions of the local error increase in magnitude with an eighth-order B-spline [26]. These local error increases are attributed to the use of the natural spline boundary conditions, as the non-monotonic behavior appears in the $\sigma = 4$ case for sufficiently small values of N and disappears in the $\sigma = 6$ case for sufficiently large values of N. The local error associated with the fourth-order B-spline is less than that of the explicit method for $\alpha > 1$. However at lower wavenumbers, the explicit method produces less error. As is the case in the optimized Lagrangian methods, there is some portion of the lower wavespace over which a lower-order B-spline method will outperform a higher-order method. The impact of the natural-spline boundary conditions is also observed in the following section on the integrated error and during the validation of the interpolation methods in the high-order overset-grid code in Section 5.

3.2. Integrated error analysis

Additional analysis of the various interpolation methods is provided by considering the integrated error defined in Eq. (3). One of the key parameters when utilizing the integrated error is the range of wavespace over which the integration takes place, denoted by R. This parameter is different from the optimization parameter r, defined in Eq. (5), which governs the range of wavespace over which the integrated error is optimized. All integrated error analysis presented here assumes a value of $R = 1/\pi$, which is equivalent to a gridpoint density of 2π points-per-wave (PPW) unless otherwise indicated.

To determine the effect that the location of the interpolation point within a stencil has on the error, the integrated error is plotted as a function of δ_C for both explicit and compact non-optimized Lagrangian interpolation methods of various stencil sizes in Fig. 4. The integrated error is seen to vanish when the interpolation point lies right on top of a stencil point (the direct-injection situation). When located between stencil points, the integrated error reaches its maximum near the center of the subinterval. The dashed lines in Fig. 4(a) connect the peak error in each subinterval for a given stencil size as the interpolation point location to demonstrate the increase in error as the interpolation stencil becomes more one-sided. For the $N = 6$ case, an order-of-magnitude increase is observed in the maximum error between the center and outermost subintervals. For overset-grid applications using greater than second-order interpolation, the location of the interpolation point within the stencil is related to the amount of overlap between grids. Overset grids with large amount of overlap will produce stencils with smaller values of δ_C, i.e., more centered stencils, while small overlaps will result in stencils with larger values of δ_C, i.e., more one-sided stencils.

In Fig. 4(b), heavy solid lines are shown connecting the peak error for a given stencil size N to the peak error for an $(N + 2)$-point stencil in the adjacent outward subinterval. This represents the situation where the order of the interpolation method is increased by the addition of points on one side of the stencil only, which is often encountered in overset-grid systems due to the presence of valid donor points on only one side of the interpolation stencil. The decreasing nature of the curves represented by these lines for all of the stencil sizes considered here (and for stencil sizes up to $N = 10$ in [26]) demonstrate the benefit of increasing the order of the interpolation method even if it results in a more one-sided stencil. However, in [26] it was shown that as the value of R is increased, the errors present at the higher wavenumbers due to the one-sidedness of the stencil eventually overcomes the reduction of the error at the lower wavenumbers obtained with the larger stencil, and the overall integrated error will go up. However, this does not occur until very high wavenumbers ($R > \approx 0.5$, or 4 PPW) for explicit methods up to sixth-order, and still higher values for compact methods.

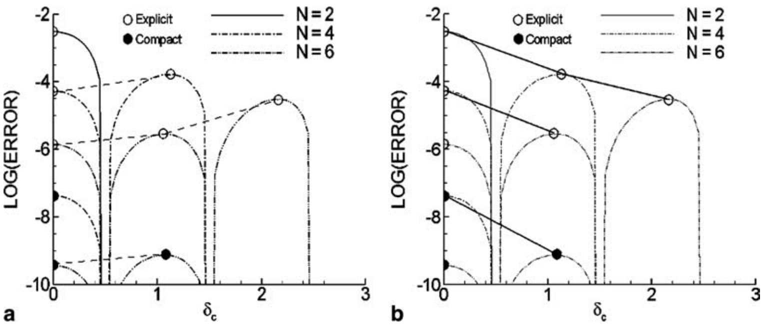

Fig. 4. Integrated error profiles as a function of δ_C for various non-optimized Lagrangian methods: (a) explicit and compact profiles with the peak integrated error identified in each subinterval with a circle; (b) behavior of integrated error at midpoint of outermost subinterval as stencil width is increased by two (indicated by heavy solid line).

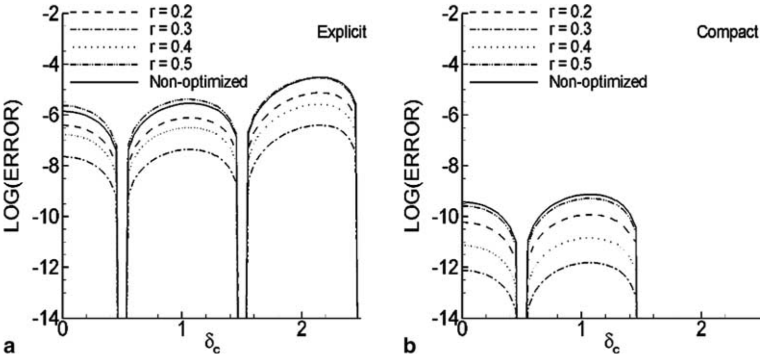

Fig. 5. Integrated error profiles as a function of δ_C for optimized Lagrangian interpolation methods: (a) explicit, (b) compact.

In Fig. 5, the integrated error profiles obtained with $R = 1/\pi(R \approx 0.32)$ are shown for optimized explicit and compact methods using various values of the optimization parameter r for $N = 6$. The integrated error for the $r = 0.3$ case is two to three orders of magnitude less than its non-optimized counterpart due to the close correlation between r and R in this case, as the integration is taking place over nearly the same range of wavespace as the optimization. With the compact method, all values of the optimization parameter up to $r = 0.5$ result in a reduction in the error compared to the non-optimized method. For the explicit method, the use of optimization generates a lower error than the non-optimized method for all values of r shown except for $r = 0.5$. In this case, the previously observed local error increases at the lower wavenumbers become large enough to offset any improvement obtained at the higher wavenumbers.

This trade-off is further demonstrated in Fig. 6, which plots the contours of the error reduction factor ϵ, defined as the log of the ratio of the integrated error between the optimized and non-optimized situations, or

$$\epsilon = \log_{10} \frac{E_{\text{int}}^{\text{opt}}}{E_{\text{int}}^{\text{non-opt}}} \tag{8}$$

as a function of the optimization parameter r and the integration parameter R for an explicit method with $N = 6$. A maximum error reduction of approximately two orders of magnitude takes place along the curve

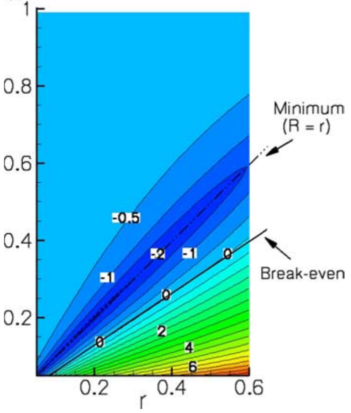

Fig. 6. Contours of the error reduction factor (ERF) (defined by Eq. (8)) as a function of the integration parameter R and the optimization parameter r.

$R = r$, where the integrated error is evaluated over the exact range of wavespace in which optimization is taking place. Both above and below this line, lesser values of ϵ are obtained. Somewhere below this curve, ϵ switches signs, indicating an error increase rather than a reduction when utilizing the optimized method. The curve across which this sign change takes place, reasonably represented by a straight line for $r < 0.6$ given by the formula

$$R_{\mathrm{BE}} = 1.504r + 0.005 \tag{9}$$

is the breakeven curve for the explicit, $N = 6$ method. Along this curve, the errors generated by the optimized and non-optimized methods are equal, while below this curve, the error associated with the optimized method can be orders-of-magnitude higher than the non-optimized method.

The error behavior associated with B-spline methods of various orders is considered in Fig. 7, which shows the integrated ($R = 1/\pi$) error as a function of the interpolation stencil size N for two interpolation point locations; at the midpoint of the outermost subinterval of the stencil ($\delta_C = (N - 2)/2$) and at the midpoint of the center subinterval of the stencil ($\delta_C = 0$). In the outermost subinterval, the integrated error quickly reaches a value independent of the stencil size but dependent upon the order of the B-spline. The error in this region is dominated by the natural spline boundary conditions, and is several orders-of-magnitude higher than the error from the center of the stencil. In the center subinterval, the error levels decrease with increasing stencil size as the effects of the natural spline assumption lessen as the distance from the boundary grows. For stencil sizes $N \geqslant 10$, the error at $\delta_C = 0$ for the fourth-order B-spline has essentially reached a constant value independent of the stencil size. For the sixth-order B-spline this constant value is not reached until $N = 24$, while for the eighth-order B-spline (not shown here) the constant value had not yet been reached at $N = 30$ [26].

Summary plots of the integrated error associated with all of the general interpolation methods analyzed here is given in Fig. 8. This figure shows the integrated error for optimized/non-optimized and explicit/compact Lagrangian methods as well as B-spline methods as a function of the parameter ϕ. This parameter is defined by $\phi = \sigma_{\max}$ for Lagrangian methods and $\phi = \sigma$ for B-spline methods, and provides a means to consistently compare methods whose formal order-of-accuracy is independent of the stencil size (optimized Lagrangian methods and B-spline methods), to those whose order is linked to the stencil size (non-optimized Lagrangian methods). The two extreme interpolation point locations are again considered where the primary interpolation point is located at the midpoint of the center subinterval ($\delta_C = 0$) and the midpoint of the outermost subinterval ($\delta_C = (N - 2)/2$). For the optimized methods, the errors are only shown for the value of r that produced the lowest integrated error, which is $r = 0.3$ for $R = 1/\pi$. For the B-spline methods, the stencil sizes employed are large enough such that the errors have become independent of N.

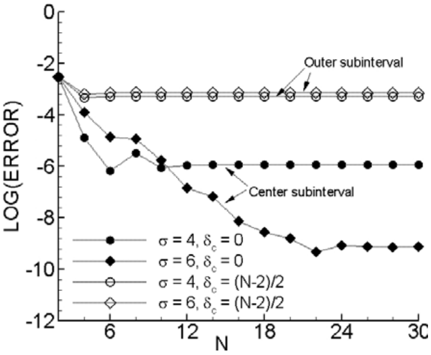

Fig. 7. Integrated error profiles as a function of stencil size N for B-spline interpolation methods. The interpolation point is located at either the middle of the center subinterval ($\delta_C = 0$) or the middle of the outermost subinterval ($\delta_C = (N - 2)/2$) of the stencil.

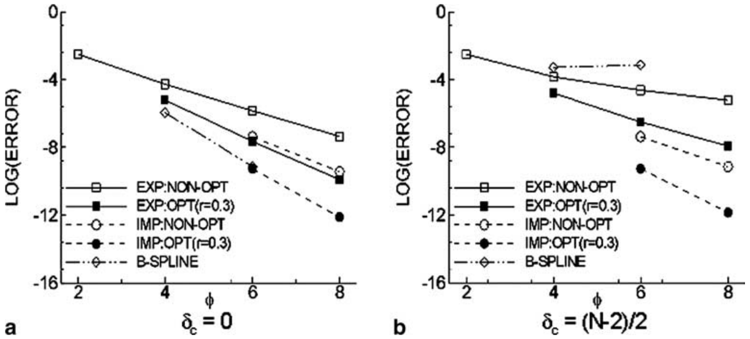

Fig. 8. Summary of integrated errors ($R = 1/\pi$) in middle of center and outermost subintervals for various interpolation methods as a function of ϕ.

Several general points may be drawn from this figure and the one-dimensional analysis and discussion presented in the rest of this section. These include the fact that the use of compact Lagrangian methods results in errors that are several orders of magnitude less than comparable explicit Lagrangian methods. Also, with judicious selection of the optimization parameter r, the optimization of the interpolation coefficients in the generalized Lagrangian method may result in orders of magnitude reduction in error over their non-optimized counterparts. However, because of the increase in error at the lower wavenumbers and the dependence on r, the use of optimization may also result in integrated errors that are greater than comparable non-optimized methods over some portion of wavespace. With regards to the B-spline method, it is very competitive from an absolute error standpoint compared to the generalized Lagrangian methods when the interpolation point lies in the center of the stencil. For example, with $R = 1/\pi$, the sixth-order B-spline has an integrated error that is comparable to the optimized compact method on a four-point stencil. Due to the use of the natural spline boundary conditions, the error at the ends of the stencil exceeds those from the Lagrangian methods, which could limit the usefulness of the B-spline approach with natural spline boundary conditions for overset-grid applications.

4. Implementation in high-order overset-grid code

4.1. Description of flow solver algorithm

The set of equations solved by the current high-order algorithm are the full Navier–Stokes equations, written in strong conservation form after application of a general time-invariant curvilinear coordinate transformation $(x, y, z) \to (\xi, \eta, \zeta)$ as

$$\frac{\partial}{\partial t}\left(\frac{\vec{U}}{J}\right) + \frac{\partial \hat{F}_I}{\partial \xi} + \frac{\partial \hat{G}_I}{\partial \eta} + \frac{\partial \hat{H}_I}{\partial \zeta} = \frac{1}{Re}\left[\frac{\partial \hat{F}_v}{\partial \xi} + \frac{\partial \hat{G}_v}{\partial \eta} + \frac{\partial \hat{H}_v}{\partial \zeta}\right]. \tag{10}$$

The solution vector in the above equation is given by $\vec{U} = \{\rho, \rho u, \rho v, \rho w, \rho E\}$, while the inviscid flux vectors $\hat{F}_I, \hat{G}_I, \hat{H}_I$, the viscous flux vectors $\hat{F}_v, \hat{G}_v, \hat{H}_v$, the metric identities used in the grid transformation, and the supplemental equations for a calorically and thermally perfect gas (as is assumed here) are given in [18]. For the applications shown later in this work that involve solving the Euler equations, the three viscous flux vectors Eq. (10) are set equal to zero.

The above set of equations is spatially discretized using the centered, finite-difference formulae presented in [18]. The resulting family of schemes ranges from an explicit, three-point, second-order accurate method up to a compact, five-point, sixth-order method requiring the solution of a tridiagonal system of equations to evaluate the spatial derivatives. When at a computational boundary for three-point stencils or within one point of a boundary for five-point stencils, the centered formulae are replaced with appropriate one-sided differences. The formulation and associated coefficients for the various centered and one-sided differencing schemes used here are presented in [18,34].

Because the centered schemes are non-dissipative, a mechanism is required to prevent the uncontrolled growth of high-frequency modes that can cause numerical instability. A high-order, low-pass, Padé-type filter [2] is thus employed to removed these spurious, high-frequency artifacts. Up-to-10th-order centered filters requiring an 11-point stencil are used on the interior of the computational domain, while high-order one-sided filters are employed within five points of a computational boundary. A free parameter, defined as α_f and referred to as the filter coefficient, provides spectral control of the filter [16]. A compilation of the filter coefficients as functions of α_f for both interior (centered) and boundary (one-sided) filters are given in [17,34].

The notation employed to denote the particular spatial algorithm used for a given case is taken from [17]. The order of the scheme used to discretize the metrics, flux vectors or flow variables at the boundaries as well as on the interior of the domain is given by a set of digits equal to the order-of-accuracy of the scheme. The number in the center of this set indicates the order-of-accuracy of the interior method, while the outer numbers represent the order in the vicinity of each of the boundaries. For example, a scheme denoted by 45654 would employ a sixth-order accurate discretization scheme on the interior of the domain, and reverting to fifth-order at one point away from each boundary and fourth-order on each boundary. Compact and explicit schemes are indicated by the prefix C and E, respectively, followed by a single number representing the interior order of the solver ($C6$ for a compact sixth-order interior solver, $E4$ for an explicit fourth-order interior solver). The interior filter is denoted by its order-of-accuracy superscripted with the filter coefficient employed and subscripted with the minimum non-zero order-of-accuracy of the one-sided filters utilized at computational boundaries. For example, $F10_6^{0.4}$ would indicate an interior 10th-order filter is employed with $\alpha_f = 0.4$ and a minimum sixth-order boundary filter at one point from the computational domain. The points directly on the computational domain are not filtered [17].

For use with the time-dependent problems considered in this work, the high-order overset-grid algorithm is been coupled with both an explicit time integration scheme as well as implicit time integration scheme. The explicit approach employs the classical fourth-order Runge-Kutta method (RK4) implemented in low-storage form [35], and is utilized for wave propagation problems. Problems requiring very fine resolution such as wall-bounded viscous flow are handled with the approximately factored Beam-Warming (BW) implicit method [36]. This approach also uses the more efficient diagonal form of the approximately factored algorithm [37], and includes non-linear artificial dissipation terms added to the implicit operator in order to enhance stability [38]. Newton-like subiterations are employed to maintain time-accuracy by diminishing the effects of the linearization, diagonalization, explicit boundary conditions and artificial dissipation. Additional details on both the RK4 and BW implementations in the current method may be found in [18].

4.2. Implementation of overset capability

In general overset-grid methods, the computational domain is broken down into individual grids typically consisting of individual body-fitted grids around each geometric feature of interest, and a background grid in which the body-fitted grids are embedded. Extra grids can be included to provide additional resolution in key areas or for some other specific purposes. In general, the points in the overlapping regions between grid components are non-coincident. The application of overset-grid methods requires additional procedures and data structures to manage the various datasets associated with multiple grids. In the serial version of the current algorithm, these include one-dimensional arrays for the efficient storage of gridpoint, flowfield, and connectivity data, and pointer arrays to access the region of the arrays corresponding to the current grid being manipulated.

In the parallel version of the algorithm, these grids are decomposed into overlapping regions referred to here as "blocks" for parallel processing using the Message-Passing Interface (MPI) communications library. Each block is then solved on a single, unique processor. Each processor has the grid and flowfield data corresponding to the its block already in its local memory, and communication between blocks is handled though finite-sized overlaps consisting of coincident grid points. Based on previous studies [17], five-point overlap regions are sufficient to recover the interior high-order differencing and filtering of the original single-block algorithm, and thus for all results shown here, a two-point fringe (two levels of interpolation points) is employed at all interpolation boundaries. The number of processors available for the particular problem, the amount of memory available on each processor, and load balancing issues all drive the decomposition of grids into blocks.

After each grid in the serial version or block in the parallel version has been advanced one subiteration of the BW method or one stage of the RK4 method, boundary conditions, including the interpolation boundary conditions, are applied. After completing the current all subiterations/stages, the spatial filter is then applied to each grid, and the boundary conditions are applied again. Thus, each time step requires $T + 1$ applications of the interpolation boundary conditions, where T is the number of subiterations or stages in the time-integration method.

The extension of the one-dimensional interpolation formulae presented in Section 2 to multiple dimensions is done by lines, with the three-dimensional interpolation problem broken down into three successive one-dimensional problems [39]. Fig. 9 shows a schematic of this approach for a two-dimensional situation using four-point stencils in each direction. In Fig. 9(a), a set of interpolation points denoted by asterisks are shown along with a portion of the donating grid in computational (ξ, η)-space. The first step is to find the interpolated values at the extended interpolation points denoted by the filled circles shown in Fig. 9(b). This is done using the one-dimensional interpolation formulae along each constant-η line possessing at least one extended interpolation point. For the compact Lagrangian approach, the extended interpolation points in the rectangular boxes along lines of constant η are coupled together and solved via a single tridiagonal system. Similarly, the B-spline approach employs a donor stencil that encompasses extended interpolation points grouped along a single constant-η line to calculate the collocation coefficients via Eq. (B.15) via a single matrix inversion, after which the interpolated values at each point are calculated. These extended

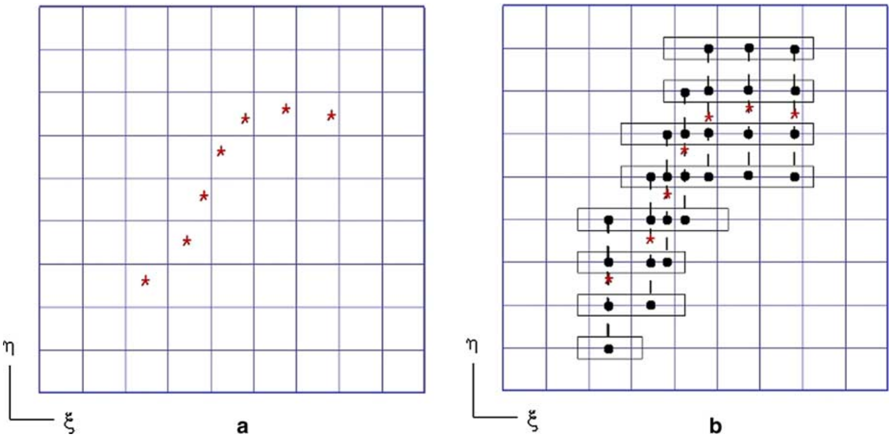

Fig. 9. Implementation of one-dimensional interpolation formulae in multiple dimensions: (a) distribution of interpolation points within the computational space of a donor grid; (b) coupling of extended interpolation points along a constant-η coordinate line.

interpolation points then serve as donor points for the one-dimensional interpolation along the constant-ξ lines (indicated by dashed vertical lines) to calculate the necessary values at the original interpolation points. In three dimensions, this two-dimensional process is repeated on each plane, and a final interpolation is performed normal to the planes to fix the interpolated value in the third dimension.

While straightforward to implement with explicit methods, the approach shown in Fig. 9 does result in some computational challenges for the compact and B-spline interpolation methods. For the compact approach, the issue of the uneven distribution of interpolation points is one of the most troublesome. The use of compact methods to perform spatial differencing or filtering benefits from the fact that the operation is being performed at regular locations within a uniformly distributed set of grid points in computational space. Since the distribution of the interpolation points within a donor grid is arbitrary, forming a consistent and well-behaved coupled system of equations is much more challenging. Issues such as the clustering or scattering of extended interpolation points, including the limiting situation where multiple extended interpolation points are stacked on top of each other, result in coefficient matrices that could potentially be singular or ill-conditioned and must receive special treatment. In addition, the lack of sufficient interpolation points to form a coupled system along a line, and the potential for one or more interpolation points to be nearly coincident with donor points within a coupled system, require specific logic elements to be built into the algorithm to address these issues. For both the compact and B-spline approach, the extended interpolation points that are to be coupled along each line must be identified and grouped together, and in the compact case must be sorted as well in order to be processed correctly. The implication of this is that the compact approach for interpolation associated with general overset grids is less robust than the explicit or B-spline methods, and requires a considerable amount of additional coding to deal with potential difficulties arising from arbitrary interpolation-point distributions. The B-spline and explicit approaches are much more robust in this regard, but the B-spline method does require the additional step of grouping together the extended interpolation points to be coupled. More detail regarding these issues as well as discussions of additional challenges faced by compact and B-spline methods may be found in [26].

4.3. Calculation of high-order offsets and coefficients

The calculation of the high-order offsets and interpolation coefficients is accomplished during a pre-processing step. This step requires as input grid connectivity data as well as second-order offsets and coefficients calculated by one of the many existing grid assembly software packages [11]. The preprocessing code takes the second-order donor stencil identified by the grid assembly software for each interpolation point and expands it as necessary in all three coordinate directions. The expanded stencils are as centered as possible, with an equal amount of donor points added onto each side of the original stencil whenever possible. In the event that an potential expanded stencil point is not useable as a donor point, the necessary points are added to the opposite side of the stencil thus making it one-sided.

With the expanded stencils determined, the interpolation offsets are calculated using a high-order extension of the isoparametric mapping procedure of [21]. The interpolation offsets (the distance from the interpolation point to the base point of the stencil, indicated by δ in Fig. 1) map the physical location of the interpolation points into the computational space of the associated donor grid and are initially generated during the grid-assembly process using a second-order accurate method. They are then extended to higher orders using explicit, non-optimized Lagrangian interpolants for the isoparametric mapping. The high-order offsets in each coordinate direction are found by solving the set of three mapping equations

$$\vec{F}\left(\vec{x},\vec{\hat{x}},\vec{\delta}\right)=\sum_{k=0}^{N_\zeta-1}\sum_{j=0}^{N_\eta-1}\sum_{i=0}^{N_\xi-1}R^{\zeta}_{i,j,k}(\delta_\zeta)R^{\eta}_{i,j,k}(\delta_\eta)R^{\xi}_{i,j,k}(\delta_\xi)\vec{x}_{i,j,k}-\vec{\hat{x}}=0, \tag{11}$$

where the interpolation coefficients R^{ξ}, R^{η}, R^{ζ} in each direction are given as functions of the offset in that direction by Eq. (A.9), while $\vec{x}$ and $\vec{\tilde{x}}$ are the physical (x, y, z)-coordinate locations for the stencil and interpolation points, respectively. The above equation is solved iteratively for the offset vector $\vec{\delta}$ using Newton's method,

$$\vec{\delta}^{n+1} = \vec{\delta}^{n} - \left[\left(\frac{\partial \vec{F}}{\partial \vec{\delta}}\right)^{-1} \cdot \vec{F}\left(\vec{x}, \vec{\tilde{x}}, \vec{\delta}\right)\right]^{n} \tag{12}$$

starting with an initial guess for $\vec{\delta}$ given by the second-order offsets calculated by the grid-assembly program.

4.4. Hole management

A key aspect in the flexibility of the overset-grid approach is the ability to cut holes in a grid. This capability is useful to remove points from a background grid that would lie inside of a solid object in order to prevent non-physical solutions from evolving at these points. Holes may also be used to remove points from coarser grids in regions discretized by two or more grids to prevent the development of multiple solutions in those regions. An example of a hole cut in a background grid by a body-fitted curvilinear grid is shown in Fig. 10(a). In this figure, both functions of holes can be observed; the elimination of points from a grid that

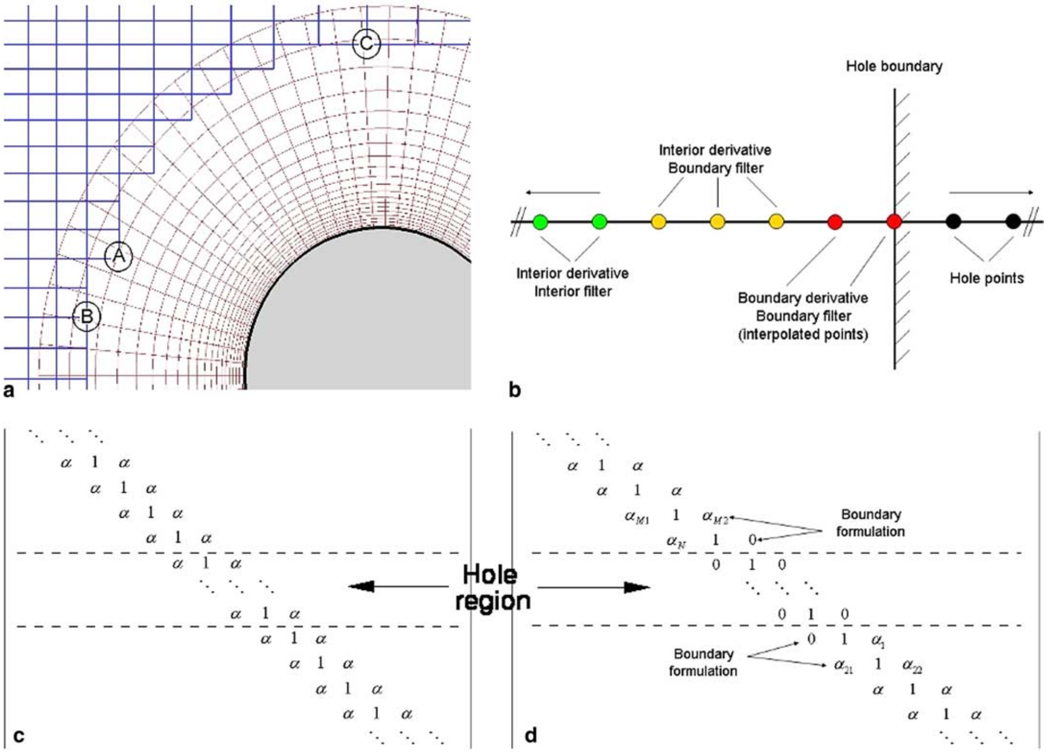

Fig. 10. Implementation of hole-cutting in high-order algorithm: (a) classification of hole-boundary points; (b) points impacted by a hole boundary for the *C6F*10 interior scheme; (c) portion of one-dimensional implicit filter operator before hole is accounted for; (d) portion of one-dimensional implicit filter operator after hole is accounted for.

would lie within a solid body, and the removal of points from coarser grids in regions containing multiple discretizations.

The presence of holes introduces computational boundaries within the regular structure of a block. These computational boundaries are typically irregular in nature, as is the case in Fig. 10(a). The preprocessor manages all holes created in the computational domain during the grid-assembly process by identifying points that lie on the hole boundary and determining which coordinate directions of the cut grid are impacted by the hole. For example, in the two-dimensional case shown in Fig. 10(a), the hole boundary point identified as point "B" only requires hole treatment in the computational coordinate direction corresponding to the horizontal ξ-direction of the background grid. In the computational coordinate direction corresponding to the vertical η-direction, no special treatment is required at point "B". Similarly, point "C" is only affected by the hole in η-direction, with no special treatment required in the ξ-direction. Point "A" is an example of a hole boundary point that would require special treatment in both computational coordinate directions. When performing differencing or filtering sweeps along a coordinate direction impacted by a hole, one or more computational boundaries are inserted into the algorithm as shown in the schematic in Fig. 10(b). This example assumes a five-point interior differencing stencil and an 11-point interior filtering stencil. Thus, locations within two points of a hole boundary must have their differencing formulae modified to the appropriate boundary formulation. Likewise, locations within five points of the hole boundary must employ a boundary filter treatment. Points that lie within the hole are decoupled from those outside the hole by replacing the tridiagonal coefficients corresponding to these points with the decoupled coefficients $\{0, 1, 0\}$ and zeroing out any contribution from the right-hand side of the operator. This is shown schematically in Fig. 10(c), which demonstrates the approach using a coefficient matrix corresponding to a compact differencing operator. The modification of the coefficients within the hole boundary, along with the insertion of the boundary formulations on each side of the hole, effectively break the implicit system into three decoupled regions; the hole region and the regions on each side of the hole. The resulting system of equations retains its tridiagonal form and thus may be efficiently solved. The same approach may also be used for the implicit system arising from the filtering operator with the necessary modifications to the boundary formulations. This approach may be repeatedly applied along a coordinate direction as often as necessary to account for the presence of holes, however care must be taken to ensure that adequate support for both the solver and filter formulations are maintained in all computational directions.

5. Results

5.1. Grid-to-grid interpolation test case

The first set of results presented is the interpolation of data generated by various multi-dimensional analytic test functions from one grid onto another. The results presented here were obtained from application of the interpolation routines only, which were isolated from the other aspects of the algorithm such as the temporal integration, spatial differentiation, and filtering. The purpose of this analysis is twofold; to validate the interpolation routines against known analytical test functions in a multi-dimensional setting, and to examine the error behavior of the various interpolation methods in a controlled and decoupled manner from the rest of the algorithm.

The two- and three-dimensional grid systems used for this study is shown in Fig. 11. Each system consists of a analytically generated curvilinear grid overlaid upon a background Cartesian grid. The minimum distance between the outer boundaries of the grids is approximately 10 background grid points in both cases, and the foreground and background grids in both cases have the same nominal grid spacing. A known analytical function is superimposed on the background grid and then interpolated onto the foreground grid. Because the test function and the foreground grid are both specified analytically, the exact

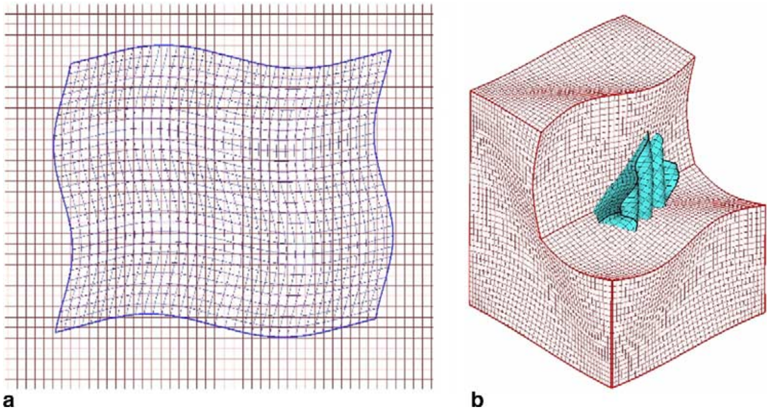

Fig. 11. Meshes employed in grid-to-grid interpolation test cases: (a) two-dimensional grids, (b) three-dimensional grids.

values of the function at the grid point locations of the foreground grid may be calculated and compared to the interpolated results.

Three two-dimensional analytic functions are considered here. These include a periodic function of the form

$$F1 =: \; f(x,y) = \cos\left(\frac{\omega x}{L_x} IL\right) \sin\left(\frac{\omega y}{L_y} JL\right) \tag{13}$$

with $\omega = \pi/2$, a general exponential function given by

$$F1 =: \; f(x,y) = (0.02)^2 x \exp\left(\frac{-\sqrt{x^2+y^2}}{2}\right) \tag{14}$$

and an arbitrary, manufactured, non-periodic function with a sinusoidal component given by

$$F3 =: \; f(x,y) = \frac{27x}{x^4 + 9(y^2+9)} \cos\left(\frac{20x}{y+3}\right) \cosh\left(\frac{x}{3}\right) \sinh\left(\frac{y}{3}\right). \tag{15}$$

On the three-dimensional grid, a periodic function of the form

$$F4 =: \; f(x,y,z) = \cos\left(\frac{\omega x}{L_x} IL\right) \sin\left(\frac{\omega y}{L_y} JL\right) \cos\left(\frac{\omega z}{L_z} KL\right) \tag{16}$$

with $\omega = \pi/2$ is considered. The accuracy of the various non-optimized interpolation methods is considered in Fig. 12, which plots the log of the L_2-error norm for each function as the grids are uniformly refined. All four functions considered here exhibit very similar error behavior, with the calculated order-of-accuracy of the explicit and compact methods shown in parentheses very closely matching the formal order of the method. A decrease of several orders of magnitude in interpolation error is observed when using the high-order Lagrangian formulations compared to standard second-order interpolation. In general, the use of compact methods results in a decrease in the error from about one-half to one order of magnitude compared with explicit methods of the same order. This is, however, considered an optimal scenario for the compact approach as the distribution of interpolation points is uniform since the entire foreground grid is receiving interpolated data as opposed to the standard overset situation where only points along one or more boundaries are interpolated.

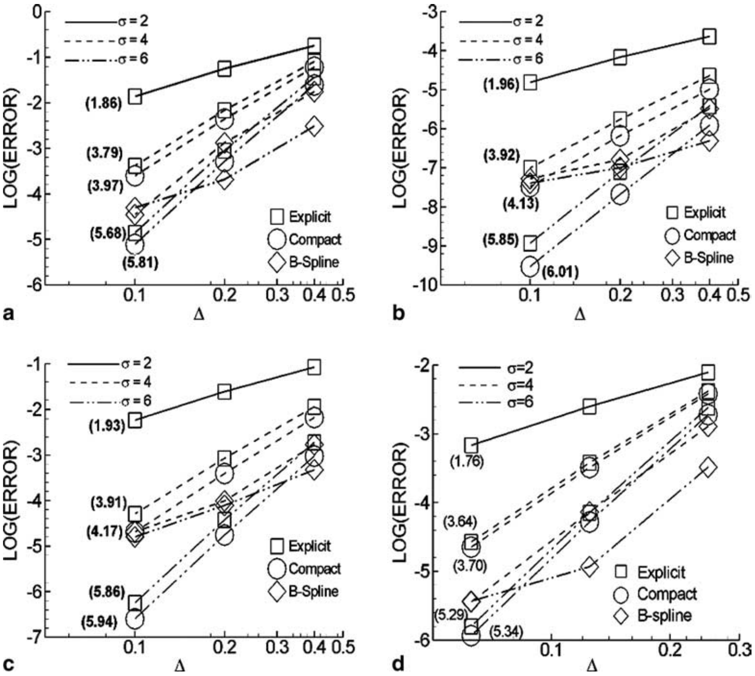

Fig. 12. Grid-convergence results for two- and three-dimensional analytic test functions interpolated grid-to-grid using non-optimized interpolation methods: (a) Function F1 (Eq. (13)), (b) Function F2 (Eq. (14)), (c) Function F3 (Eq. (15)), (d) Function F4 (Eq. (16)).

The behavior of the B-spline methods do not show a consistent linear decrease in error as the grid is refined, but instead decrease at a much lower and non-uniform rate from one resolution to the next. This behavior is more predominant in the sixth-order B-spline than in the fourth-order B-spline, and is related to the increase in error at the lower wavenumber range due to the natural spline boundary conditions observed previously in the Fourier analysis. For all of the test functions considered here, the fourth-order B-spline considerably outperformed in terms of interpolation error levels the Lagrangian methods of the same order on the coarsest grid, while being comparable or better on the finest grid. Conversely, while outperforming the Lagrangian methods of the same order on the coarse grid, the sixth-order B-spline has the highest error on the fine grid.

The effect of optimizing the explicit Lagrangian method on a six-point stencil is shown for the two-dimensional cases in Fig. 13, which demonstrates the grid-convergence behavior for the various test functions for several values of the optimization parameter r. Results from the non-optimized, $N = 6$ explicit interpolation method are also shown for comparison purposes. While nearly all values of r result in some reduction in error compared to the non-optimized approach on the coarse grid, the non-optimized method typically generates lower errors as the grid is refined. On the finest grid, the error consistently increases with increasing r for all functions. This is again due to the local error growth at the low wavenumbers seen in the Fourier analysis. Results obtained with the compact optimized method produced qualitatively similar results [26]. The optimized method can produce a lower interpolation error than the non-optimized method, particularly on coarse grids and/or for poorly resolved waves. However, the optimized approach was dropped from further consideration for use in the high-order overset-grid approach for the rest of this work.

5.2. Inviscid convecting vortex

The first problem examined with the entire high-order overset-grid algorithm is the inviscid convection of a vortex by an otherwise uniform freestream. This problem was previously examined with the current

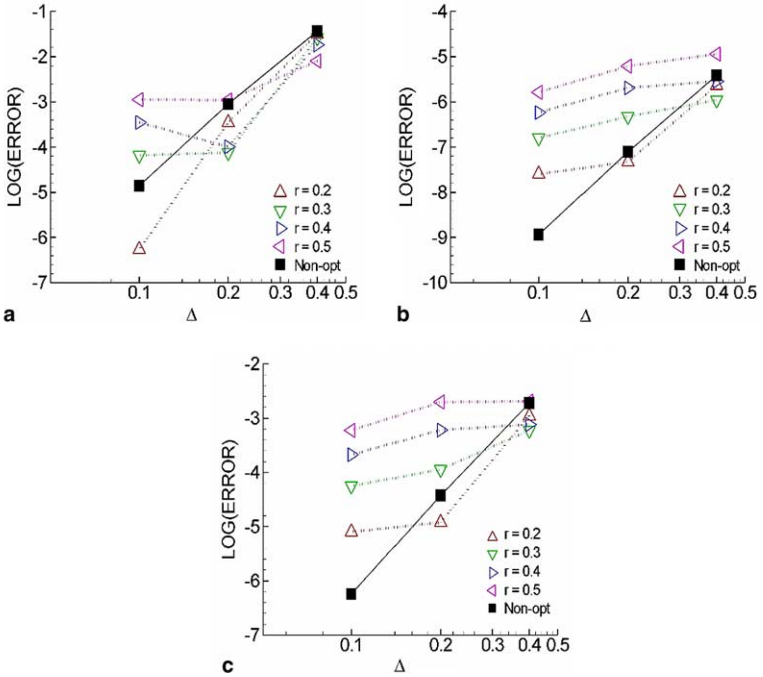

Fig. 13. Grid-convergence results for two-dimensional analytic test functions interpolated grid-to-grid using optimized interpolation methods with various values of the interpolation parameter r: (a) Function F1 (Eq. (13)), (b) Function F2 (Eq. (14)), (c) Function F3 (Eq. (15)).

high-order algorithm using various single-grid/single-block systems in [16,18], and with a single-grid system using two side-by-side blocks with coincident grid points in the overlap region in [17]. It is useful as a validation case for the new features of the high-order, overset-grid algorithm, such as the high-order interpolation routines and the techniques for handling hole points. It also allows for the investigation of issues associated with the introduction of computational boundaries that arise within the domain due to the grid decomposition process in a simplified manner.

The vortex is defined by the initial conditions [18]

$$\begin{aligned}
u &= 1 - \frac{C(y-y_c)}{R^2}\exp\left(\frac{-r^2}{2}\right),\\
v &= \frac{C(x-x_c)}{R^2}\exp\left(\frac{-r^2}{2}\right),\\
p_\infty - p &= \frac{\rho C^2}{2R^2}\exp(-r_2),\\
r^2 &= \frac{(x-x_c)^2+(y-y_c)^2}{R^2},
\end{aligned} \tag{17}$$

where u, v, p, ρ and R denote the Cartesian velocity components, static pressure, density and vortex core radius respectively. The non-dimensional vortex strength parameter $C/U_\infty R$ was set to 0.02, while the freestream Mach number and vortex core radius were taken to be $M_\infty = 0.1$ and $R = 1$. Three types of grid systems were considered; a single-grid/single-block system to provide baseline results, a multiple-grid/multiple-block overset Cartesian grid system shown in Fig. 14(a), where all overlap regions consist of coincident

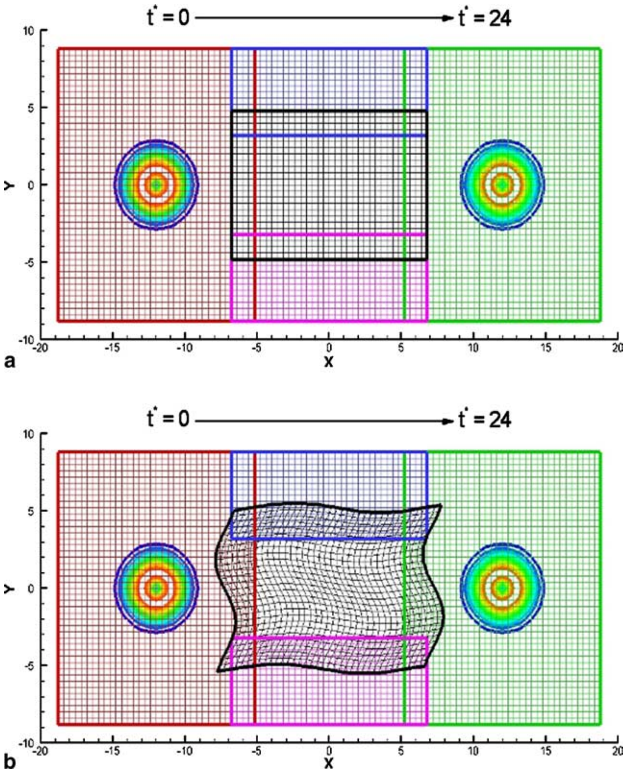

Fig. 14. Overset-grid systems used for convecting vortex problem; (a) direct-injection overset-grid system; (b) curvilinear overset-grid system.

grid points, and a multiple-grid/multiple-block overset curvilinear grid system shown in Fig. 14(b), where the overlap regions are no longer coincident. In all cases, the background Cartesian grid ranged from $-18 \leqslant x \leqslant 18$, $-8 \leqslant y \leqslant 8$, while the boundaries of the center grid in the two-grid systems are approximately $x \approx \pm 8$, $y \approx \pm 6$. The overset center grid is also used to cut a hole in the background grid. The vortex is convected downstream from its starting point at $x_c = -12$, $y_c = 0$ for 24 characteristic times using the explicit RK4 method with $\Delta t = 0.005$ until it reaches $x_c = 12$, $y_c = 0$. The error metric selected to evaluate the accuracy of the algorithm for this problem is the maximum difference between the exact and computational values of the v-component of velocity (swirl velocity) along the centerline of the vortex at its final location.

The first analysis shown here examines the roles that the spatial discretization scheme and minimum-order boundary filter employed at the computational boundaries has on the accuracy and stability of the solution. The vortex was convected on both a fine-grid system with a nominal grid spacing of $\Delta = 0.2$ in both directions and a coarse-grid system with $\Delta = 0.4$ using the multiple-block topology from Fig. 14(a). The vortex passes through two computational boundaries as it convects downstream. Since the grid points in the overlap regions are coincident, no interpolation is required and the effect of the presence of computational boundaries alone may be examined. The peak errors obtained with various solver algorithms and values of β, the minimum order-of-accuracy of the one-sided filter formulations used at one point off the computational boundary, are shown in Fig. 15 for the two grid resolutions considered. Three variations of the spatial-differencing algorithm were employed, including a fully fourth-order

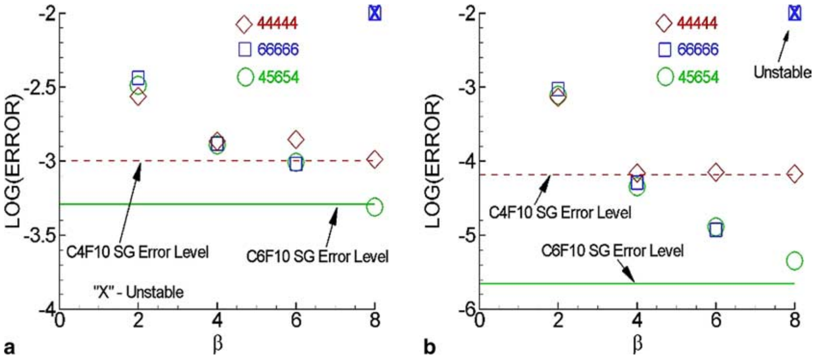

Fig. 15. Effect of solver algorithm and minimum-order boundary filter on error associated with direct-injection overset-grid system: (a) coarse grid (Δ = 0.4), (b) fine grid (Δ = 0.2).

compact algorithm (denoted by 44444), a fully sixth-order compact algorithm (66666), and a compact algorithm that is sixth-order on the interior of the computational blocks, but lowers to fifth-order accurate one point away from block boundaries and to fourth-order on the boundaries themselves (45654). On the interior of the blocks, the filter is given by $F10^{0.45}$ for all cases, while the boundary filters ranged from a minimum second-order filter ($\beta = 2$) up to a minimum eighth-order filter ($\beta = 8$). For comparison purposes, the baseline results obtained with the single-block topology using the $C4F10^{0.45}$ and the $C6F10^{0.45}$ spatial algorithms are indicated by the horizontal lines. The overall trend toward the single-block results as the minimum-order boundary filter is increased is observed. Clearly, the use of the second-order boundary filter (the "low-order centered" approach from [17]) limits the accuracy of the solution for all of the solver algorithms. As one would expect, the higher-order boundary filters are of limited usefulness with the 44444 scheme, as the order of the solver limits the overall accuracy of the method. Going to the 66666 method removes this limitation for the higher-order boundary filters, but this solver algorithm was unstable on both grids examined here for $\beta = 8$. However, use of the 45654 scheme allowed a stable solution to be obtained for the minimum eighth-order boundary filter, which resulted in solutions that very closely match the accuracy of the fully sixth-order algorithm applied on a single-block topology. Thus, the 45654 solver algorithm is used throughout the rest of this work and will be generally referred to as the $C6$ algorithm.

Next, the vortex was convected on the grid system shown in Fig. 14(b) in order examine the interpolation methods and further validate the handling of hole points. This grid system consists of a single curvilinear grid overset on a background Cartesian grid system, with the curvilinear grid cutting an irregularly shaped hole in the background grid. Grid-convergence results obtained with the $C6F10_4^{0.45}$ algorithm are shown in Fig. 16(a) for the single-grid/single-block topology, the single-grid/multiple-block direct-injection topology, and the multiple-grid/multiple-block topology with second- and sixth-order interpolation. The use of the second-order interpolation method results in an overall second-order accuracy. However, the results from the sixth-order interpolation method approach those obtained in the direct-injection case, where there is no interpolation error and the presence of the computational boundaries provides the only source of difference from the single-block results. The swirl velocity along the horizontal centerline is plotted in Fig. 16(b) and demonstrates the dissipation of the vortex due to the use of second-order interpolation. The results for the $\sigma = 4$ and $\sigma = 6$ interpolation are in much better agreement with those obtained on the direct-injection grid. Preliminary results obtained using a fourth-order B-spline interpolation method and a sixth-order compact interpolation method showed that these interpolation approaches produced results that were only marginally better or even slightly worse than those obtained with the non-optimized explicit interpolation shown here [26]. Fig. 17, showing contours of velocity perturbation at five different times as the vortex convects

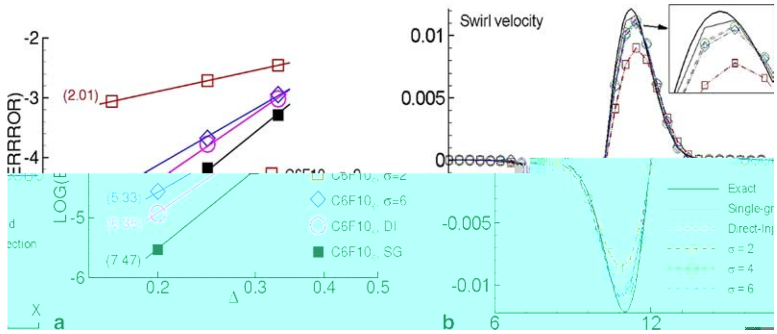

Fig. 16. Results for curvilinear overset-grid system; (a) grid convergence behavior for various grid systems and orders of interpolation; (b) swirl velocity profiles on course grid (Δ = 0.4).

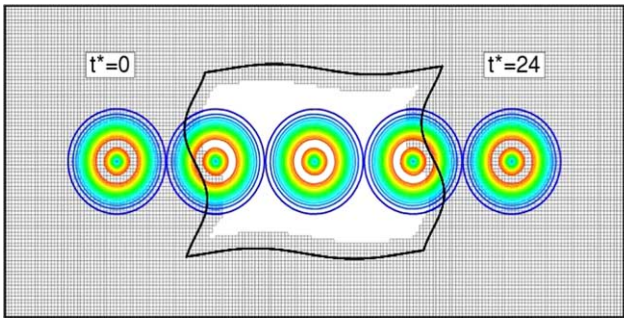

Fig. 17. Velocity perturbation contours showing vortex passing across grid interfaces and through hole region on curvilinear overset-grid system.

downstream on the Δ = 0.2 grid, demonstrates the smooth passage of the vortex through the overlap region and across the hole with the sixth-order explicit interpolation method.

The final study utilizing the convecting vortex model problem examined the effect of cell-volume discrepancies between overlapping grids and evaluated the robustness of the algorithm for handling such situations. The previous grid systems were generated with cell volume ratios of unity in the overlap region between the two grids. Now, the volume of the cells for the inner curvilinear grid was halved and then halved again to generate two topologies that possessed significant grid-resolution discrepancies in the overlap region (see Fig. 18(a)). The resulting topologies thus had volume ratios of 4 and 16, respectively, between inner and outer grids. The vortex was again convected downstream using the same $C6F10_4^{0.45}$ algorithm as previously used. The swirl velocities, plotted in Fig. 18(b), show no degradation in the solution due to the large change in cell volumes between the grids, but instead improve compared to the exact results due to the superior resolution of the inner grids. The filter is effective at removing the spurious waves that are generated due to the change in cell volumes at the grid overlaps, thus enhancing the robustness of the overall algorithm.

5.3. Viscous flow over one and two circular cylinders

The next validation case examined here is the low-speed ($M_\infty = 0.1$), viscous, two-dimensional flow over one- and two-circular cylinder configurations at $Re_D = 100$. The one-cylinder problem has been previously

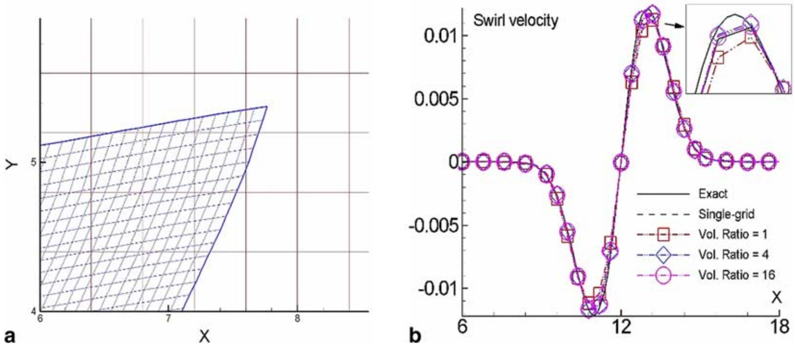

Fig. 18. Effect of cell volume ratio between grids on algorithm: (a) partial view of topology showing a volume ratio of 16 between inner and outer grids; (b) swirl velocities for topologies with various volume ratios.

examined for these flow conditions using the current high-order algorithm on single-block topologies [16] as well as multi-block, overset-grid topologies with coincident grid points in the overlap region [17,40]. This problem has also been examined using B-spline methods with zonal grids [41]. For the two-cylinder case, experimental data exists and is summarized, expanded upon, and interpreted by Williamson [42] and Xu et al. [43]. This data consists primarily of flow visualization shedding frequencies for the various modes that exist for this general configuration. For the computations shown here, all cylinders are identical, with diameters given by $D = 1$. For the two-cylinder configuration, the centers of the cylinders are located a distance of $1.3D$ apart, and the line connecting the centers is perpendicular with the freestream flow (i.e., no stagger).

The solution obtained on a single-grid topology consisting of an O-type grid with clustering in the wake region and near the surface of the cylinder was used to provide baseline data for comparison with the overset-grid results. The overset-grid topology is shown in Figs. 19(a) and 19(b) and consists of two O-type, topologically similar grids created by splitting the single grid and rotating the outer grid such that the points in the overlap region are no longer coincident. For this topology, the overlap is uniform and consists of the minimum number of points to prevent circular interpolation (whereby interpolation donor points are also receiver points obtaining their data from the grid that they themselves are trying to update) when using a two-point fringe. A direct-injection overset-grid system was also constructed for comparison purposes in which the grid was split but not rotated, thus resulting in a two-grid system with coincident grid points in the overlap region. For the cases examined here, the $C6F10_4^{0.45}$ spatial algorithm was employed, and the diagonalized second-order Beam-Warming algorithm was used with a non-dimensional time step of $\Delta t = 0.002$ to advance the solution in time.

Figs. 19(c) and 19(d) show the time history of the drag coefficient C_D on the single cylinder for various interpolation methods. The results obtained using the fourth- and sixth-order explicit interpolation methods match very well with those obtained in the direct-injection case, indicating again that the interpolation adds little error beyond that generated by simply splitting the grid. In this case, the results obtained using fourth-order B-spline interpolation with natural spline boundary conditions are also shown, but this method did not perform as well as the high-order explicit methods. This is due to the small amount of overlap, which results in interpolation points located in the outer subintervals of their stencils where the effect of the natural spline boundary conditions are felt. As expected, the second-order explicit interpolation method shows the poorest behavior compared to the single-grid results. A snapshot of the vortex street downstream of the cylinder at an instant in time obtained using sixth-order explicit interpolation is shown in Fig. 19(e), and demonstrates the smooth passage of the vortices across the interpolation boundary.

For the two-cylinder problem, a grid system consisting of seven grids was constructed that included a body-fitted polar grid around each cylinder and a well-resolved wake grid downstream. While the previous

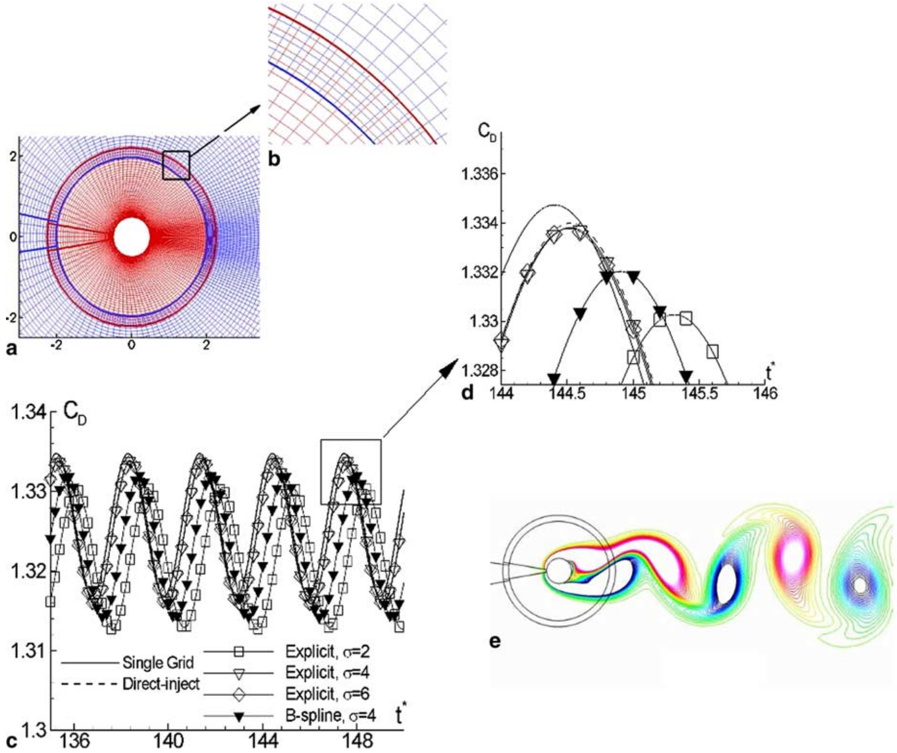

Fig. 19. Single-cylinder overset-grid system and associated results: (a) overset-grid system, (b) close-up of overlap region, (c) drag coefficient profile for several interpolation methods, (d) close-up of drag coefficient profile, (e) instantaneous vortex street behind cylinder obtained with sixth-order interpolation.

calculations were performed with a vector version of the algorithm, these seven grids were decomposed into a total of 32 blocks for processing on a parallel platform. These grids are used to cut holes in each other as well as the background grids as shown in Fig. 20, resulting in complex holes and overlap regions. The spatial and temporal algorithms employed are identical to those from the single-cylinder case, with the results shown here obtained using sixth-order explicit interpolation at all non-coincident overlaps.

The instantaneous vortex street formed by the two cylinders is shown in Fig. 21. It consists of vortex pairs being shed through the roll-up of the shear layers on the outer surfaces of the cylinders and a clear asymmetry in the gap region between the cylinders. This behavior qualitatively matches experimental results where asymmetric gap flow has been observed even at low Reynolds numbers. This asymmetry also manifests itself in the formation of the outer vortices, as those formed on the side towards which the gap flow deflects tend to be long in the cross-flow direction and thin in the streamwise direction, while those formed on the side away from the gap deflection tend to be shorter and broader. Experimental visualization indicated that for small gap widths, the gap vortices are ‘‘squeezed, weakened and amalgamated’’ [42] with the more dominant, outer vortices. This behavior was qualitatively observed in animations generated of the computed time-dependent vortex street here as well.

From comparison of the computational visualization to its experimental counterparts, the current calculation appears to depict the fundamental mode of vortex shedding from the two cylinders. Comparing spectral measurements of the data further reinforces this view. In [43], hot-wire measurements were taken downstream of the cylinders at the locations $x = 4D$, $y = \pm 2.65D$, referenced from the midpoint of the line connecting the centers of the cylinders. These measurements were then used to calculate power spectra of

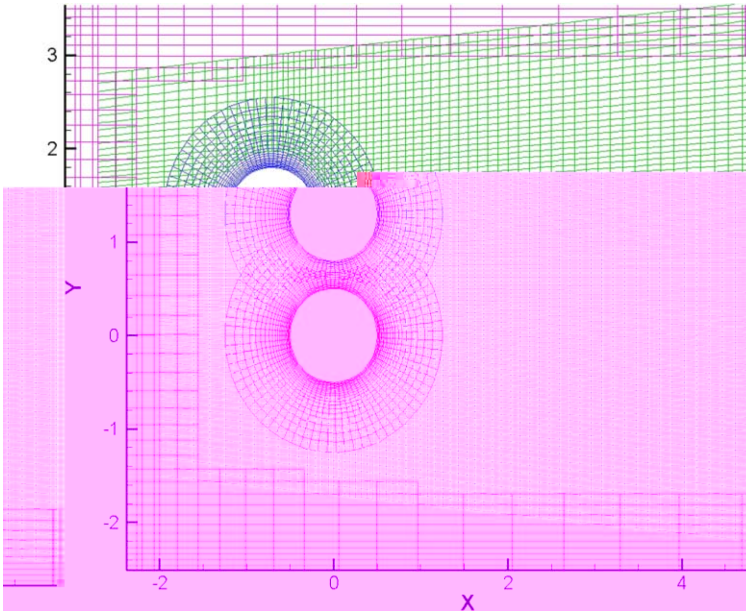

Fig. 20. Close-up of near-cylinder region of overset-grid system for viscous flow over two cylinders showing complex hole boundaries.

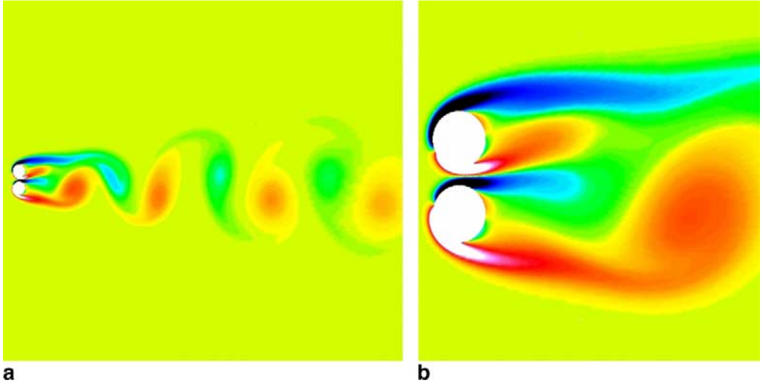

Fig. 21. Instantaneous vorticity field showing interaction between two cylinders: (a) vortex street, (b) close-up of near-cylinder region showing gap bias.

the u-component of velocity to obtain the fundamental frequency of the vortex shedding and any higher harmonic modes that may be present for the particular Reynolds number and gap width. Here, the u-component of velocity obtained via computational means was recorded at these same locations, and the power spectra obtained from these measurements at the lower point ($y = -2.65D$) is shown in Fig. 22. The fundamental frequency given in terms of the non-dimensional Strouhal number was found to be $S_F \approx 0.09$, which matches the experimentally measured value of $S_F \approx 0.09$ for the fundamental harmonic mode at low Reynolds number for this cylinder spacing [42,43]. The presence of higher harmonic modes are also detected at much lower amplitudes than the fundamental frequency. The frequencies of these modes also match those of the higher-order modes detected experimentally at higher Reynolds numbers and/or larger gap widths than was considered here. However, at this low Reynolds number and small gap width, only the

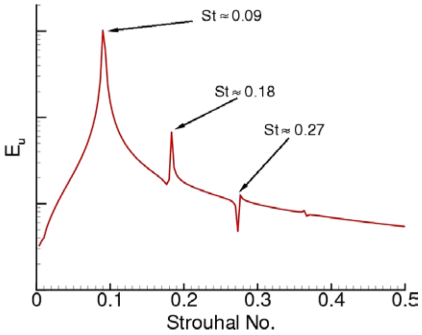

Fig. 22. Power spectrum of u-velocity at point $x = 4D$, $y = -2.65D$ downstream of the cylinders.

fundamental mode was detected experimentally, which, based on the flow visualizations and large difference in amplitude of the various mode peaks, appears to be true computationally as well.

5.4. Acoustic scattering from circular cylinders

The final problem considered here to demonstrate the utility of the high-order overset-grid method is the scattering of waves generated by a time-dependent, spatially distributed acoustic source by single and multiple circular cylinders. The cylinders are assumed to be acoustically rigid, and the acoustic source is represented by the expression

$$S(x,y,t) = \exp\left\{-25\ln 2[(x-x_c)^2 + (y-y_c)^2]\right\}\sin(\omega t)f(t). \tag{18}$$

Here, (x_c, y_c) represents the location of the source center, ω is its frequency, and $f(t)$ is a ramping function used to gradually introduce the source into the computational domain at the onset of the calculations. This problem provides a stringent test for the algorithm and the treatment at boundaries due to the small amplitudes of the acoustic waves and long propagation distances. It also serves as an excellent validation case as analytical solutions may be generated for different source/cylinder arrangements using the procedure outlined in [44].

The scattering of sound from a single cylinder was initially examined to validate the high-order overset-grid approach for computational aeroacoustics. This problem consists of a single cylinder of diameter $D = 1$ located at the origin of the computational domain and insonified by a source with $\omega = 6\pi$ centered at $x_c = 4$, $y_c = 0$. Two grid systems were examined for the single-cylinder problem and are shown in Fig. 23. The first topology, shown in Fig. 23(a), consists of a single polar grid centered on the cylinder and decomposed into multiple blocks for parallel processing. The grid is clustered near the cylinder and is gradually stretched in the radial direction until it reaches a constant grid spacing corresponding to 12 points-per-wave (PPW) at $\omega = 6\pi$. The constant radial spacing persists until the beginning of a rapid grid-stretching region which, in conjunction with the numerical filter, dissipates outgoing acoustic waves and prevents reflection at the far field boundary [45]. The second grid system, shown in Fig. 23(b) is made up of three overset grids, which are also decomposed into blocks for parallel processing. These grids include a body-fitted grid around the cylinder, a uniformly spaced background Cartesian grid, and a highly stretched polar grid in the far-field. A close-up of the overlapping region between the body-fitted and background grids for the multiple-grid cases is shown in Fig. 23(c). The grid points from the near-body polar grid are identical to the corresponding points from the single-grid polar topology, while the spacing in both directions of the Cartesian grid match the constant radial spacing from the single-grid topology.

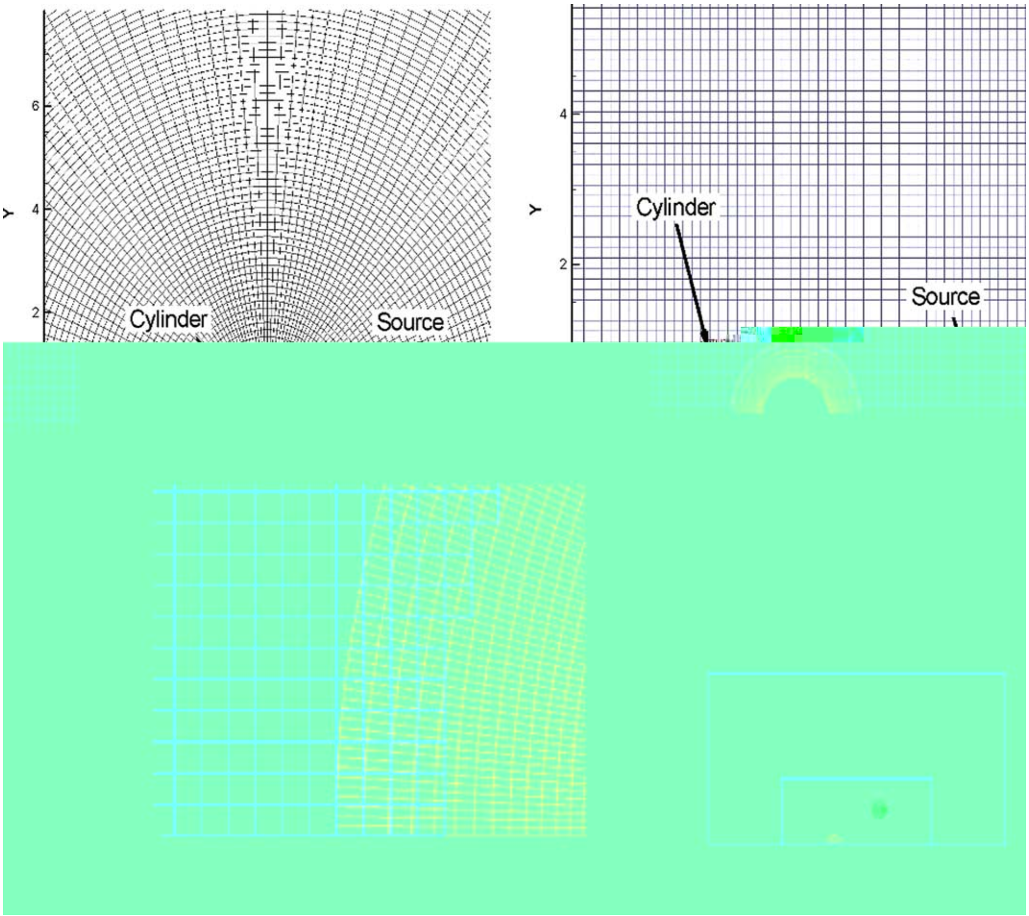

Fig. 23. Overset-grid topologies employed for acoustic scattering simulations: (a) single-cylinder/single-grid topology (every fifth grid line shown for clarity); (b) single-cylinder/multiple-grid topology (every fifth grid line shown for clarity); (c) close-up of overset region between body-fitted and background grids (all gridlines shown); (d) schematic for three-cylinder topology with all gridlines removed.

For all single-cylinder results shown here, a time step of $\Delta t = 0.004$ was used to advance the solution 7000 time steps from a quiescent state using the RK4 time integration method. This number of time steps was found to be sufficient to allow the initial transients to exit the domain and a time-periodic solution to develop. The $C6F10^{0.45}$ scheme was employed, and various values of β were considered to determine the effect that the minimum-order boundary filter has on the accuracy of the algorithm. Fig. 24 shows the root-mean-squared fluctuating pressure (p_{rms}) averaged over the last 1000 time steps of the computation plotted on the surface of the cylinder (Figs. 24(a) and 24(b)) as well as along the centerline behind (with respect to the source) the cylinder (Figs. 24(c) and 24(d)). Results obtained with the $\beta = 2$ boundary filter differ significantly from the analytic solution, with large spikes in the mean pressure observed on the surface of the cylinder corresponding to the block boundaries. While all one-sided ($\beta \geqslant 4$) filters perform well for the grid resolution used here, the performance of the $\beta = 4$ one-sided filter begins to deviate from the analytic solution for coarser grids which dictate the use of the $\beta = 6$ and $\beta = 8$ one-sided filters [46]. No stability issues were encountered for these aeroacoustic simulations at the block boundaries for the one-sided boundary filters of up to eighth-order, the highest case examined here.

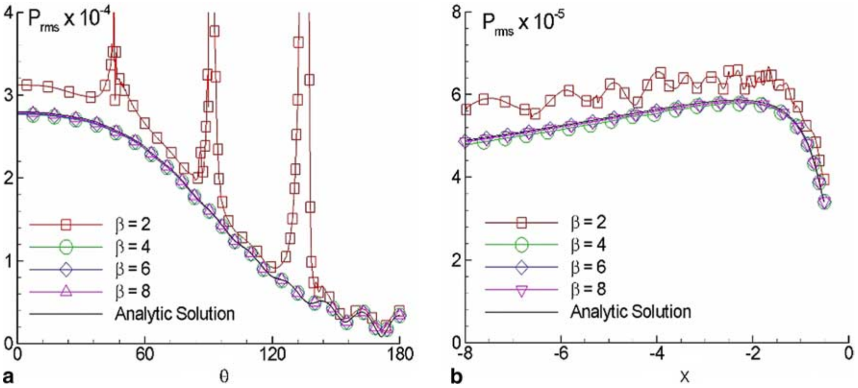

Fig. 24. Effect of variation of minimum-order boundary filter on RMS-pressure for the single-cylinder/single-grid topology with a frequency of $\omega = 6\pi$: (a) surface of cylinder, (b) along centerline to the left of the cylinder (side facing away from source).

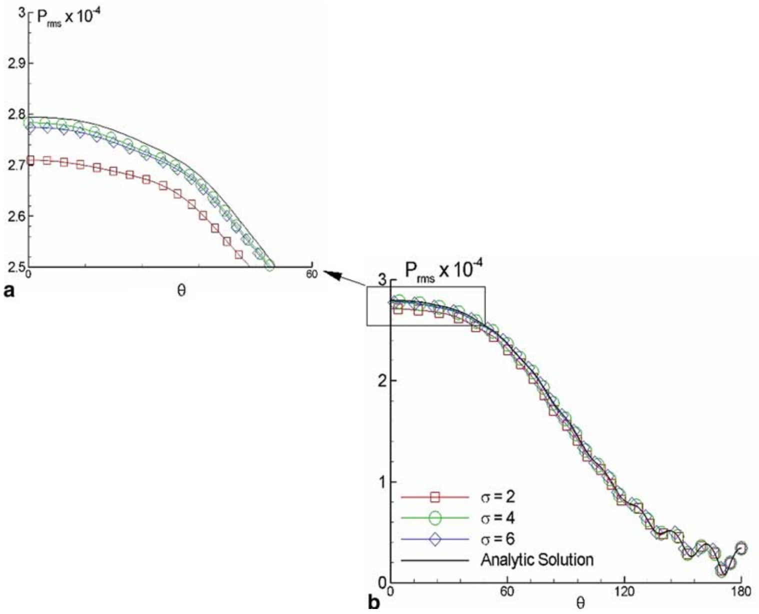

Fig. 25. Order of interpolation effects on RMS-pressure on surface of cylinder for single-cylinder/multiple-grid topology, $\omega = 6\pi$: (a) over entire top half of cylinder, (b) close-up of backscatter region.

In Fig. 25, the root-mean-squared pressure on the part of the cylinder surface facing the source obtained using the multiple-grid topology with $C6F10_8^{0.45}$ and various orders of interpolation is shown at a source frequency of $\omega = 6\pi$. These results are compared to the analytic solution as well as results obtained on the single-grid topology where no interpolation error is present. The results obtained with both the fourth- and sixth-order interpolation are nearly identical to the single-grid results and correspond well to the analytic solution, thus reinforcing the notion that the high-order interpolation adds little additional error to the solution beyond what is created due to the domain decomposition process itself. The fourth-order interpolation method performs slightly better than the sixth-order method, most likely due to the minimal amount of overlap used in this problem which results in more one-sided interpolation stencils for the higher-order method. The results obtained with second-order interpolation capture the overall behavior of the mean pressure, but exhibit additional errors in this region that are removed through the use of the high-order interpolation.

Finally, a three-cylinder configuration with the three identical cylinders of diameter $D = 1$ are located at $(x_o, y_o) = (-4, 0)$, (4, 3), and (4, −3) was simulated for $\omega = 8\pi$ using a symmetric, four-grid overset system consisting of two body-fitted O-type grids around the cylinders in the upper half of the computational domain, a background Cartesian grid, and a rapidly stretched far-field grid as shown in Fig. 23. Both the $C6F10_8^{0.45}$ scheme with sixth-order interpolation as well as an explicit, fourth-order scheme with explicit eighth-order filtering ($E4F8$) and fourth-order interpolation were employed for comparison. The solution was advanced in time 12,000 steps with $\Delta t = 0.004$ to ensure a time-periodic solution was reached. The mean pressure on the surfaces of the left and top-right cylinders as well as on the centerline of the computational domain obtained with the compact solver are nearly indistinguishable from the analytic solution as shown in Fig. 26. However, the $E4F8$ explicit method exhibits considerable error when solved on the same grid, despite the fact that it possesses the same five-point difference stencil as the compact method. In Fig. 26(d), the complex interference pattern set up by the multiple scattering of acoustic waves is shown through the instantaneous total pressure field and demonstrates the smooth transition of the solution across both block (coincident) and grid (non-coincident) boundaries.

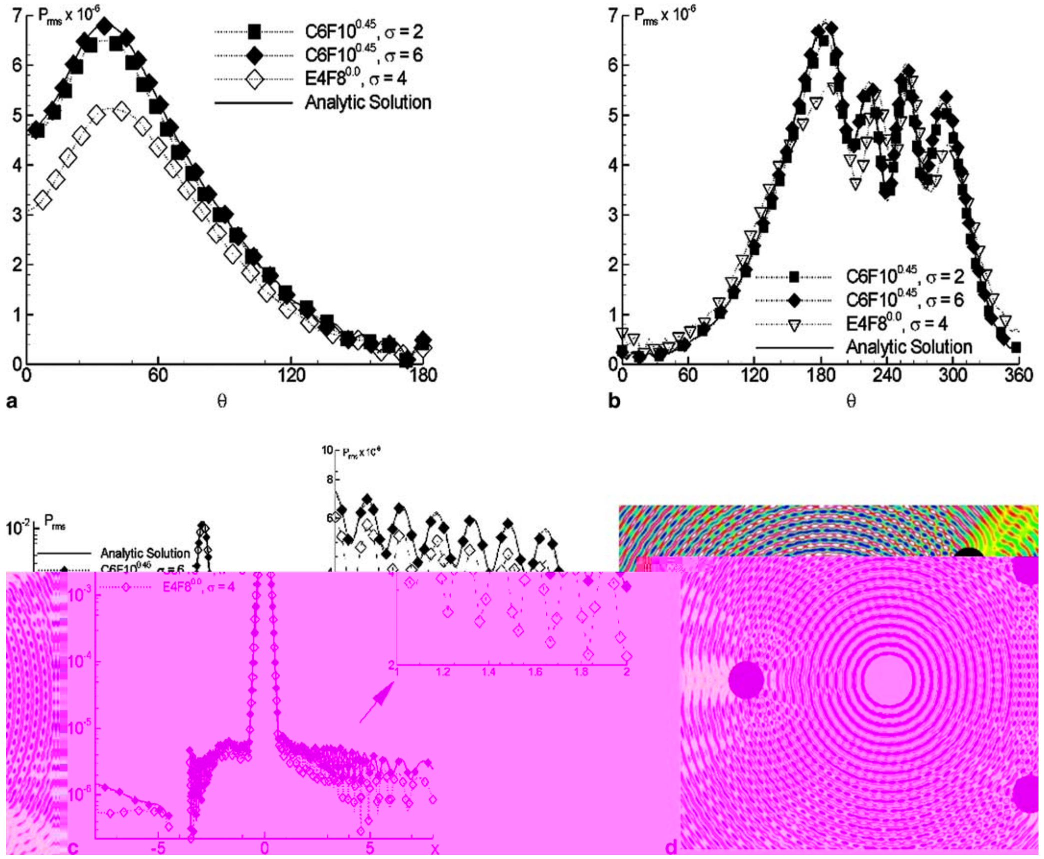

Fig. 26. Results for three-cylinder acoustic scattering simulation: (a) RMS-pressure on surface of left cylinder; (b) RMS-pressure on surface of top right cylinder; (c) RMS-pressure along domain centerline; (d) instantaneous total (incident plus scattered) pressure field.

6. Conclusions

In this work, an algorithm coupling the accuracy of high-order compact finite-difference methods with the flexibility of overset-grid methods has been developed for use in solving the Euler and Navier–Stokes equations on curvilinear grids. This algorithm has been developed for use on massively parallel platforms using the overset-grid approach for domain decomposition. Two key features of this algorithm introduced here were the incorporation of holes in the high-order spatial algorithm and the use of high-order interpolation at non-coincident overlapping boundaries.

Two classes of high-order/high-accuracy interpolation methods were considered for use with the high-order algorithm; a generalized Lagrangian method consisting of optimized or non-optimized methods as well as explicit or compact methods, and a method utilizing basis-splines, or B-splines. Based on the results from a one-dimensional Fourier analysis, implementation of the methods, and results obtained from several benchmark calculations, several general conclusions were drawn regarding the interpolation methods considered. First, while proven to be very effective for spatial differencing and filtering where there is a uniform distribution of points to be operated on, the use of compact Lagrangian interpolation methods does not appear to be suited as an interpolation method for overset-grid applications. This conclusion is based on the complexity of the implementation of the compact approach for interpolating in multiple-dimensions when the interpolation point distribution is not known a priori. Despite the potential for order-of-magnitude reductions in error demonstrated in the one-dimensional Fourier analysis, this improvement was not seen in the various benchmark cases the compact method was applied to. Thus it was felt that the significant increase in complexity required to implement the compact interpolation method was not warranted by the small improvements in the errors obtained in multiple-dimension problems compared to other interpolation methods.

Optimized Lagrangian interpolation methods (both compact and explicit) were also demonstrated in the one-dimensional Fourier analysis to have the potential to reduce error by orders of magnitude compared to equivalent non-optimized methods. However, by lowering the error via optimization in the higher wavenumber range, non-monotonic local increases in the error in the lower wavenumber range develop. Because of this, there will always be some range of the lower portion of wavespace over which optimized methods will possess an integrated error that exceeds the associated non-optimized error. The use of optimized methods will therefore require judicious selection of the optimization parameter r based on the particular problem being considered, the wavenumber range of interest, and the resolution and filtering characteristics of the spatial algorithm. As demonstrated in the analysis of grid-to-grid interpolation of analytic test functions, whether the optimized method resulted in greater or less error than the comparable non-optimized method depended on the wavenumber via the frequencies of the function and the grid resolution. Because of the use of a low-pass filter to remove spurious waves at the higher wavenumbers, it does not appear to be warranted to press for improvement in these regions at the expense of the lower portion of wavespace if the content at the higher wavenumbers will be filtered out anyway. And while the optimized explicit method was relatively straightforward to implement, it does require the solution of a densely populated system of equations to find the interpolation coefficients, compared to the non-optimized method which simply requires the evaluation of analytical expressions for the interpolation coefficients. Currently, the calculation of interpolation coefficients is only done once per interpolation point during pre-processing, and they are stored in an array for use by the interpolation routines. However, if future implementation requires that the coefficients be calculated “on the fly” (to handle grids in relative motion for example), then the inversion required to find the optimized coefficients may become an issue.

The B-spline methods appeared to be very competitive with the generalized Lagrangian methods in both the one-dimensional Fourier analysis as well as the multi-dimensional validation cases when the interpolation points are away from the edges of the interval. However, their poor performance in the outer subintervals of the stencil due to the use of the natural spline boundary conditions would predicate the use of more overlap between grids in order to force the interpolation points towards the center of the interval. This larger overlap translates into additional grid points in the overset topology. The B-spline methods also required more effort to implement compared to explicit methods, although it was much more robust than the compact Langrangian method. Thus, the B-spline approach, and particularly the fourth-order method, could be a viable approach if other boundary conditions besides natural spline boundary conditions are employed to improve performance in the outer subintervals.

Based on the results presented here, the non-optimized, explicit, Lagrangian interpolation approach was the simplest to implement, and proved to be robust and fairly insensitive to location of the interpolation point within the stencil. It possesses many positive features, including analytic expressions for the interpolation coefficients and uniformity of coefficients from one extended interpolation point to the next. It was also shown to converge linearly at or close to the prescribed order-of-accuracy for grid-to-grid interpolation of analytic test functions. Overall, the high-order explicit non-optimized approach offered the best combination of accuracy, robustness and ease of use to make it the preferred choice for use in the current high-order overset-grid flow solver. The use of sixth-order interpolation for the test cases considered here show only marginal improvement, or in some cases a degradation, of solution accuracy compared to fourth-order interpolation. This is most likely attributed to the minimal size of the overlap regions used in several of the cases here, which results in more one-sided stencils for the sixth-order interpolation.

The high-order overset-grid algorithm was successfully employed on several benchmark fluid dynamics problems, including the inviscid convection of a vortex, laminar flow over single and double cylinder configurations, and the scattering of sound from multiple-cylinder configurations. The results obtained for each problem showed consistent improvement when switching from a low-order interpolation method to a high-order method. As was demonstrated in the convecting vortex problem, the use of second-order interpolation results in a globally second-order accurate method, even when a high-order spatial algorithm is employed. The solutions of the acoustic scattering problems show the capability of the new approach for simulating unsteady fluid phenomena associated with complex geometries that demand high accuracy to model. Such an approach would also appear to be well suited for turbulent flow simulations via direct numerical simulation and large-eddy simulation, or other wave-propagation problems such as computational electromagnetics.

Appendix A. Background for generalized Lagrangian interpolation

To generate the interpolation coefficients L_k and R_j for the generalized Lagrangian interpolation approach, a Lagrange multiplier optimization method is used to minimize the error given by Eq. (5) subject to constraints on the formal order-of-accuracy of the resulting method. A σ-order accurate method will satisfy σ constraints arising from the cancellation of terms in the Taylor series expansion of Eq. (1). These order-of-accuracy constraints may be expressed on a uniform grid with a grid spacing of unity as [26]

$$\sum_{k=1}^{\mathrm{BW}} L_k \delta_k^n - \sum_{j=0}^{N-1} R_j j^n = 0, \quad 0 \leqslant n \leqslant \sigma - 1. \tag{A.1}$$

An additional constraint fixes the value of the left-hand side interpolation coefficient in the middle of a BW-wide method to unity,

$$L_{k_c} = 1, \quad \text{where } k_c = \frac{(\mathrm{BW}+1)}{2}. \tag{A.2}$$

The use of Eq. (A.2) allows for the generation of non-trivial solutions to the optimization problem. The fixing of this coefficient could also be explicitly done in Eqs. (5) and (A.1), but is handled through an additional constraint to streamline the notation in the general situation.

The Lagrangian function for the minimization of Eq. (5) subject to the constraints of Eqs. (A.1) and (A.2) may be expressed as

$$\mathscr{L} = \int_0^{r\pi} \left| \sum_{k=1}^{\mathrm{BW}} L_k \mathrm{e}^{\imath\delta_k\alpha} - \sum_{j=0}^{N-1} R_j \mathrm{e}^{\imath j\alpha} \right|^2 \mathrm{d}\alpha + \sum_{n=0}^{\sigma-1} \lambda_n \left(\sum_{k=1}^{\mathrm{BW}} L_k \delta_k^n - \sum_{j=0}^{N-1} R_j j^n \right) + \mu(L_{k_c} - 1), \tag{A.3}$$

where λ_n and μ are the Lagrangian multipliers. The optimized interpolation coefficients are found by solving the linear system of equations that arises by taking the derivatives of Eq. (A.3) with respect to L_k, R_j, and the Lagrangian multipliers. This linear system may be written as

$$[A]\vec{\theta} = \vec{b}, \tag{A.4}$$

where the solution vector $\vec{\theta}$ containing the interpolation coefficients and the Lagrangian multipliers and the right-hand side vector $\vec{b}$ is given by

$$\vec{\theta} = [R_0 \cdots R_{N-1} | L_1 \cdots L_{\mathrm{BW}} | \lambda_0 \cdots \lambda_{\sigma-1} | \mu_1]^{\mathrm{T}} \tag{A.5}$$

and

$$\vec{b} = [0 \cdots 0 | 0 \cdots 0 | 0 \cdots 0 | 1]^{\mathrm{T}}. \tag{A.6}$$

The matrix $[A]$ is symmetric and may be represented by submatrices of various dimensions by

$$[A] = \left[\begin{array}{c|c|c|c} \mathscr{A}_1 & \mathscr{A}_2 & \mathscr{A}_3 & 0 \\ \hline \mathscr{A}_2^{\mathrm{T}} & \mathscr{A}_4 & \mathscr{A}_5 & \mathscr{A}_6 \\ \hline \mathscr{A}_3^{\mathrm{T}} & \mathscr{A}_5^{\mathrm{T}} & 0 & 0 \\ \hline 0 & \mathscr{A}_6^{\mathrm{T}} & 0 & 0 \end{array} \right] \tag{A.7}$$

with the elements of each submatrix given by

$$\begin{aligned} &[\mathscr{A}_1]_{i,j} = \frac{2\sin[(j-i)r\pi]}{j-i}, \quad [\mathscr{A}_2]_{i,j} = -\frac{2\sin\left[(\delta_j - i + 1)r\pi\right]}{\delta_j - i + 1}, \\ &[\mathscr{A}_3]_{i,j} = -\frac{(i-1)^{(j-1)}}{(j-1)!}, \quad [\mathscr{A}_4]_{i,j} = \frac{2\sin\left[(\delta_i - \delta_j)r\pi\right]}{\delta_i - \delta_j}, \\ &[\mathscr{A}_5]_{i,j} = \frac{\delta_i^{(j-1)}}{(j-1)!}, \quad [\mathscr{A}_6]_i = \begin{cases} 0, & i \neq k_c, \\ 1, & i = k_c, \end{cases} \end{aligned} \tag{A.8}$$

where 0! = 1 and $\frac{\sin(xr\pi)}{x}|_{x=0} = r\pi$. Gaussian elimination with partial pivoting is used to solve this system of equations to obtain values for L_k and R_j. For optimized methods, this is done numerically, but for the non-optimized case, analytical expressions for the interpolation coefficients may be found. For the BW = 1 case, the coefficients are given by

$$R_j = \alpha_j \prod_{\substack{i=0 \\ i\neq j}}^{N-1} (\delta_1 - i) \quad L_k = 1, \tag{A.9}$$

where

$$\alpha_j = \frac{(-1)^{N+j-1}}{[N-(j+1)]!j!} \tag{A.10}$$

and $L_1 = 1$. The non-optimized, tridiagonal (BW = 3) coefficients are given by

$$R_j = \alpha_j \frac{(\delta_1 - \delta_2)(\delta_3 - \delta_2)}{(\delta_1 - j)(\delta_3 - j)} \prod_{\substack{i=0 \\ i \neq j}}^{N-1} (\delta_2 - i), \tag{A.11}$$

$$L_k = \frac{(2-\hat{k})(\delta_2 - \delta_{\hat{k}})}{(\delta_1 - \delta_3)} \prod_{i=0}^{N-1} \frac{(\delta_2 - i)}{(\delta_k - i)} \quad \text{for } k = 1,3, \tag{A.12}$$

where α_j is given by Eq. (A.10), $L_2 = 1$, and

$$\hat{k} = \begin{cases} 1 & \text{for } k = 3, \\ 3 & \text{for } k = 1. \end{cases} \tag{A.13}$$

Appendix B. Background for B-spline interpolation

Consider a uniformly spaced, N-point stencil such as shown in Fig. 1 with $N - 1$ subintervals between grid points labelled from 1 to $N - 1$. The grid points making up this stencil correspond to a subset of a knot point sequence T. It will be assumed here that no interior knot point is repeated, and thus there will be $M = N + \sigma - 2$ B-splines of order σ that form a spanning set of the piecewise-continuous polynomial space over the stencil. The jth B-spline of such a set is denoted as B_j^σ and is defined by the recursive relationship [32]

$$B_j^\sigma(x) = \frac{x - t_j}{t_{j+\sigma-1} - t_j} B_j^{\sigma-1}(x) + \frac{t_{j+\sigma} - x}{t_{j+\sigma} - t_{j+1}} B_{j+1}^{\sigma-1}(x), \tag{B.1}$$

where the first-order B-spline is the top-hat function,

$$B_j^1(x) = \begin{cases} 1, & t_j \leqslant x < t_{j+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{B.2}$$

and t_j is the jth knot point of the sequence $T = \{t_1, t_2, \ldots, t_K\}$ containing K knot points. An important property of B-splines is that they have compact support,

$$B_j^\sigma(x) = 0 \quad \text{for } x \in [t_j, t_{j+\sigma-1}] \tag{B.3}$$

and thus are non-zero only over σ contiguous subintervals. This results in banded matrices with a bandwidth of BW = $\sigma - 1$ that may be stored and solved efficiently. B-splines also form a partition of unity,

$$\sum_{j=1}^{M} B_j^\sigma(x) = 1 \quad \text{for } x \in [x_0, x_{N-1}]. \tag{B.4}$$

See [32,47,33,48] for examples of the resulting basis functions for both periodic and bounded domains using the above definitions. Because of the assumption that all interior knot points have a multiplicity of one, the piecewise polynomial that is generated will be $C^{\sigma-2}$ (the maximum possible continuity) as it crosses each knot point.

From the definition of B-splines given in Eq. (B.1), it is observed that additional knot points are needed at each end of the stencil. For the situation where no interior knot points are repeated, the total number of points in the knot point sequence that defines the B-splines is given by $K = M + \sigma$. Of these, N knot points come from the stencil points themselves, and thus an additional $2\sigma - 2$ knot points are required. For periodic problems, these additional knot points may be found by exploiting the periodicity, but this approach is of limited use in general applications and will not be considered here. In general, the placement of the additional knot points will determine the near-boundary basis functions, but they do not affect the spline solution space and thus their specific location is arbitrary [33]. The typical approach is to treat the endpoints of the stencil as knot points of multiplicity σ, which results in the knot points defined as

$$t_j = \begin{cases} 0, & 1 \leqslant j \leqslant \sigma, \\ j - \sigma, & \sigma + 1 \leqslant j \leqslant N + \sigma - 1, \\ N - 1, & N + \sigma \leqslant j \leqslant K, \end{cases} \tag{B.5}$$

where a uniform grid with grid spacing of unity is assumed.

To utilize B-splines as interpolating functions, a general interpolation equation for a single interpolation point located at $\hat{x} = x_0 + \delta$ is first written as

$$\hat{f}(\hat{x}) = \sum_{j=I}^{I+\sigma-1} \gamma_j B_j^{\sigma}(\hat{x}). \tag{B.6}$$

In the above equation, I is the subinterval number ranging from 1 to N containing $\hat{x}$ and the compactness property as expressed in Eq. (B.3) is exploited by performing the summation only over those B-splines that are non-zero. The unknown coefficients γ_j are determined by requiring the known functional value at each grid point to be reproduced when the grid point location x_i is inserted into the right-hand side of Eq. (B.6). The resulting N equations generated by this requirement are given by

$$\sum_{j=1}^{M} B_j^{\sigma}(x_i)\gamma_j = f(x_i), \quad i = 0, \ldots, N-1, \tag{B.7}$$

where $B_j^{\sigma}(x_i)$ and $f(x_i)$ are known quantities and the coefficients γ_j are to be determined. However, these N equations do not form a consistent linear system for the M unknown values of γ_j, and thus an additional $\sigma - 2$ equations are required to close the system. Several approaches exist to do this [32], but the only method considered in this work is the natural spline case approach. In this case, $(\sigma - 2)/2$ derivatives of the resulting piecewise interpolating polynomial are set equal to zero at each boundary of the stencil. While any order derivative less than or equal to $\sigma - 2$ may be set in order to obtain the necessary equations, here only the highest-order derivative(s) possible will be used. Thus the second-derivative at each end of the stencil will be fixed for the $\sigma = 4$ case and both the third- and fourth-derivatives will be set when $\sigma = 6$. These boundary conditions may be expressed in a general fashion as

$$\frac{\partial^n \hat{f}}{\partial x^n} = 0, \quad n = \frac{\sigma}{2}, \ldots, \sigma - 2. \tag{B.8}$$

The derivative boundary condition equations are incorporated into the system of equations in such a way as to maintain the banded structure of the matrices. The resulting linear system may be written as

$$[\beta]\vec{\gamma} = \vec{\phi}, \tag{B.9}$$

where $\vec{\gamma}$ is the vector consisting of the unknown coefficients. The B-spline matrix $[\beta]$ is constructed using Eq. (B.7) by evaluating each B-spline at the knot points and incorporating the necessary boundary condition equations from Eq. (B.8) and is given in general form by

$$[\beta] = \begin{bmatrix} c_1 & b_{M-1} & a_{M-2} & 0 & \cdots & 0 \\ b_1 & c_2 & b_{M-2} & a_{M-3} & \ddots & \vdots \\ a_1 & b_2 & c_3 & b_{M-3} & \ddots & 0 \\ 0 & a_2 & b_3 & c_4 & \ddots & a_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & b_1 \\ 0 & \cdots & 0 & a_{M-2} & b_{M-1} & c_M \end{bmatrix}, \tag{B.10}$$

where the entries on the diagonals are given by

$\sigma = 4$

$$\begin{aligned} \vec{a} &= \{\vec{0}\}, \\ \vec{b} &= \left\{6, \frac{1}{4}, \frac{1}{6}, \ldots, \frac{1}{6}, 3, 0\right\}, \\ \vec{c} &= \left\{1, -9, \frac{7}{12}, \frac{2}{3}, \ldots, \frac{2}{3}, \frac{7}{12}, -9, 1\right\}, \end{aligned} \tag{B.11}$$

$\sigma = 6$

$$\begin{aligned} \vec{a} &= \left\{120, \frac{1}{16}, \frac{1}{54}, \frac{1}{96}, \frac{1}{120}, \ldots, \frac{1}{120}, 5, -10, 0\right\}, \\ \vec{b} &= \left\{-60, -225, \frac{175}{432}, \frac{7}{27}, \frac{103}{480}, \frac{13}{60}, \ldots, \frac{13}{60}, \frac{163}{1440}, -\frac{125}{3}, 55, 0\right\}, \\ \vec{c} &= \left\{1, 105, \frac{425}{3}, \frac{355}{864}, \frac{179}{360}, \frac{11}{20}, \ldots, \frac{11}{20}, \frac{179}{360}, \frac{355}{864}, \frac{425}{3}, -105, 1\right\}. \end{aligned} \tag{B.12}$$

Likewise from Eq. (B.7), the vector $\vec{\phi}$ is given by the known functional values at each knot point plus the additional equations obtained from the boundary conditions and is given by

$\sigma = 4$

$$\vec{\phi} = [f_0 \quad 0 \quad f_1 \quad f_2 \quad \cdots \quad f_{N-3} \quad f_{N-2} \quad 0 \quad f_{N-1}]^{\mathrm{T}}, \tag{B.13}$$

$\sigma = 6$

$$\vec{\phi} = [f_0 \quad 0 \quad 0 \quad f_1 \quad f_2 \quad \cdots \quad f_{N-3} \quad f_{N-2} \quad 0 \quad 0 \quad f_{N-1}]^{\mathrm{T}}, \tag{B.14}$$

where the zeros near each end these vectors show where the additional equations provided by Eq. (B.8) were inserted into the linear system.

Upon solving Eq. (B.9), the collocation coefficients are given in general form as

$$\vec{\gamma} = [\beta]^{-1}\vec{\phi}. \tag{B.15}$$

In terms of the functional values at the grid points, the above equation may be rewritten for each individual coefficient γ_i as

$$\gamma_i = \sum_{j=0}^{N-1} [\beta]^*_{i,j+1} f(x_j). \tag{B.16}$$

Note that the matrices $[\beta]^{-1}$ and $[\beta]^*$ differ by the presence of $\sigma - 2$ additional columns in $[\beta]^{-1}$ due to the inclusion of the boundary condition equations, and thus $[\beta]^*$ is formed by solving for $\vec{\gamma}$ and then extracting the appropriate coefficients. Inserting Eq. (B.16) back into the general B-spline interpolation formula given by Eq. (B.6) yields the expression

$$\hat{f}(\hat{x}) = \sum_{i=I}^{I+\sigma-1} \sum_{j=0}^{N-1} [\beta]^*_{i,j+1} f(x_j) B_{i,\sigma}(\hat{x}) \tag{B.17}$$

which, after switching the order of the summation and regrouping, may be written as

$$\hat{f}(\hat{x}) = \sum_{j=0}^{N-1} \left\{ \sum_{i=I}^{I+\sigma-1} [\beta]^*_{i,j+1} B_{i,\sigma}(\hat{x}) \right\} f(x_j). \tag{B.18}$$

It is observed that the above equation is equivalent to the general interpolation formula given by Eq. (1), with $L_K = 1$ and

$$R_j = \sum_{i=I}^{I+\sigma-1} [\beta]^*_{i,j+1} B_{i,\sigma}(\hat{x}). \tag{B.19}$$

In this form, the Fourier error analysis given by Eqs. (2) and (3) may be applied to compare the accuracy of B-spline interpolation method with the Lagrangian approach. Also, while the above expressions are explicit in the sense that only a single interpolation point is solved for at a time, the matrix $[\beta]^*$ needs to be formulated only once for all interpolation points that lie within the boundaries of a stencil. Interpolated data may then be obtained by evaluating $B_i^{\sigma}(\hat{x})$ at each interpolation point location $\hat{x}$ within the stencil in turn.

References

[1] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, New York, 1987.

[2] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[3] C.K. Tam, J. Webb, Dispersion-relation-preserving finite difference schemes for computational aeroacoustics, J. Comput. Phys. 107 (1993) 262–281.

[4] R. Hirsch, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.

[5] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds numbers, J. Fluid Mech. 177 (1987) 133–166.

[6] J. Benek, J. Steger, F. Dougherty, A flexible grid embedding technique with application to the Euler equations, AIAA Paper 83-1944, 1983.

[7] J. Benek, P. Buning, J. Steger, A 3-D chimera grid embedding technique, AIAA Paper 85-1523, 1985.

[8] J.L. Steger, F.C. Dougherty, J.A. Benek, A chimera grid scheme, in: K. Ghia, U. Ghia (Eds.), Advances in Grid Generation, ASME-FED-Vol. 5, The American Society of Mechanical Engineers, 1983, pp. 59–69.

[9] S.M. Murman, Y.M. Rizk, L.B. Schiff, Coupled numerical simulation of the external and engine inlet flows for the F-18 at large incidence, Aircraft Design 3 (2000) 65–77.

[10] R.J. Gomez, D. Vicker, S.E. Rogers, M.J. Aftosmis, W.M. Chan, R. Meakin, S. Murman, STS-107 investigation ascent CFD support, AIAA Paper 2004-2226, AIAA 34th Fluid Dynamics Meeting Portland OR, June 2004.

[11] N.C. Prewitt, D.M. Belk, W. Shyy, Parallel computing of overset grids for aerodynamic problems with moving objects, Prog. Aero. Sci. 36 (2000) 117–172.

[12] K.-H. Kao, M.-S. Liou, C.-Y. Chow, Grid adaptation using chimera composite overlapping meshes, AIAA J. 32 (5) (1994) 942–949.

[13] R.L. Meakin, Adaptive spatial partitioning and refinement for overset structured grids, Comput. Methods Appl. Mech. Engrg. 189 (2000) 1077–1117.

[14] P.E. Morgan, M.R. Visbal, K. Tomko, Chimera-based parallelization of an implicit Navier–Stokes solver with applications, AIAA Paper 2001-1088, AIAA 39th Aerospace Sciences Meeting Reno NV, January 2001.

[15] D.M. Belk, The role of overset grids in the development of the general purpose CFD code, in: Surface modeling, grid generation, and related issues in computational fluid dynamics (CFD) solutions, NASA CP-3291, 1995, pp. 193–204.

[16] M.R. Visbal, D.V. Gaitonde, High-order-accurate methods for complex unsteady subsonic flows, AIAA J. 37 (10) (1999) 1231–1239.

[17] D.V. Gaitonde, M.R. Visbal, Padé-type higher-order boundary filters for the Navier–Stokes equations, AIAA J. 38 (11) (2000) 2103–2112.

[18] M.R. Visbal, D.V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, J. Comput. Phys. 181 (2002) 155–185.

[19] J. Yin, J. Delfs, Sound generation from gust–airfoil interaction using caa-chimera method, AIAA Paper 2001-2136, 7th AIAA/CEAS Aeroacoustics Conference Maastricht Netherlands, May 2001.

[20] J.W. Delfs, An overlapped grid technique for high resolution caa schemes for complex geometries, AIAA Paper 2001-2199, 7th AIAA/CEAS Aeroacoustics Conference Maastricht Netherlands, May 2001.

[21] J. Benek, J. Steger, F. Dougherty, P. Buning, Chimera: A grid-embedding technique, Technical Report AEDC-TR-85-64, Arnold Engineering Development Center, April 1986.

[22] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1990) 1–64.

[23] S.E. Sherer, M.R. Visbal, Implicit large eddy simulations using a high-order overset grid solver, AIAA Paper 2004-2530, AIAA 34th Fluid Dynamics Meeting Portland OR, June 2004.

[24] R. Vichnevetsky, J.B. Bowles, Fourier Analysis of Numerical Approximations of Hyperbolic Equations, Society of Industrial and Applies Mathematics, Philadelphia, PA, 1982.

[25] C.K. Tam, K.A. Kurbatskii, A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes, J. Comput. Phys. 157 (2) (2000) 588–617.

[26] S.E. Sherer, Investigation of high-order and optimized interpolation methods with implementation in a high-order overset grid CFD code, Ph.D. Thesis, The Ohio State University, 2002.

[27] S. Rubin, P. Khosla, Higher-order numerical solutions using cubic splines, AIAA J. 14 (7) (1976) 851–858.

[28] S. Rubin, P. Khosla, Polynomial interpolation methods for viscous flow calculations, J. Comput. Phys. 24 (1977) 217–244.

[29] A. Kravchenko, P. Moin, R. Moser, Zonal embedded grids for numerical simulations of wall-bounded turbulent flows, J. Comput. Phys. 127 (1996) 412–423.

[30] A. Kravchenko, P. Moin, Numerical studies of flow over a circular cylinder at $Re_D = 3900$, Phys. Fluids 12 (2) (2000) 403–417.

[31] A. Chaniotis, D. Poulikakos, High-order interpolation and differentiation using B-splines, J. Comput. Phys. 197 (2004) 253–274.

[32] C. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

[33] W.Y. Kwok, R.D. Moser, J. Jimenez, A critical evaluation of the resolution properties of B-spline and compact finite difference methods, J. Comput. Phys. 174 (2001) 510–551.

[34] D.V. Gaitonde, M. Visbal, High-order schemes for Navier–Stokes equations: algorithm and implementation into FDL3DI, Technical Report AFRL-VA-WP-TR-1998-3060, Air Vehicles Directorate, Air Force Research Laboratory, August 1998.

[35] D. Fyfe, Economical evaluation of Runge–Kutta formulae, Math. Comput. 20 (1966) 392–398.

[36] R. Beam, R. Warming, An implicit factored schemes for compressible Navier–Stokes equations, AIAA J. 16 (4) (1978) 393–402.

[37] T.H. Pulliam, D. Chaussee, A diagonal form of an implicit approximate-factorization algorithm, J. Comput. Phys. 29 (2) (1981) 347–363.

[38] T.H. Pulliam, Artificial dissipation models for the Euler equations, AIAA J. 24 (12) (1986) 1931–1940.

[39] B. Carnahan, H. Luther, J.O. Wilkes, Applied Numerical Methods, John Wiley, New York, 1969.

[40] P. Morgan, M.R. Visbal, D. Rizzetta, A parallel high-order flow solver for LES and DNS, AIAA Paper 2002-3123, AIAA 32th Fluid Dynamics Meeting, St. Louis, MO, June 2002.

[41] A. Kravchenko, P. Moin, K. Shariff, B-spline method and zonal grids for simulations of complex turbulent flows, J. Comput. Phys. 151 (1999) 757–789.

[42] C. Williamson, Evolution of a single wake behind a pair of bluff bodies, J. Fluid Mech. 159 (1) (1985) 1–18.

[43] S. Xu, Y. Zhou, R. So, Reynolds number effects on the flow structure behind two side-by-side cylinders, Phys. Fluids 15 (5) (2003) 1214–1219.

[44] S.E. Sherer, Scattering of sound from a axisymmetric sources by multiple circular cylinders, J. Acoust. Soc. Am. 115 (2) (2004) 488–496.

[45] M.R. Visbal, D.V. Gaitonde, Very high-order spatially implicit schemes for computational acoustics on curvilinear meshes, J. Comput. Acoust. 9 (4) (2001) 1259–1286.

[46] S. Sherer, M. Visbal, High-order overset-grid simulations of acoustic scattering from multiple cylinders, in: Proceedings of the Fourth Computational Aeroacoustics (CAA) Workshop on Benchmark Problems, NASA/CP-2004-212954, pp. 255–266.

[47] A.G. Kravchenko, P. Moin, B-spline methods and zonal grids for numerical simulation of turbulent flows, Technical Report TF-73, Department of Mechanical Engineering, Stanford University, Stanford, CA, February 1998.

[48] O. Botella, K. Shariff, B-spline methods in fluid dynamics, Int. J. Comput. Fluid Dyn. 17 (2) (2003) 133–149.