AIAA JOURNAL
Vol. 39, No. 3, March 2001

Automated Approach to Very High-Order Aeroacoustic Computations

Rodger W. Dyson* and John W. Goodrich*
NASA John H. Glenn Research Center at Lewis Field, Cleveland, Ohio 44135

Computational aeroacoustics requires efficient, high-resolution simulation tools. For smooth problems, this is best accomplished with very high-order in space and time methods on small stencils. However, the complexity of highly accurate numerical methods can inhibit their practical application, especially in irregular geometries. This complexity is reduced by using a special form of Hermite divided-difference spatial interpolation on Cartesian grids, and a Cauchy–Kowalewski recursion procedure for time advancement. In addition, a stencil constraint tree reduces the complexity of interpolating grid points that are located near wall boundaries. These procedures are used to develop automatically and to implement very high-order methods (>15) for solving the linearized Euler equations that can achieve less than one grid point per wavelength resolution away from boundaries by including spatial derivatives of the primitive variables at each grid point. The accuracy of stable surface treatments is currently limited to 11th order for grid aligned boundaries and to 2nd order for irregular boundaries.

I. Introduction

NOISE generation and propagation are difficult to simulate numerically for a variety of well-documented reasons and require high-order numerical schemes.[1,2] However, high-order schemes can introduce a number of complications, such as the following:

1) Large stencils near boundaries, with either many fictitious grid points, or large one-sided stencils, introduce programming complexity and numerical instability.[3,4]

2) High-order finite difference equations can require boundary treatments beyond the physical conditions of the original problem, which can excite spurious waves and instabilities.[5]

3) Generation of high-order, smooth, body-fitted grids around complex configurations can be difficult.[6]

4) High-order formulations can lack nonlinear robustness.[6]

5) The general usefulness of high-order methods is limited by first-order accurate shock capturing.[7]

The approach described in this paper will focus on the first three issues. The fourth issue is being investigated.[8] The last issue may possibly be avoided by including the physical viscosity and resolving the steep gradients,[9] but with very high-resolution methods.

In addressing the first three concerns, we limit ourselves to wave propagation and scattering problems governed by the linearized Euler equations. Previous studies point out the advantages of high-order methods for acoustical propagation.[10–15] Many methods in general use stop at fourth-order accuracy for time-dependent problems because they use Runge–Kutta methods. High-order Runge–Kutta methods become notoriously difficult to derive because the number of nonlinear order conditions that need to be solved grows exponentially, that is, a 12th-order method has 7813 nonlinear order conditions.[16–21] The advantages of using Runge–Kutta methods at orders less than 6 are commonly cited as flexibility, large stability limits, and ease of programming.[22] The practical limit on their order has been an impediment to the analysis of their use in high-order approaches for time-dependent acoustic applications.[23]

In this paper we use a series of explicit, local, high-order methods that have the same order of accuracy in space as in time.[24,25] These methods use Hermite interpolation on stencils that are two points wide and a Cauchy–Kowalewski recursive procedure[26] for obtaining time derivatives from the space derivatives of the interpolant. The time derivatives are then used to advance the primitive variables and their spatial derivatives in time with a Taylor series expansion. This general approach is called the modified expansion solution approximation (MESA) method.[11] This method can be used to derive and implement algorithms with arbitrarily high orders of accuracy in multiple space dimensions if their complexity is properly managed and the computer's floating point precision is sufficiently high.[27]

The complex task of developing and coding a multidimensional interpolant for each MESA scheme can be eliminated using the tensor product[28] of a new divided-difference form of Hermitian spatial interpolation on a two-point stencil.[27] The task of obtaining time derivatives with Cauchy–Kowalewski recursion[26] can be implemented by unrolling the recursion.[29] With these techniques, very high-order MESA schemes may be implemented using only a few pages of code, and their accuracy may be adjusted by merely changing loop indices, not the code.[27] This in turn enables local solution adaptive order changes at each time step.

Most unsteady flow simulators using Cartesian grids in complex geometries are based on a finite volume approach[30] with the exception of the work by Kurbatskii and Tam,[31] which specifically investigates acoustic scattering using a fourth-order accurate in time finite difference approach. In their work, curved surfaces are approximated with linear segments and ghost points are used to enforce boundary conditions, all of which incurred a fair amount of programming complexity to implement.[31]

The approach presented here, however, does not need to approximate the geometry and the programming task is completed automatically using computer algebra[29] in a manner similar to the works of Wirth[32] and Steinberg and Roache.[33] This is accomplished by viewing the boundary conditions as applying uniformly in time throughout the boundary surface. The boundary conditions can be differentiated in the surface and in time, and the governing equations can be used to obtain an infinite number of constraints in the boundary.[4] A stencil constraint tree is used to simplify the task of symbolically imposing these high-order surface boundary conditions.[29]

With these techniques, the accuracy of interior propagation seems limited only by the floating point precision available. However, we stably attained only 11th-order accuracy for acoustic scattering in a box with sides that are aligned to the grid, and we attained only 2nd-order accuracy in more general cases. These limits are due to poorly conditioned matrix systems and numerical instability. However, these effects are dependent on the choice of boundary conditions and require further study as will be shown.

The objective of this paper is to present and validate a new approach to aeroacoustic computations in complex geometries that has the potential to fully utilize the precision of today's computers.

Received 5 April 2000; presented as Paper 2000-2006 at the AIAA/CEAS 6th Aeroacoustics Conference, Lahaina, HI, 12–14 June 2000; revision received 7 September 2000; accepted for publication 27 September 2000.

*Aerospace Engineer, Acoustics Branch, Structures and Acoustics Division, Research and Technology Directorate.

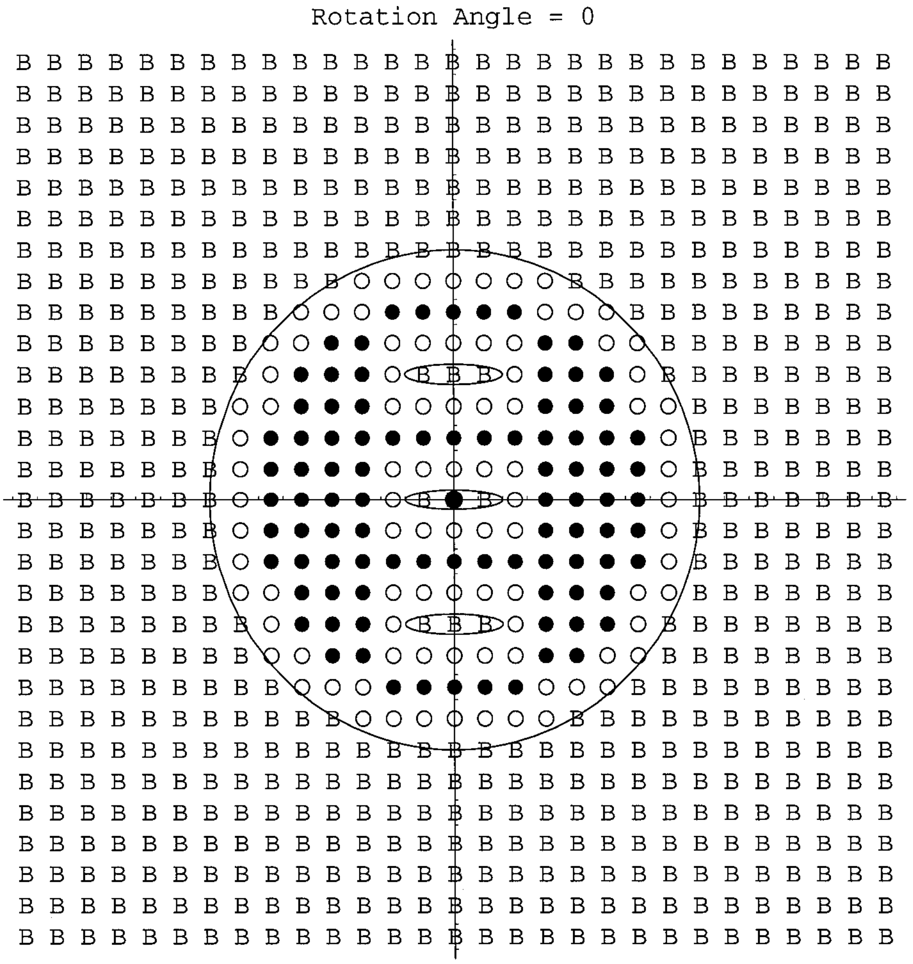


Fig. 1 Cascade superimposed on grid.

A common theme in this paper will be the many advantages of using a two-point wide stencil that include very high resolution, solution adaptability, and Courant–Friedrichs–Lewy (CFL) stability near boundaries. One of the more intriguing results to be presented here is the possibility of subgrid-scale resolution using solution adaptable algorithms.

II. Governing Equations

We will demonstrate this new approach by solving the linearized Euler equations in a uniform mean flowfield with mean velocity $\boldsymbol{M}$, perturbation velocity $\boldsymbol{u}$, pressure p, and Cartesian coordinates x_i. We assume the initial conditions are known and that no additional sources are present. The conservation of momentum and energy equations are

$$\frac{\partial u_j}{\partial t} + M_i \frac{\partial u_j}{\partial x_i} + \frac{\partial p}{\partial x_j} = 0, \qquad \frac{\partial p}{\partial t} + M_i \frac{\partial p}{\partial x_i} + \frac{\partial u_j}{\partial x_j} = 0 \quad (1)$$

and are nondimensionalized with respect to the following scales: length scale L, velocity scale c (sound speed), timescale L/c, density scale ρ_0 (ambient density), and pressure scale $\rho_0 c^2$. Also note that the continuity equation is not included because it is not necessary for calculating the acoustic response.

We will use $2s + 1$-order explicit MESA methods, where s can be any nonnegative integer. We will use only two-point stencils, where each grid point contains the primitive solution variables p and u_i and their spatial derivatives (Hermitian data), for a total of $3(s+1)^2$ or $4(s+1)^3$ terms per grid point in two or three spatial dimensions, respectively. For example, in two spatial dimensions with $s = 1$, we have a third-order method that uses the following data on each grid point: p, u, v, p_x, u_x, v_x, p_y, u_y, v_y, p_{xy}, u_{xy}, and v_{xy}.

III. Grid Classification

Generally, we are interested in simulating the acoustical scattering from objects that are defined parametrically and that are superimposed on a uniform Cartesian grid, as shown in Fig. 1 for a cascade of three airfoils. Notice that each grid point is labeled with a closed circle, an open circle, or the letter B, and represents an interior, a fill, or a boundary grid point, respectively. In Fig. 1, the assumed stencil size is three points per dimension, and a fill grid point is simply one in which one of its neighboring grid points is either on or beyond the boundary. It is referred to as a fill because that grid point cannot be computed directly and needs to be filled with data. If none of the neighboring grid points is on or beyond the boundary, then it is classified as an interior grid point and can be computed normally. All other grid points are considered boundary points and are not needed because ghost grid points are not used.

Grid point classification proceeds by first labeling all grid points as boundary type and then finding the interior and fill-type grid points with a simple recursive procedure. The procedure determines if any neighboring grid points are boundary points by checking if the surface of any object intersects the imaginary line joining the center point to its neighbor. Because we are using a finite difference method, we need to check only for line-to-surface intersections, which is much simpler than determining the surface-to-surface intersections necessary if a finite volume method is used. If there is no intersection, then the current grid point is a fill point and recursion stops. Otherwise it is an interior point, and the procedure is called again, but starting with the neighboring grid points.

IV. Advancing the Solution Using MESA

With the uniform Cartesian grid points correctly labeled and the initial data assigned to all of the interior and fill grid points, the next step is to advance the solution in time. Because interior grid points

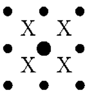


Fig. 2 Staggered grid using two-point stencil.

and their stencils are completely contained within the computational domain, they will be advanced directly using an efficient form of the MESA schemes. However, because a fill point's stencil reaches outside the computational domain field, its data will be obtained by spatial interpolation. At each time step, every grid point is first treated as an interior point to simplify the looping structure for efficient vector and parallel operations, and then the relatively few fill points are recomputed using interpolation.

Advancing the interior points with MESA is a three-step process. The spatial derivatives at the center of the stencil are approximated using spatial interpolation, the spatial derivatives are converted to time derivatives, and the solution is advanced in time with a Taylor series. The MESA scheme[34] is a general approach to developing high-order numerical schemes, but for this work we are only interested in methods with stencils that are two grid points wide in all space dimensions because they enable arbitrarily high-order spatial interpolation with no decrease in CFL stability bounds near a boundary.

These two-point stencil schemes are numerically stable only when a staggered grid is used at each time step[34] as shown in Fig. 2 for a two-dimensional stencil. The locations marked with Xs in Fig. 2 are advanced first using the 2 × 2 stencils centered about each X. Next, the Hermitian data now stored at the X locations are used as initial data to advance the solution to the center as indicated by the large dot in Fig. 2. Staggering the grid has the same effect as applying the MESA scheme to the entire 3 × 3 stencil in the Fig. 2 and then adding artificial dissipation. It is, however, more efficient to stagger the grid because neighboring grid points will reuse the data at the X locations in Fig. 2. In two dimensions, this staggered grid procedure is numerically stable if[34]

$$\lambda = \frac{\Delta t}{\Delta x} \leq \frac{1}{1 + \max\{|M_x|, |M_y|\}} \tag{2}$$

A. Hermitian Derivative Approximation

The MESA scheme requires an approximation to the solution of the primitive variables p and u_i and their spatial derivatives at the center of each stencil. Let the function $f(x, y)$ be an approximation of one of the primitive variables from Eqs. (1) and define its origin to be at the center of each two-point stencil shown in Fig. 3. Then in two dimensions, the following data must be approximated once for each primitive variable $[f(x, y) = p(x, y), u(x, y), \text{and } v(x, y)]$:

$$\frac{\partial^{a+b} f(0,0)}{\partial x^a \partial y^b} \qquad \forall \quad a, b : a, b = 0, 1, 2, \ldots, 2s + 1 \tag{3}$$

On a Cartesian grid, these approximations can be found using tensor product interpolation[28] in two or three dimensions if the interpolant

$$f(x, y) = \sum_{i=0}^{2s+1} \sum_{j=0}^{2s+1} C_{i,j}^f x^i y^j \tag{4}$$

with its origin at stencil center is used with a $2s + 1$-order method and if the following data is available at each grid point:

$$\frac{\partial^{a+b} f(x, y)}{\partial x^a \partial y^b}, \qquad \forall \quad a, b : a, b = 0, 1, 2, \ldots, s \tag{5}$$

Determining the $C_{i,j}^f = (1/i!j!)[\partial^{i+j} f(x, y)/\partial x^i y^j]$ terms in the two-dimensional interpolant [Eq. (4)] for very high-order methods is very difficult and inefficient unless it is reduced to a series of one-dimensional interpolations by using tensor products. Briefly, this is accomplished by performing one-dimensional interpolations of the form[27]

$$f(x) = \sum_{i=0}^{2s+1} C_i^f x^i \tag{6}$$

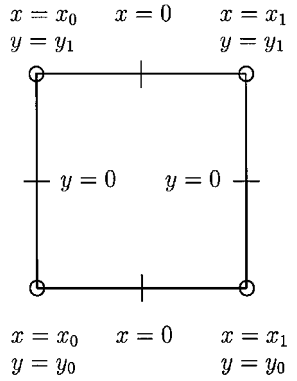


Fig. 3 Two-point two-dimensional stencil.

It is then possible to interpolate in the x direction, using the data on the grid points, the following two sets of data for $y = y_0$ and y_1:

$$\left(\frac{1}{i!}\right) \frac{\partial^{i+j} f(x = 0, y)}{\partial x^i \partial y^j}, \qquad \forall i, j : \begin{array}{l} i = 0, 1, 2, \ldots, 2s + 1 \\ j = 0, 1, \ldots, s \end{array} \tag{7}$$

Then these data are used to interpolate in the y direction using the following one-dimensional shape function:

$$f(x = 0, y) = \sum_{j=0}^{2s+1} C_j^f y^j \tag{8}$$

Once the C_j^f terms are determined, they provide the data shown in Eq. (3):

$$C_j^f = \left(\frac{1}{i!j!}\right) \frac{\partial^{i+j} f(x = 0, y = 0)}{\partial x^i \partial y^j} \tag{9}$$

The one-dimensional shape functions [Eqs. (6) and (8)] can be directly solved using computer algebra (Gröebner bases) (see Ref. 18) to create algebraic expressions for each C^f term for up to approximately 30th-order methods. However, this results in lengthy, inefficient expressions that limit the accuracy and prevent instantaneous accuracy changes necessary for resolving varying wavelength scales. A better way is to use Hermitian divided-differences, which will create the equivalent one-dimensional shape function (Newton's interpolatory formula),[35]

$$f(x) = \sum_{i=s+1}^{2(s+1)-1} Q_{i,i} (x - x_0)^{s+1} (x - x_1)^{i-(s+1)} + \sum_{i=0}^{s} Q_{i,i} (x - x_0)^i \tag{10}$$

where the $Q_{i,i}$ are the traditional Hermitian divided differences from the tableau (shown in Fig. 4) which are given by

```
DO $i = 0, s$
    DO $j = 0, i$
        $Q_{i,j} = \left(\frac{1}{j!}\right) \frac{\partial^j f(x_0)}{\partial x^j}$
        $Q_{i+s+1,j} = \left(\frac{1}{j!}\right) \frac{\partial^j f(x_1)}{\partial x^j}$
    END DO
END DO
DO $i = s + 1, 2 * s + 1$
    DO $j = i - s, i$
        $Q_{i,j} = \frac{Q_{i,j-1} - Q_{i-1,j-1}}{\Delta x}$
    END DO
END DO
```

$$\begin{array}{cccccc}
Q_{0,0} & & & & & \\
 & Q_{1,1} & & & & \\
Q_{1,0} & & Q_{2,2} & & & \\
 & Q_{2,1} & & Q_{3,3} & & \\
Q_{2,0} & & Q_{3,2} & & Q_{4,4} & \\
 & Q_{3,1} & & Q_{4,3} & & Q_{5,5} \\
Q_{3,0} & & Q_{4,2} & & Q_{5,4} & \\
 & Q_{4,1} & & Q_{5,3} & & \\
Q_{4,0} & & Q_{5,2} & & & \\
 & Q_{5,1} & & & & \\
Q_{5,0} & & & & &
\end{array}$$

Fig. 4 Fifth-order divided-difference tableau.

It is not enough to simply find Newton's interpolatory formula [Eq. (10)] because we need to approximate p, u, and v and their spatial derivatives at the stencil center. This would ordinarily require differentiating Newton's Eq. (10) using computer algebra. By the product rule of differentiation, Newton's formula will double in size after each differentiation, and a 50th-order MESA scheme would need a Newton's formula of approximately $2^{50} \approx 10^{16}$ terms. This issue is eliminated by the following result, which is applicable only to two-point stencils[27]:

$$\frac{\partial^{\mathrm{d}x} f(x=0)}{\partial x^{\mathrm{d}x}} = \sum_{i=\mathrm{d}x}^{2s+1} Q_{i,i}\, Z_{i,\mathrm{d}x} \tag{11}$$

where the function $Z_{i,\mathrm{d}x}$ is independent of space and time and can be computed simply as follows:

```
DO $\mathrm{d}x = 0, 2s+1$
  DO $i = \mathrm{d}x, s$
    $Z_{i,\mathrm{d}x} = \dfrac{i!}{(i-\mathrm{d}x)!}(-x_0)^{(i-\mathrm{d}x)}$
  END DO
END DO                                                    (12)
```

```
DO $\mathrm{d}x = 0, 2s+1$
  DO $i = s+1, 2s+1$
    $Z_{i,\mathrm{d}x} = \sum_{r=0}^{\mathrm{d}x}\left(\dfrac{\mathrm{d}x!}{(\mathrm{d}x-r)!r!}(-x_0)^{(s+1-r)}(-x_1)^{(i-s-1-\mathrm{d}x+r)}\right.$
        $\left. * \prod_{e=0}^{\mathrm{d}x-1-r}[i-(s+1)-e] * \prod_{k=0}^{r-1}[s+1-k]\right)$
  END DO
END DO                                                    (13)
```

With these developments it is now possible to approximate efficiently all solution variables and their derivatives at the center of a two-point stencil to any level of accuracy with only a page of code. Because these explicit forms are completely algebraic, it is possible to adapt dynamically the accuracy of the approximation to the solution evolution.

B. Time Evolution

To utilize fully the arbitrarily high-order accuracy in space that is possible on two-point stencils, it is necessary to achieve similar accuracy in time. This can be accomplished with the MESA method. This method uses the governing equations [Eqs. (1)] to convert the recently approximated spatial derivatives to mixed space–time derivatives in a manner reminiscent of the Lax–Wendroff approach[36] by differentiating the following equations in space and time:

$$\frac{\partial u_j}{\partial t} = -\left(M_i \frac{\partial u_j}{\partial x_i} + \frac{\partial p}{\partial x_j}\right), \qquad \frac{\partial p}{\partial t} = -\left(M_i \frac{\partial p}{\partial x_i} + \frac{\partial u_j}{\partial x_j}\right) \tag{14}$$

The mixed space–time derivatives are then used in modified series expansions with local coordinates about the center of the interpolation stencil at the current time level. For two-dimensional problems, the following series are used, with $O = 2s + 1$:

$$p(x,y,t) = \sum_{i=0}^{O}\sum_{j=0}^{O}\sum_{k=0}^{2(O)} C^{p}_{i,j,k} x^i y^j t^k$$

$$u(x,y,t) = \sum_{i=0}^{O}\sum_{j=0}^{O}\sum_{k=0}^{2(O)} C^{u}_{i,j,k} x^i y^j t^k$$

$$v(x,y,t) = \sum_{i=0}^{O}\sum_{j=0}^{O}\sum_{k=0}^{2(O)} C^{v}_{i,j,k} x^i y^j t^k \tag{15}$$

where

$$C^{f}_{i,j,k} = \left(\frac{1}{i!j!k!}\right)\frac{\partial^{i+j+k} f(x=0, y=0, t=0)}{\partial x^i \partial y^j \partial t^k}$$

The variables p, u, and v and their spatial derivatives are then advanced to the next time step by evaluating Eq. (15) and its spatial derivatives with $x = 0$, $y = 0$, and $t = \Delta t$ as follows:

$$\frac{\partial^{a+b} p(0,0,\Delta t)}{\partial x^a y^b} = \sum_{k=0}^{2(O)} (a!b!) C^{p}_{a,b,k} \Delta t^k$$

$$\frac{\partial^{a+b} u(0,0,\Delta t)}{\partial x^a y^b} = \sum_{k=0}^{2(O)} (a!b!) C^{u}_{a,b,k} \Delta t^k$$

$$\frac{\partial^{a+b} v(0,0,\Delta t)}{\partial x^a y^b} = \sum_{k=0}^{2(O)} (a!b!) C^{v}_{a,b,k} \Delta t^k \tag{16}$$

By unrolling the Cauchy–Kowalewski recursion in Eq. (14), we can quickly express the mixed space–time derivatives $C^{p}_{a,b,k}$, $C^{u}_{a,b,k}$, and $C^{v}_{a,b,k}$ in Eq. (16) in terms of the space derivatives $C^{p}_{a,b,0}$, $C^{u}_{a,b,0}$, and $C^{v}_{a,b,0}$, which were approximated at the stencil center in the last section. The following loop efficiently unrolls the recursion[29]:

```
DO $k = 1, 2O$
  DO $b = 0, O$
    DO $a = 0, O$
      $\hat{a} = a+1, \qquad \hat{b} = b+1, \qquad \hat{k} = k-1$
      $C^{p}_{a,b,k} = \Big(-\big((\hat{a}) * \big(M_x * C^{p}_{\hat{a},b,\hat{k}} + C^{u}_{\hat{a},b,\hat{k}}\big)\big)$
          $- (\hat{b}) * \big(M_y * C^{p}_{a,\hat{b},\hat{k}} + C^{v}_{a,\hat{b},\hat{k}}\big)\Big)\Big/k$
      $C^{u}_{a,b,k} = \Big(-\big((\hat{b}) * M_y * C^{u}_{a,\hat{b},\hat{k}}\big) - (\hat{a}) * \big(C^{p}_{\hat{a},b,\hat{k}} + M_x * C^{u}_{\hat{a},b,\hat{k}}\big)\Big)\Big/k$
      $C^{v}_{a,b,k} = \Big(-\big((\hat{b}) * \big(C^{p}_{a,\hat{b},\hat{k}} + M_y * C^{v}_{a,\hat{b},\hat{k}}\big)\big) - (\hat{a}) * M_x * C^{v}_{\hat{a},b,\hat{k}}\Big)\Big/k$
    END DO
  END DO
END DO                                                    (17)
```

This simple form applies to uniform mean flow, but for general flows Cauchy–Kowalewski recursion becomes complicated.[26] For example, using Eq. (14) to calculate the mixed space–time derivative $C^u_{1,0,1}$, when the mean flow varies in space, results in the expression

$$\frac{\partial^2 u}{\partial t \partial x} = -\left(\frac{\partial M_i}{\partial x} \frac{\partial u}{\partial x_i} + M_i \frac{\partial^2 u}{\partial x_i^2} + \frac{\partial^2 p}{\partial x^2} \right) \tag{18}$$

which requires knowing the derivatives of the mean flow, $\partial M_i / \partial x$, and the higher-order derivatives of Eq. (18) require higher-order derivatives of the mean flow as well. Getting this information may require computing the mean flow using a very high-order MESA method as well, and the equations will grow exponentially unless a simple form analogous to Eq. (11) is found. Despite these complications, Cauchy–Kowalewski recursion, when automated with computer algebra tools,[29] can produce numerical schemes with higher order than is currently possible with Runge–Kutta methods.

V. Including Wall Boundaries

The procedure for advancing the interior grid points is not applicable to fill grid points near boundaries because part of their stencil is on or within a solid object. We obtain data for these points at each time step with a Hermite interpolant that is consistent with the wall boundary conditions and that uses the nearby solution at interior grid points. This Hermite interpolant is equivalent to the interpolant used for the interior grid points in Eq. (4) but is written as

$$f(x, y) = \sum_{i,j=0}^{1} \sum_{\mathrm{d}x,\mathrm{d}y=0}^{s} (H_{x_i,\mathrm{d}x}(x))(H_{y_j,\mathrm{d}y}(y)) \frac{\partial^{\mathrm{d}x+\mathrm{d}y} f(x_i, y_j)}{\partial x^{\mathrm{d}x} \partial y^{\mathrm{d}y}} \tag{19}$$

where the $H_{x_i,\mathrm{d}x}(x)$ and $H_{y_j,\mathrm{d}y}(y)$ terms are $2s+1$-order polynomials in x and y, respectively. Because the data at the interior grid points are known, this Hermite form of the interpolant has the advantage of reducing the number of unknown coefficients, $C^f_{i,j}$, in Eq. (4), to only the unknown data which are required at the fill points, $\partial^{\mathrm{d}x+\mathrm{d}y} f(x_i, y_j)/\partial x^{\mathrm{d}x} \partial y^{\mathrm{d}y}$.

Each shaded box in Fig. 5 represents an interpolation region used for interpolating the values at the fill points in each box. The boundary conditions are imposed upon the shape function for each box at the locations on the surface intersected by the arrows. The numbers inside each box show the sequence in which the fill points are interpolated.

A. Choice of Wall Boundary Conditions

Each fill point on the grid provides $(s+1)^2$ unknowns in Eq. (19), so that an equivalent number of constraint equations must be obtained. An infinite number of constraints can be obtained by exploiting that the physical boundary conditions apply uniformly in time everywhere in the boundary.[4] For Eqs. (1), the inviscid boundary condition is that the normal component of velocity is zero at a surface:

$$V \cdot \hat{\eta} = 0 \tag{20}$$

Because vorticity is convected with the mean flow, we also assume in this work that there is no vorticity at a surface:

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0 \tag{21}$$

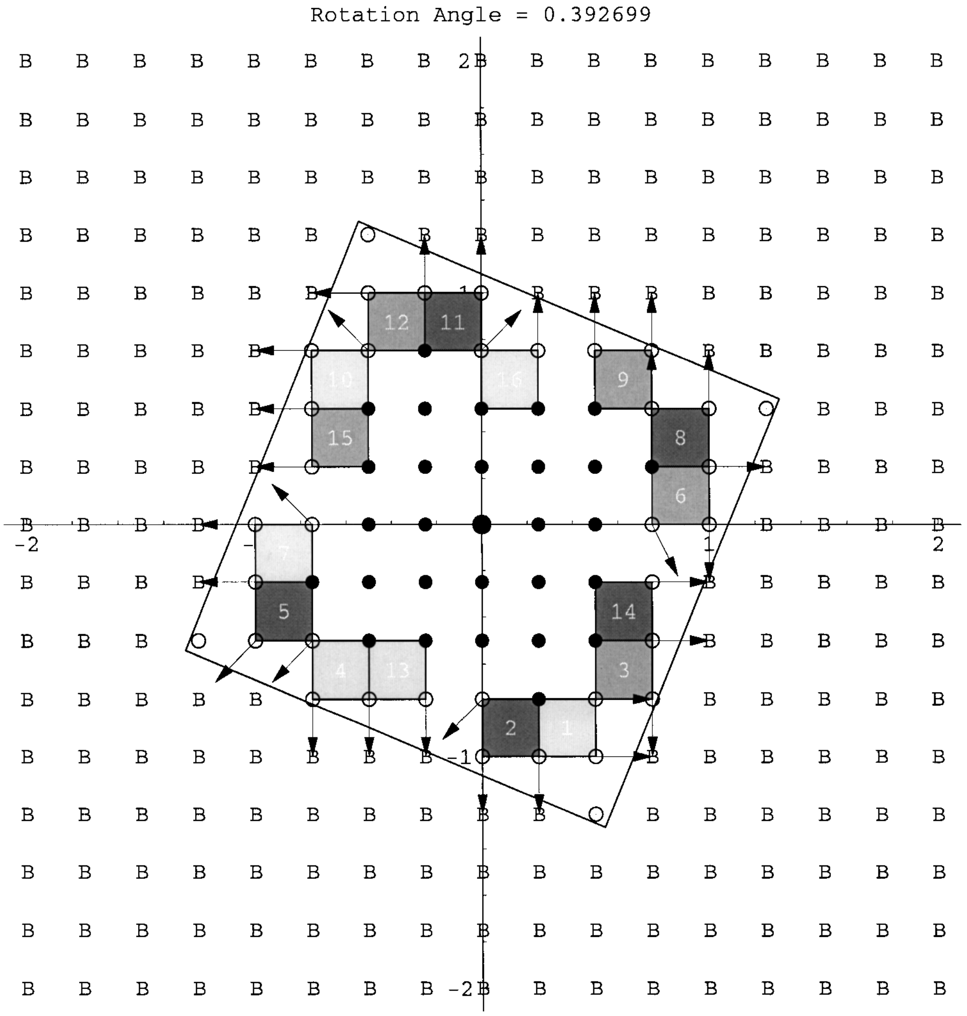


Fig. 5 Interpolating fill point data.

For flat walls, the boundary conditions for this case are

$$\frac{\partial^{2n+1+T} p}{\partial \hat{\eta}^{2n+1} \partial \hat{\tau}^{T}} = 0$$

$$\frac{\partial^{2n+T} V_{\hat{\eta}}}{\partial \hat{\eta}^{2n} \partial \hat{\tau}^{T}} = 0, \qquad \forall n, T : n, T \in (0, 1, 2, \ldots)$$

$$\frac{\partial^{2n+1+T} V_{\hat{\tau}}}{\partial \hat{\eta}^{2n+1} \partial \hat{\tau}^{T}} = 0 \tag{22}$$

where V is the magnitude of the perturbation velocity vector with velocity components u and v in the Cartesian x- and y-axis directions, p is the scalar pressure, $\hat{\eta}$ is the unit vector normal to the wall surface direction, $\hat{\tau}$ is the unit vector tangent to the wall boundary direction, $V_{\hat{\tau}}$ is the velocity tangential to the wall, and $V_{\hat{\eta}}$ is the velocity normal to the wall.

Selecting which set of boundary conditions to use and where to apply them on the boundary can be simplified by considering the general form of the normal and tangential derivatives of $f(x, y)$ (Ref. 37):

$$\frac{\partial^{N+T} f(x, y)}{\partial \hat{\eta}^{N} \partial \hat{\tau}^{T}} = \left(\eta_x \frac{\partial}{\partial x} + \eta_y \frac{\partial}{\partial y}\right)^{N} \left(-\eta_y \frac{\partial}{\partial x} + \eta_x \frac{\partial}{\partial y}\right)^{T} f(x, y) \tag{23}$$

If each operator is expressed in its binomial series form and the order of summation rearranged, then by substituting Eq. (19) into Eq. (23) the following form is derived:

$$\frac{\partial^{N+T} f(x, y)}{\partial \hat{\eta}^{N} \hat{\tau}^{T}} = \sum_{i,j=0}^{1} \sum_{\mathrm{d}x,\mathrm{d}y=0}^{s} [Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j}] \frac{\partial^{\mathrm{d}x+\mathrm{d}y} f(x_i, y_j)}{\partial x^{\mathrm{d}x} \partial y^{\mathrm{d}y}} \tag{24}$$

where $Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j}$ is defined by

$$Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j} = \sum_{a=0}^{T} \sum_{b=0}^{N} K(a, b) \frac{\partial^{N+T} (H_{x_i,\mathrm{d}x}(x) H_{y_j,\mathrm{d}y}(y))}{\partial x^{N+T-a-b} \partial y^{a+b}} \tag{25}$$

$$K(a, b) = \left(\frac{N!T!}{(N-b)!b!(T-a)!a!}\right) \eta_x^{N+a-b} \eta_y^{T-a+b} (-1)^{T-a} \tag{26}$$

Because maintaining numerical stability for hyperbolic problems normally requires methods that include all information from within the domain of dependence defined by the characteristic surfaces,[38] we want to select $N = 2n + 1$ and T so that

$$\frac{\partial^{N+T} \left(H_{x_i,\mathrm{d}x} H_{y_j,\mathrm{d}y}\right)}{\partial x^{N+T-a-b} \partial y^{a+b}}$$

in Eq. (25) never becomes zero because this would exclude some of the influence of neighboring grid points. For example, if $Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j} = 0$, then the grid point data $\partial^{\mathrm{d}x+\mathrm{d}y} f(x_i, y_j)/\partial x^{\mathrm{d}x} \partial y^{\mathrm{d}y}$ would be excluded from the boundary condition in Eq. (24). Similarly, if a part of the sum forming $Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j}$ is missing, then some influence of the corresponding grid point is absent as well. Maintaining the full influence is accomplished by choosing $(n : n = 0, 1, 2, \ldots, s)$ and $[T : T = 0, 1, 2, \ldots, 2(s - n)]$ for the first and third boundary conditions in Eq. (22). Those conditions will produce linearly consistent systems that can be solved to determine the interpolant or, equivalently, the data required at the fill point. However, in the special case in which $\eta_x = 0$ or $\eta_y = 0$, we need to use the different conditions $(n : n = 0, 1, 2, \ldots, s)$ and $(T : T = 0, 1, 2, \ldots, s)$ to ensure nonsingular matrix systems. This special case does not result in a loss of information because the affected terms in Eq. (25) are supposed to be zero. For example, a third-order MESA method would use the following pressure boundary conditions at each location indicated by the arrows in Fig. 5:

$$\frac{\partial p}{\partial \hat{\eta}} = 0, \qquad \frac{\partial^2 p}{\partial \hat{\eta} \partial \hat{\tau}} = 0, \qquad \frac{\partial p}{\partial \hat{\eta} \partial \hat{\tau}^2} = 0, \qquad \frac{\partial^3 p}{\partial \hat{\eta}^3} = 0 \tag{27}$$

The same method applied to the special case shown in Fig. 6 would use the conditions

$$\frac{\partial p}{\partial \hat{\eta}} = 0, \qquad \frac{\partial^2 p}{\partial \hat{\eta} \partial \hat{\tau}} = 0, \qquad \frac{\partial^3 p}{\partial \hat{\eta}^3} = 0, \qquad \frac{\partial^4 p}{\partial \hat{\eta}^3 \hat{\tau}} = 0 \tag{28}$$

B. Symbolic Solution

The shape function for each fill point can now be solved using the boundary conditions and the known interior data in each shaded region of Fig. 5. The boundary condition Eqs. (22) are applied to the shape function Eq. (24) to solve for the unknown data at the fill points. This linear system of $(s + 1)^2$ equations per fill point may be written as

$$\boldsymbol{Mf} = \boldsymbol{Nd} \tag{29}$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ are matrices whose elements are given by $Y_{\mathrm{d}x,\mathrm{d}y,x_i,y_j}$ in Eq. (25), where $\boldsymbol{f}$ is a vector of the data needed for the fill points and where $\boldsymbol{d}$ is a vector of known data from the interior points.

This system can be solved using any linear solver at each time step, but for high-order methods this becomes expensive and the matrices become ill conditioned. A better approach is to solve symbolically this system once to form the algebraic solution

$$\boldsymbol{f} = \boldsymbol{M}^{-1}\boldsymbol{Nd} \tag{30}$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ are numerical matrices dependent on the geometry but the vectors are symbolic. This results in an algebraic expression for each fill point as a linear combination of the interior grid point data. It is then a simple matter to evaluate the fill points by updating the vector $\boldsymbol{d}$ at each time step and evaluating their linear combinations.

This approach worked well for up to 11th-order accuracy in two dimensions for the case (shown in Fig. 6) of no mean convection in an unrotated box that is aligned with the coordinate axes. For higher-order algorithms, the matrix $\boldsymbol{M}$ becomes ill conditioned, and finding its numerical inverse becomes difficult, inefficient, and unstable. Resorting to Gaussian elimination would be more stable for high-order cases, but the cost at each time step is prohibitive.

VI. Stencil Constraint Tree

Instead of using a poorly conditioned numeric matrix $\boldsymbol{M}$, its symbolic form can be inverted. There are actually only a relatively few symbolic matrices that need to be inverted regardless of the geometry. These few cases can be found by using a new tree data structure, referred to as a stencil constraint tree (SCT), that represents all possible stencil configurations. For example, in Fig. 7, all possible 3 × 3 stencil configurations are shown in which 1) the top, center grid point is a boundary point (indicated by B); 2) the center, center grid point is a fill point (open circles); and 3) the bottom, center grid point is an interior point (filled-in circles). The SCT for the simple case of a 2 × 2 stencil that has an interior grid point in its top-right location is shown in Fig. 8. It is constructed by propagating symbolic constraints in a manner analogous to Waltz's symbolic constraint propagation algorithm (see Ref. 39).

Once the tree is constructed, a particular stencil configuration is found by traversing the SCT from top to bottom along any path and converting the branch labels using Table 1 and Fig. 9. For example, the branch number 10 corresponds to position 4 and label 1 in Table 1 and represents an interior grid point that is located at the top-right corner of the stencil. A similar procedure is applicable to larger stencils as well. Notice that identifying a particular n-point stencil using an SCT can be done in n^2 or n^3 steps for two- and three-dimensional stencils, respectively, instead of the $3^{n^2}n^2$ and $3^{n^3}n^3$ comparisons typically required using a brute-force comparison of all stencil configurations.

Table 1 Branch number legend

Position	Label	Branch
1	1–3	1–3
2	1–3	4–6
3	1–3	7–9
4	1–3	10–12

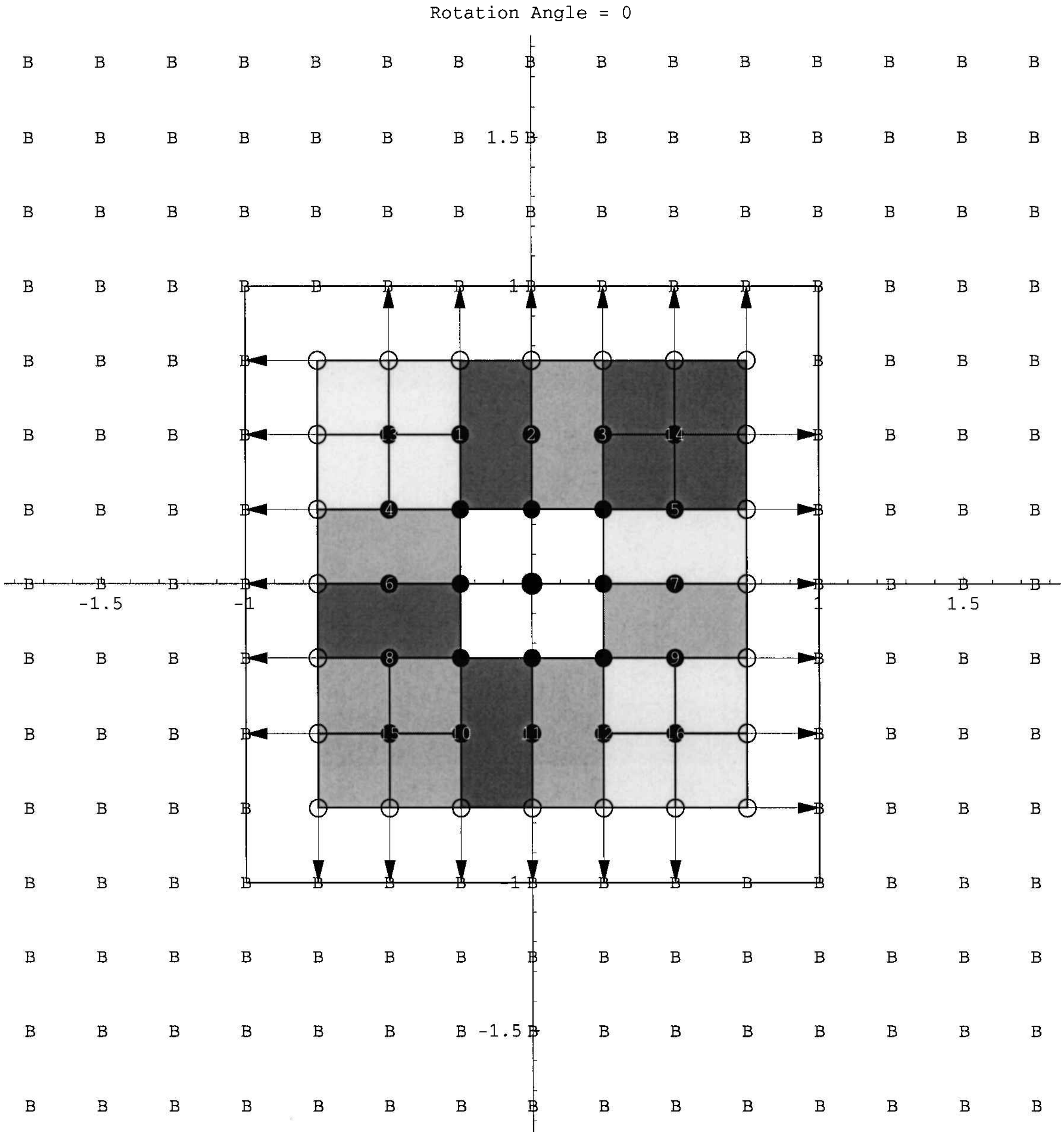


Fig. 6 Stable unrotated box case.

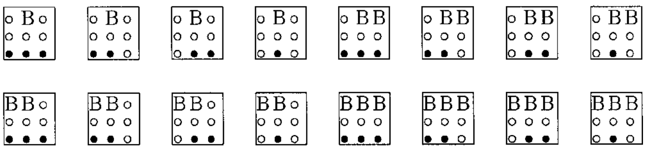


Fig. 7 All possible S7 stencil configurations with fill at center: 16 cases.

A. Constructing the Tree

The most significant advantage of the SCT is it can reduce the number of symbolic matrices from Eq. (29) that need to be considered. This is accomplished by propagating natural constraints during the construction of the tree.[29] Natural constraints are simply a list of rules that are a natural consequence of the definitions of interior, fill, and boundary grid points on a Cartesian grid. For example, some natural constraints that limit the set of stencil configurations are 1) an interior grid point is never adjacent to a boundary grid point and its converse and 2) for two- and three-point stencils, a fill grid point will always be adjacent to at least one boundary point. If the natural constraints are not enforced, then the list of stencil configurations in Fig. 7 grows to $3^6 = 729$ because only the middle column is defined. The advantages of enforcing natural constraints increase with both spatial dimension and stencil size.

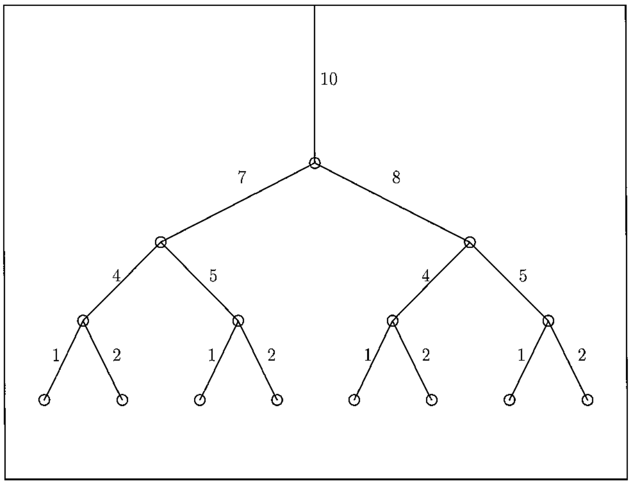


Fig. 8 Two-point stencil constraint tree.

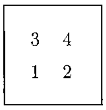


Fig. 9 Stencil grid positions.

●= Interior Grid Point Type
○= Fill Grid Point Type
Z= Not a Boundary Grid Point Type
X= Unspecified Grid Point Type
B= Boundary Grid Point Type

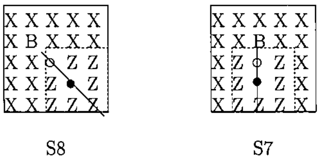


Fig. 10 Sub 3 × 3 stencil symmetry.

B. Small Stencils Guarantee CFL Stability

These techniques can be applied to both of the 5 × 5 stencils shown in Fig. 10 to construct their SCT. This SCT will represent all possible 5 × 5 stencils that can occur with a fill point at its center. We find that there are only 65 unique rotationally symmetric cases of 3 × 3 stencils centered about the fill point.[29] All other possible stencil configurations are simply rotations of this core group. Note that we assume there will be at least one interior point next to the fill point, otherwise that fill point is not needed as shown in the corners of Fig. 5. As shown in Fig. 10, there will always be a 3 × 3 stencil contained within the 5 × 5 stencil that does not contain a boundary point and yet always contains a fill point on its edge or corner. This guarantees the existence of a 3 × 3 stencil for each fill point in any geometry that is not intersected by the surface geometry. Therefore, it will always be possible to expand the stencil outward from each fill point to the surface where the boundary conditions are imposed. This in turn guarantees that the CFL stability criterion will be satisfied because the domain of dependence for the fill interpolant is increased from a standard stencil size.

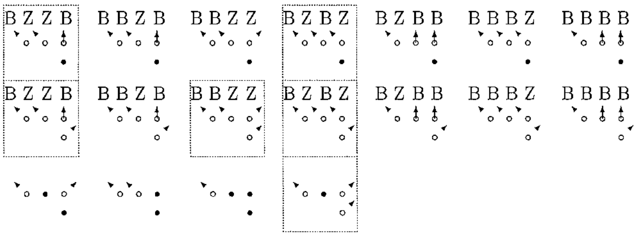
This CFL guarantee is a property of two-point and three-point stencils in two or three dimensions. Four-point stencils or larger will require smaller time steps near the boundary and make it difficult to choose locations on the surface that guarantee a linearly consistent system in Eq. (29).

VII. Mapping

All possible 5 × 5 stencil configurations containing a fill point are known after the construction of the SCT for the stencils in Fig. 10. This information is used to select locations on the boundary at which the boundary conditions are enforced. A surface will always occur between a fill point and a boundary point, so that even before we know exactly where the surface is, we know in which direction to go to find the surface.

We would like to use the points on the surface that are closest to the interpolation stencil while ensuring that never more than N locations are collinear with the coordinate axes for each N-point interpolation stencil. These conditions maximize the accuracy of the interpolant and ensure that the matrix $\boldsymbol{M}$ in Eq. (29) is nonsingular. One such mapping is shown in Fig. 11 for the upper triangular part of the outlined S8 3 × 3 stencil in Fig. 10. The degenerate cases occur because it is not guaranteed that a nearby surface will be found using the mapping shown. These cases are usually not an issue in practice and can be avoided by choosing alternate interpolation regions as in Fig. 12 or by constructing the SCT for a larger stencil around the fill point in Fig. 10 and then choose a mapping with this additional information.

Implementing higher accuracy boundary conditions with this approach requires increasing the depth of Hermitian data at each grid point and increasing the number of boundary conditions used from Eq. (22), but the stencil size used for interpolation is never changed. Therefore, we never need to remap the fill points to the boundary and the CFL stability is not affected.



□ = Possibly Degenerate

Fig. 11 S8 symmetrical mapping: 18 cases.

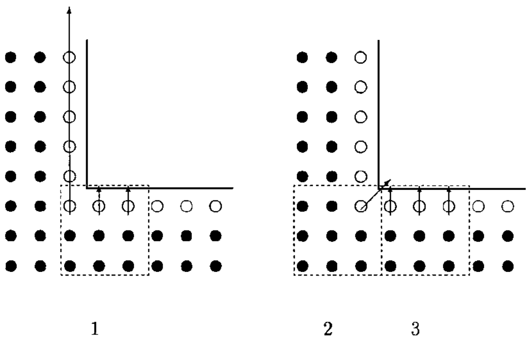


Fig. 12 Use regions 2 and 3 instead of region 1.

A. Efficient Retrieval of Fill Point Solutions

Once all of the stencil configurations are known and the direction in which to map each fill point to the surface is found, then the symbolic form is known for every matrix $\boldsymbol{M}$ and $\boldsymbol{N}$ in Eq. (29). Each of these systems may then be symbolically solved, and their subsequent solutions may be stored in the leaves of the SCT for fast retrieval later.

The leaves of the tree will contain algebraic solutions for every fill point in any geometrical configuration. They can be used and reused many times without the need for solving Eq. (29) again. For example, if the walls are moving then the Cartesian grid points will need to be relabeled, but the same SCT, mapping, and symbolic solutions can be used at each time step to interpolate efficiently the data needed at each fill point.

VIII. Numerical Results

All of the following results solve the linearized Euler equations [Eqs. (1)] in uniform mean flow using the approaches already described. This approach to developing very-high order methods was applied to both wave propagation and acoustic scattering problems in various test geometries.

A. Acoustic Propagation

For mean convection velocity $\boldsymbol{M}$ in an open biperiodic (triperiodic) domain (for $d = 2$ or 3 dimensions), the linearized Euler equations have the following analytical solution:

$$p(\boldsymbol{x}, t) = C_t \prod_{i=1}^{d} S_{x_i}, \quad u_i(\boldsymbol{x}, t) = -S_{t,i} C_{x_i} \prod_{\substack{j \neq i \\ j=1}}^{d} S_{x_j} \tag{31}$$

where

$$C_{x_i} = \cos(W_i \pi \hat{x}_i), \qquad S_{x_i} = \sin(W_i \pi \hat{x}_i)$$

$$C_t = \cos(\pi t \|\boldsymbol{W}\|), \qquad S_{t,i} = \frac{W_i \sin(\pi t \|\boldsymbol{W}\|)}{\|\boldsymbol{W}\|}$$

$$\|\boldsymbol{W}\| = \sqrt{\sum_{a=1}^{d} W_a^2}, \qquad \hat{x}_i = (x_i - M_i t)$$

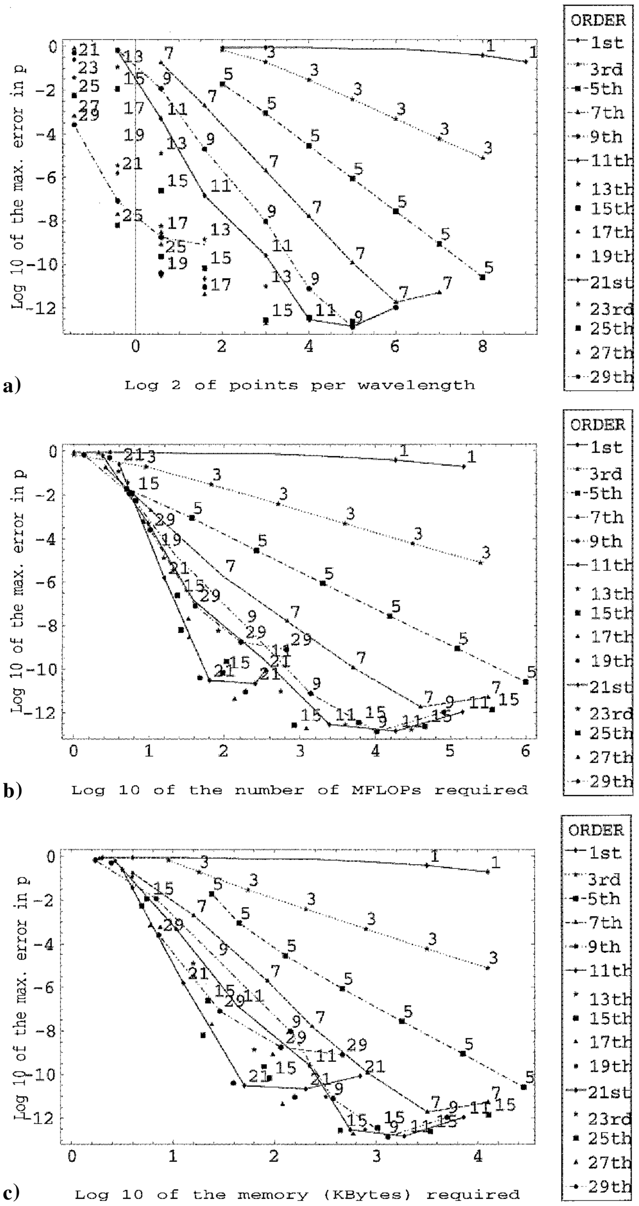


Fig. 13 Resolution, efficiency, and memory usage of higher-order methods.

The properties of very high-order methods were tested on this problem with mean velocity $\boldsymbol{M} = (1, 1)$ and wave number $\boldsymbol{W} = (1, 1)$, and the results were used to validate the automatically generated codes. The methods were compared for efficiency and resolution; the results for up to 29th-order accuracy (using 64-bit precision) are shown in Fig. 13. Notice that higher-order methods perform better in each part of Fig. 13 until machine precision is reached. By increasing the wave number in Eq. (31) to $\boldsymbol{W} = (W_1 \geq 2, W_2 \geq 2)$, we also see that very high-order methods offer subgrid scale resolution as shown to the left of the ordinate axis at the top of Fig. 13. Similar results were found while solving three-dimensional cases as well.[29]

The resolution of the methods improves while using 128-bit precision as the accuracy increases until about the 21st order for signals with eight grid points per wavelength: This limit varies with signal frequency. If the order is increased above this limit, then the resolution of each signal degrades from roundoff errors that begin to dominate around 57th-order accuracy for signals with a wavelength of eight grid points. However, by selecting wave number $\boldsymbol{W} = (16, 16)$ in Eq. (31), and by using 40th–60th order methods on a two-point stencil, it is possible to resolve 4 wavelengths per grid point. Note that this is not a violation of the Nyquist criterion[40] because each grid point contains multiple data values for Hermite interpolation.

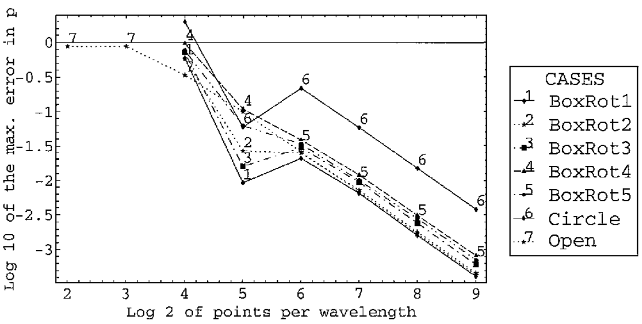


Fig. 14 Resolution of rotated boxes and circles.

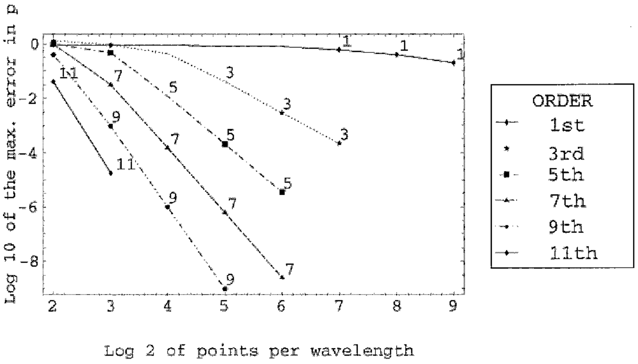


Fig. 15 Unrotated box grid resolution studies.

For extremely high-order methods, the convergence slope at the top of Fig. 13 is so steep that essentially only a specific frequency is correctly resolved with finite precision floating point hardware unless a method for detecting roundoff error is available. Fortunately, roundoff error may be detected by checking the columns of the Hermite divided-difference tableau in Fig. 4, which is used for the interpolation. The sum of column j should equal the difference of the first and last terms in column $j - 1$ (Refs. 27 and 41). If roundoff error begins to occur, simply stop constructing the tableau at that point and use the resulting lower-order method. This saves computational effort and enables the solution of a field containing widely varying wavelengths.

B. Acoustic Scattering

With the exceptional qualities of very high-order methods established for wave propagation, the simulation of acoustic scattering from within a rotated box and a circle was completed (Fig. 14). An analytical solution to wave scattering within the rotated box with no mean convection is

$$p(x, y, t) = -\cos(\sqrt{2}\pi t)\cos(\pi x)\cos(\pi y) \tag{32}$$

$$u(x, y, t) = -[\cos(\pi y)\sin(\sqrt{2}\pi t)\sin(\pi x)/\sqrt{2}] \tag{33}$$

$$v(x, y, t) = -[\cos(\pi x)\sin(\sqrt{2}\pi t)\sin(\pi y)/\sqrt{2}] \tag{34}$$

where the coordinate system is aligned with the box. An analytical solution to wave scattering within a circle with no mean convection is

$$p(r, \theta, t) = \frac{5J_0(\lambda r)\cos(\lambda t)}{\sqrt{\pi}J_0(\lambda)} \tag{35}$$

where $\lambda = 3.83171$, J_0 is the Bessel function of the first kind of order 0, and polar coordinates are used.

For the case of an unrotated box, we found stable solutions with 2nd-, 3rd-, 5th-, 7th-, 9th-, and 11th-order accuracy using boundary conditions from Eq. (22) with $(n : n = 0, 1, 2, \ldots, s)$ and $(T : T = 0, 1, 2, \ldots, s)$ as shown in Fig. 15. Using the boundary conditions in Eq. (22) with $n, T = 0$, we found stable second-order solutions for all box and circle cases, regardless of the rotation angle. The first five plots in Fig. 14 correspond to box rotation angles

0, $\pi/16$, $\pi/8$, $\pi/4$, and $\pi/3$, respectively. The last curve in Fig. 14 shows a second-order method without wall boundaries.

C. Numerical Stability

For second-order boundary treatments, ordering interpolation regions by the number of fill points they contain from fewest to most and then interpolating the fill points in that order resulted in stable solutions for irregular geometries. There are, however, multiple stable choices of shaded regions and interpolation sequences, and rather than relying on numerical experiments, an important next step is to use the symbolic form in Eq. (29), of which only a relatively few cases are possible, to perform a complete stability analysis. With this better understanding, perhaps higher-order conditions can be made reliably stable in irregular geometries by explicitly adding artificial dissipation or by rearranging the interpolation regions.

IX. Conclusions

Small stencils containing Hermitian data were found to possess key properties necessary for developing exceptionally efficient, explicit algorithms of arbitrary formal accuracy in space and time for scattering problems with arbitrary surface orientations. This is accomplished by approximating spatial derivatives using a special form of two-point Hermitian divided-difference spatial interpolation and an unrolled Cauchy–Kowalewski recursion procedure for time advancement. A stencil constraint tree is used to find all possible stencil configurations and their respectively well-posed boundary conditions so that grid points near arbitrarily oriented surfaces may be interpolated. Computer algebra is then used to find the symbolic shape function for each stencil configuration, and the stencil constraint tree is used to identify quickly the correct interpolant for each near boundary grid point.

In spite of the development of simple procedures for the time evolution at all grid points, the FORTRAN application code of a very high-order method is complicated, and so the task of programming was automated using computer algebra procedures. These procedures have successfully produced 1) stable, parallel 57th-order wave propagation methods in two and three spatial dimensions with periodic boundary conditions, 2) stable 2nd-order methods for scattering problems with generalized surface boundary conditions in two spatial dimensions, and 3) stable 11th-order methods for scattering problems with surfaces aligned to the grid in two spatial dimensions.

The resolution and efficiency of these methods improve with order until the accuracy exceeds the limits of machine precision. At these very high orders of design accuracy, it is necessary to control roundoff error, and a method for doing this is suggested.

The procedures discussed here provide a systematic method for quickly changing the accuracy of both the interior propagation and wall boundary condition implementations to suit flow conditions. It was shown that very high-order methods (>15) provide an opportunity for subgrid-scale resolution.

These results demonstrate the potential value of very high accuracy in time as well as in space for aeroacoustic calculations. However, a detailed stability analysis for these procedures remains to be done for the generalized high-order surface conditions. A method for unrolling nonlinear Cauchy–Kowalewski recursion needs to be developed to achieve arbitrarily high accuracy in time for nonlinear problems. Finally, arbitrary order accuracy nonreflecting boundary conditions need to be developed to take full advantage of the arbitrary interior domain accuracy.

Nonetheless, it is necessary to use Hermitian data for achieving very high-order solutions because Lagrangian stencils simply become too large. Because since small stencils allow for a very simple interpolation procedure of arbitrary accuracy while maintaining CFL stability near boundaries, future work will be devoted to exploiting them in substantial applications.

References

[1] Lele, S. K., "Computational Aeroacoustics: A Review," AIAA Paper 97-0018, Jan. 1997.

[2] Tam, C. K. W., "Computational Aeroacoustics: Issues and Methods," *AIAA Journal*, Vol. 33, No. 10, 1995, pp. 1788–1796.

[3] Vemuri, V., and Karplus, W., "Higher-Order Difference Approximations," *Digital Computer Treatment of Partial Differential Equations*, Prentice–Hall, Upper Saddle River, NJ, 1981, pp. 92–98.

[4] Goodrich, J. W., "High Order Implementations of Accurate Boundary Conditions," AIAA Paper 99-1942, May 1999.

[5] Tam, C. K. W., "Advances in Numerical Boundary Conditions for Computational Aeroacoustics," AIAA Paper 97-1774, 1997.

[6] Carpenter, M. H., Nordstrom, J., and Gottlieb, D., "A Stable and Conservative Interface Treatment of Arbitrary Spatial Accuracy," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 98-12, Hampton, VA, Feb. 1998.

[7] Carpeneter, M. H., and Casper, J. H., "The Accuracy of Shock Capturing in Two Spatial Dimensions," AIAA Paper 97-2107, July 1997.

[8] Goodrich, J. W., "A Low Order and a High Order Solution For A Converging-Diverging Nozzle Problem," *Third Computational Aeroacoustics Workshop*, NASA CP 2000-209-790, Aug. 2000.

[9] Woodward, P., and Collela, P., "The Numerical Solution of Two- Dimensional Fluid Flow with Strong Shocks," *Journal of Computational Physics*, Vol. 54, No. 1, 1984, pp. 115–173.

[10] Casper, J., and Carpeneter, M. H., "Computational Considerations for the Simulation of Shock-Induced Sound," *SIAM Journal of Scientific and Statistical Computation*, Vol. 19, No. 3, 1998, pp. 813–828.

[11] Goodrich, J. W., "High Accuracy Finite Difference Algorithms for Computational Aeroacoustics," AIAA Paper 97-1584, May 1997.

[12] Lee, T. K., and Zhong, X., "Spurious Numerical Oscillations in Simulation of Supersonic Flows Using Schock-Capturing Schemes," *AIAA Journal*, Vol. 37, No. 3, 1999, pp. 313–319.

[13] Lockard, D. P., Brentner, K. S., and Atkins, H. L., "High-Accuracy Algorithms for Computational Aeroacoustics," *AIAA Journal*, Vol. 33, No. 2, 1995, pp. 246–251.

[14] Tam, C. K. W., and Dong, Z., "Wall Boundary Conditions for High-Order Finite-Difference Schemes in Computational Aeroacoustics," *Theoretical Computational Fluid Dynamics*, Vol. 6, 1994, pp. 303–322.

[15] Strand, B., "Simulations of Acoustic Wave Phenomena Using High-Order Finite Difference Approximations," *SIAM Journal of Scientific and Statistical Computation*, Vol. 20, No. 5, 1999, pp. 1585–1604.

[16] Butcher, J. C., "The Non-Existence of Ten Stage Eighth Order Explicit Runge–Kutta Methods," *BIT*, Vol. 25, No. 3, 1985, pp. 521–540.

[17] Buchberger, B., "Gröebner Bases in Mathematica: Enthusiam and Frustration" RISC-Linz Technical Rept. Series No. 91-11, Linz, Austria, 1991.

[18] Cox, D., Little, J., and O'Shea, D., *Ideals, Varieties, and Algorithms*, 2nd ed., Springer-Verlag, 1996, Chaps. 1, 2.

[19] Mayr, E., and Meyer, A., "The Complexity of the Word Problem for Commutative Semigroups and Polynomial Ideals," *Advanced Mathematics*, Vol. 46, No. 3, 1982, pp. 305–329.

[20] Wolfram Research, "Numerical Mathematics," Mathematica 4.0: Standard Add-On Packages, Wolfram Media, Champaign, IL, 1999, pp. 354–363.

[21] Sofroniou, M., "Symbolic Derivation of Runge–Kutta Methods," *Journal of Symbolic Computation*, Vol. 18, No. 3, 1994, pp. 265–296.

[22] Stanescu, D., and Habashi, W. G., "2N-Storage Low Dissipation and Dispersion Runge–Kutta Schemes for Computational Acoustics," *Journal of Computational Physics*, Vol 143, No. 2, 1998, p. 674.

[23] Goodrich, J. W., "A Comparison of Numerical Methods for Computational Aeroacoustics," AIAA Paper 99-1943, May 1999.

[24] Goodrich, J. W., "Application of a New High Order Finite Difference Scheme to Acoustic Propagation With the Linearized Euler Equation," NASA TM-1993-106454, June 1993.

[25] Goodrich, J. W., "Accurate Finite Difference Algorithms for Computational Aeroacoustics," *Computational Fluid Dynamics Review 1998*, edited by M. M Hafez and K. Oshima, Vol. 2, World Scientific, Singapore, 1998, pp. 1034–1051.

[26] Garabedian, P. R., "The Method of Power Series: The Cauchy–Kowalewski Theorem," *Partial Differential Equations*, 2nd ed., Chelsea, New York, 1986, pp. 6–16.

[27] Dyson, R. W., and Goodrich, J. W., "A Very High Order, Adaptable MESA Implementation for Aeroacoustic Computations," NASA TM-2000-209944, April 2000.

[28] Botha, J. F., and Pinder, G. F., "Two-Dimensional Polynomial Approximation," *Fundamental Concepts in the Numerical Solution of Differential Equations*, Wiley, New York, 1983, pp. 17–22.

[29] Dyson, R. W., "An Automated Code Generator For Three-Dimensional Acoustic Wave Propagation With Geometrically Complex Solid Wall Boundaries," Ph.D. Dissertation, Computer Engineering Dept., Case Western Reserve Univ., Cleveland, OH, May 1999; also NASA TM-1999-209182, 1999.

[30] Berger, M. J., and Leveque, R. J., "Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2298–2316.

[31] Kurbatskii, K. A., and Tam, C. K. W., "Cartesian Boundary Treatment of Curved Walls for High-Order Computational Aeroacoustics Schemes," *AIAA Journal*, Vol. 35, No. 1, 1997, pp. 133–140.

[32] Wirth, M. C., “On the Automation of Computational Physics,” Ph.D. Dissertation, Engineering-Applied Science, Univ. of California, Davis, CA, Sept. 1980.

[33] Steinberg, S., and Roache, P., “Automatic Generation of Finite Difference Code,” *Symbolic Computation in Fluid Mechanics and Heat Transfer*, edited by H. H. Bau, T. Herbert, and M. M Yovanovich, American Society of Mechanical Engineers, Fairfield, NJ, 1988, p. 81.

[34] Goodrich, J. W., “An Approach to the Development of Numerical Algorithms for First Order Linear Hyperbolic Systems in Multiple Space Dimensions: The Constant Coefficient Case,” NASA TM-1995-106928, Sept. 1995.

[35] Powell, M. J. D., “Hermite Interpolation,” *Approximation Theory and Methods*, Cambridge Univ. Press, Cambridge, England, U.K., 1981, pp. 53–57.

[36] Lax, P. D., and Wendroff, B., “Systems of Conservation Laws,” *Communications on Pure and Applied Mathematics*, Vol. 13, No. 2, 1960, pp. 217–237.

[37] Fulks, W., “The Gradient. The Del Operator. Directional Derivatives,” *Advanced Calculus: An Introduction to Analysis*, Wiley, New York, 1978, pp. 315–320.

[38] Zauderer, E., “Classification of Equations in General and their Characteristics,” *Partial Differential Equations of Applied Mathematics*, 2nd ed., Wiley, New York, 1989, pp. 121–136.

[39] Winston, P. H., “Propagating Symbolic Constraints,” *Artificial Intelligence*, Addison Wesley Longman, Reading, MA, 1984, pp. 46–71.

[40] Sklar, B., “The Sampling Theorem,” *Digital Communications: Fundamentals and Applications*, Prentice–Hall, Upper Saddle River, NJ, 1988, pp. 59, 60.

[41] Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw–Hill, New York, 1956, pp. 35–175.

P. J. Morris
Associate Editor