A01-16899

AIAA 2001-1118

Parallel Computation of Complex Aeroacoustic Systems

Foluso Ladeinde & Xiaodan Cai
Aerospace Research Corp., L.I.
Stony Brook, L.I., New York 11790

Miguel R. Visbal & Datta V. Gaitonde
Air Vehicles Directorate, AFRL
Wright-Patterson AFB, OH 45433-7521

39th AIAA Aerospace Sciences Meeting & Exhibit
8-11 January 2001 / Reno, NV

Parallel Computation of Complex Aeroacoustic Systems

Foluso Ladeinde*and Xiaodan Cai†
Aerospace Research Corp., L.I.
25 East Loop Road,
Stony Brook, NY 11790-0609

Miguel R. Visbal‡and Datta V. Gaitonde§
Air Vehicles Directorate
Air Force Research Laboratory
Wright-Patterson AFB, OH 45433

Abstract

The ability of high-order compact differencing and filtering schemes to compute realistic aeroacoustic situations is examined. The strong conservation form of the Euler equations are employed in a curvilinear coordinate system with particular emphasis on recently developed procedures which minimize freestream preservation errors. A powerful filter-based absorbing boundary condition is also utilized. Time-integration was achieved with either the fourth-order classical R-K method or with a third-order, iterative, approximate-factorization implicit scheme. The algorithm is formulated for use on massively parallel platforms, with specific focus on the SGI Origin 2100 computer. Several canonical problems have been solved to establish the accuracy of the overall implementation. These include propagation of a spherical pulse and scattering from a cylinder. Finally, a preliminary analysis has been conducted of acoustic scattering from a generic aerospace vehicle configuration. These calculations, which employ a domain-decomposition approach, demonstrate that the various components of the scheme are suitable for use on realistic geometries, particularly when executed on parallel machines.

*Senior Member, AIAA; Director of Research
†Member, AIAA; Senior Research Engineer
‡Associate Fellow, AIAA; Technical Area Leader
§Associate Fellow, AIAA; Senior Research Aerospace Engineer

1. Introduction

The impact of aerodynamically-generated sound on communities and structures is an important aspect of both civilian and military aircraft operation. Weapon cavity acoustics, jet screech, sonic boom, cabin noise and sound generated by blade/vortex interaction are examples of applications. The need to meet more stringent community noise level standards has resulted in increased attention being paid to the relatively new field of time-domain computational aeroacoustics (CAA). However, aeroacoustic predictions are complicated by the requirement for high accuracy, low dissipation and dispersion, treatment of outflow radiation conditions, complicated geometries, and demanding computational load.

Recent reviews of CAA have been given by Tam [1] and Wells and Renault [2] who discussed various numerical schemes. These include, among others, the dispersion-relation-preserving (DRP) scheme of Tam and Webb [3], the method of minimization of group velocity errors due to Holberg [4], the compact differencing schemes [5], and the essentially non-oscillatory (ENO) schemes [6].

The emphasis of the present work is on the simulation of realistic aerodynamic systems which usually involves complicated geometries and requires large computational resources. Therefore, the method has to be carefully selected in terms of the numerical difficulties associated with poor mesh quality in a curvi-

linear coordinate formulation and the ability to minimize metric cancellation and freestream preservation errors. The DRP scheme was developed with aeroacoustics in mind. However, the application of the method to realistic engineering geometries has not received enough attention, although the method was implemented in curvilinear coordinates in [7]. The work of Visbal and Gaitonde [8] is also relevant in the present context. They developed and implemented a high-order, compact-differencing and filtering algorithm to simulate aeroacoustic phenomena over curved geometries. An important aspect of their procedure pertains to the successful demonstration in highly curvilinear systems and the use of a high-order filter procedure as an alternative to the asymptotic treatment of outflow radiation boundary conditions in [9].

The management of the computational load associated with aeroacoustic computations with compact schemes is an important contribution for realistic systems. To this end, the efforts at NASA by Hixon [9] and Mankbadi, Hixon, and Povinelli [10] are relevant. These authors use the compact schemes in the prefactored form with the aim of reducing the computational speed. In a recent study [10], CPU time reduction was accomplished by solving only for what the authors call the "very large scale structures" of the flow and noise generation. They used the $k-\varepsilon$ turbulence model to account for the effect of the unresolved scales.

In the present work, the approach taken toward the analysis of realistic systems is based on a combination of the fourth- and sixth-order compact scheme, high-order filtering scheme, and the execution of the procedures in massively parallel computers. Parallel issues relevant to the compact schemes have been investigated by the authors [11], wherein three procedures (the one-sided method, the parallel diagonal dominant method, and the parallel Thomas algorithm) were rigorously analyzed for their computational advantages. From the studies, the one-sided procedure appeared to be the best based on a compromise between simplicity, extension to realistic systems, and accuracy. One major issue with the method pertains to the accuracy of the solution at the interface between subdomains. However, recent work by the authors seems to suggest that the procedure could give accurate results if the number of overlapped cells is five or greater and an appropriate filtering scheme is invoked.

The present work demonstrates the applicability of parallel aeroacoustics computation to complex aerodynamic systems, using the one-sided parallelization approach. In Section 2, the governing equations are presented, followed by the numerical schemes in Section 3 and results in Section 4. Concluding remarks are presented in Section 5.

2. Governing Equations

The relevant equations are the inviscid form of the Euler equations written in strong conservation form for generalized curvilinear coordinates (ξ, η, ζ):

$$\frac{\partial}{\partial t}\left(\frac{q}{J}\right) + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = \frac{\vec{S}}{J}, \tag{1}$$

where $q = \{\rho, \rho u, \rho v, \rho w, \rho E_t\}$ is the solution vector, J is the Jacobian of the coordinate transformation, F, G, H are the inviscid fluxes, and $\vec{S}$ is a vector included to account for acoustic sources. The fluxes are

$$F = \frac{1}{J}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (\rho E + p) U \end{bmatrix} \tag{2}$$

$$G = \frac{1}{J}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (\rho E + p) V \end{bmatrix} \tag{3}$$

$$H = \frac{1}{J}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (\rho E + p) W \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} U &= \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_x u + \zeta_y v + \zeta_z w \\ E &= \frac{T}{(\gamma-1)M_\infty^2} + \frac{1}{2}(u^2 + v^2 + w^2). \end{aligned} \tag{5}$$

Note that (x, y, z) are the Cartesian coordinate direction components, (ξ, η, ζ) the coordinates in the transformed plane, (u, v, w) the vector of Cartesian velocity components, (U, V, W) the contravariant velocity components, ρ the density, p the pressure, T the temperature, and M_∞ the freestream Mach number. The perfect gas law $p = \rho RT$ is assumed.

3. Numerical Procedure

A finite-difference method is used to discretize the equations given above. Details of the numerical procedures are provided below.

3.1 Differencing Scheme

With the compact schemes, the derivative u' for any generic variable u in the transformed coordinate frame is represented as

$$\alpha u'_{i-1} + u'_i + \alpha u'_{i+1} = b\frac{u_{i+2} - u_{i-2}}{4\Delta\xi} + a\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}, \quad (6)$$

where α, a, and b are constants which determine the spatial properties of the algorithm. The base compact differencing schemes used in this paper are the three-point, fourth order scheme, C4, with $(\alpha, a, b) = (\frac{1}{4}, \frac{3}{2}, 0)$, the five-point, sixth order scheme, C6, with $(\alpha, a, b) = (\frac{1}{3}, \frac{14}{9}, \frac{1}{9})$ and the five-point, fourth order scheme. Note that the symbol u above also represents components of vector quantities such as the F vector defined in equation (2).

Equation (6) is used to calculate the various derivatives in the (ξ, η, ζ) plane, as well as the metrics of the coordinate transformation. The derivatives of the inviscid fluxes are obtained by first forming these fluxes at the nodes and subsequently differentiating each component with the above formulas. In order to reduce the error on stretched meshes, the required metrics are computed with the same scheme as employed for the fluxes.

The physical boundary conditions are applied after each update of the interior solution vector. These conditions include Dirichlet and Neumann (extrapolation and symmetry) conditions. For the inviscid calculations, at Dirichlet nodes, the normal velocity component is set to zero, whereas the gradient of the other velocity components and of the pressure, density, and energy are set to zero. Similar conditions are also enforced on symmetry planes.

3.2 Interior Filtering Scheme

Filters are employed to numerically stabilize the compact differencing calculations. In the formulation, the filtered values $\widetilde{u}$ for any quantity u in the transformed space is represented as:

$$\alpha_f \widetilde{u}_{i-1} + \widetilde{u}_i + \alpha_f \widetilde{u}_{i+1} = \sum_{n=0}^{N} \frac{a_n}{2}\left(u_{i+n} + u_{i-n}\right). \quad (7)$$

For multi-dimensional problems, the filter is applied sequentially in each of the three directions. This equation, with proper choice of coefficients, provides a $2N^{\text{th}}$-order formula on a $2N+1$ point stencil. The $N+1$ coefficients, $a_0, a_1 ... a_N$, are derived in terms of α_f with Taylor- and Fourier-series analyses and are listed in [12]. Thus Eqn. 7 can be written as

$$\alpha_f \widehat{\phi}_{i-1} + \widehat{\phi}_i + \alpha_f \widehat{\phi}_{i+1} = f_{2N}\left(\alpha_f, \phi_{i-N}, ... \phi_{i+N}\right),$$

where the right hand side is known once α_f and the order of accuracy, $2N$, are chosen. On uniform meshes, the resulting filters are non-dispersive. They do not amplify any waves and they preserve constant functions and completely eliminate the odd-even mode. Since α_f is a free parameter, an explicit filter, i.e., one that does not require the solution of a tridiagonal matrix, can be easily extracted by setting $\alpha_f = 0$. The primary constraint on α_f is that it must satisfy the inequality $-0.5 < \alpha_f < 0.5$. In this range, higher values of α_f correspond to a less dissipative filter. At $\alpha_f = 0.5$, Eqn. 7 reduces to an identity and there is no filtering effect. Detailed spectral responses of these filters may be found in Refs. 12 and 13.

Computations on a range of 2-D and 3-D problems suggest that on meshes of reasonable quality, a value $0.3 \leq \alpha_f < 0.5$ is appropriate. Only in cases where the mesh is of extremely poor quality, if it contains metric discontinuities, for example, will a lower value of $\alpha_f \sim 0.1$ be required. The impact of filtering on the fully discretized 1-D advection equation with periodic end conditions has been examined in Ref. 13.

The relatively large stencils of high-order filters require special formulations at several points near the boundaries. For instance, the 10th order interior filter requires an 11-point stencil and thus can not be applied at the "near-boundary" points 1, 2...5 and correspondingly at $IL-4, ...IL$, where it protrudes the boundary. The values at points 1 and IL are specified explicitly through the boundary conditions and are not filtered. At the remaining near-boundary points, two approaches have been noted in the literature. In Ref. 11, it was suggested that lower-order centered formulas be applied near the boundaries with appropriate ad-

justment (or optimization) of the value of α_f. This approach is based on the observation (see Ref. 14) that, for any given order of accuracy, as values of α_f approach 0.5, the dissipative effect of the filter is muted. The second method, introduced in Ref. 20 employs higher-order one-sided formulas. For the problems of present interest, either approach may be employed. Due to its simplicity, all computations reported in this work utilize the first approach.

3.3 Metric Evaluation

Freestream preservation and metric cancellation errors have to be ensured in order to extend high-order schemes to non-trivial three-dimensional generalized curvilinear coordinates systems. These errors arise in finite difference discretizations of governing equations written in strong conservation form and could easily degrade the fidelity of high-order calculations. Grid-induced errors may appear, for instance, in regions of large grid variations or near singularities. Pulliam and Steger [15] introduced a simple averaging procedure which guarantees freestream preservation on three-dimensional curvilinear meshes. Unfortunately, this procedure, which works very well for second-order scheme, is difficult to extend to high-order formulations. An alternative method to enforce the metric identities consists of writing the metric relations in conservative form:

$$\xi_x/J = (y_\eta z)_\zeta - (y_\zeta z)_\eta \tag{8}$$

$$\eta_x/J = (y_\zeta z)_\xi - (y_\xi z)_\zeta \tag{9}$$

$$\zeta_x/J = (y_\xi z)_\eta - (y_\eta z)_\xi. \tag{10}$$

Similar relations apply for the other metric terms.

3.4 Time Integration

The developed procedure allows the use of the classical fourth-order four-stage Runge-Kutta method and the time-accurate implementation of the Beam-Warming approximate factorization methods. With R denoting the residual, the governing equation is:

$$\frac{\partial U}{\partial t} = R = -J\left(\frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta}\right) - S$$

The classical four-stage Runge-Kutta method integrates from time t_0 (step n) to $t_0 + \Delta t$ (step $n+1$) through the operations

$$k_0 = \Delta t R(U_0) \quad k_1 = \Delta t R(U_1)$$
$$k_2 = \Delta t R(U_2) \quad k_3 = \Delta t R(U_3)$$

$$U^{n+1} = U^n + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + 2k_3)$$

where $U_0 = U(x, y, z, t_0), U_1 = \frac{k_0}{2}, U_2 = U_1 + \frac{k_1}{2}, U_3 = U_2 + k_2$. The scheme is implemented in the low-storage form described in Ref. 16, requiring 3 levels of storage for each variable.

Time-accurate solutions to the Euler equations were also obtained numerically by the implicit approximately, factorized finite-difference algorithm of Beam and Warming employing Newton-like subiterations, which has evolved as an efficient tool for generating solutions to a wide variety of complex fluid flow problems, and may be represented notationally as:

$$\left[I + \left(\frac{6\,\Delta\,t}{11}\right)\delta_\xi\left(\frac{\partial F^p}{\partial Q}\right)\right] \times$$
$$\left[I + \left(\frac{6\,\Delta\,t}{11}\right)\delta_\eta\left(\frac{\partial G^p}{\partial Q}\right)\right] \times$$
$$\left[I + \left(\frac{6\,\Delta\,t}{11}\right)\delta_\zeta\left(\frac{\partial H^p}{\partial Q}\right)\right]\Delta Q$$
$$= -\left(\frac{6\,\Delta\,t}{11}\right)\left[\left(\frac{1}{6\,\Delta\,t}\right)(11Q^p - 18Q^n + 9Q^{n-1} - 2Q^{n-2})\right.$$
$$\left. + \delta_\xi F^p + \delta_\eta G^p + \delta_\zeta H^p\right]$$

In this expression, which was employed to advance the solution in time, Q^{p+1} is the $p+1$ approximation to Q at the $n+1$ time level Q^{n+1}, and $\Delta Q = Q^{p+1} - Q^p$. For $p = 1$, $Q^p = Q^n$. This procedure is implicit and third-order in time. The spatial difference operators appearing in the explicit portion of the algorithm (right-hand side) were evaluated by a sixth-order compact difference scheme. For convenience, the sourse term S has been treated explicitly, which does not adversely impact stability due to use of subiteration.

Temporal accuracy, which can be degraded by use of the diagonal form, is maintained by utilizing subiterations within a time step. This technique has been commonly invoked in order to reduce errors due to factorization, linearization, and explicit application of boundary conditions. It is useful for achieving temporal accuracy on overset zonal mesh systems, and for a domain decomposition implementation on parallel computing platforms. Any deterioration of the solution caused by the use of artificial dissipation and by lower-order spatial resolution of implicit operator is also reduced by the procedure. Three subiterations per time step have been applied for the computations presented here.

3.5 Parallelization Strategy

The parallelization procedure used for the aeroacoustic computations is based on domain decomposition which has been implemented within the framework of compact schemes. The strategy, which we refer to as one-sided [11], involves the advancement of the solution independently in each subdomain, with individual interior and boundary formulas used in the same manner as in single-domain computations. Data is exchanged between adjacent subdomains at the end of each sub-iteration of the implicit scheme (or each stage of RK4), as well as after each application of the filter. Gaitonde & Visbal[17] applied the interface algorithm to 2D inviscid and viscous flow calculations using the compact scheme in a sequential execution mode. It was found that the lower-order one-sided boundary scheme could cause a serious distortion of the flow structure but that this distortion could be reduced by superior higher-order one-sided filter formulas and a deeper overlap size.

4. Results

4.1 Parallel Performance

The parallelization of the compact schemes is not a trivial matter, due to the implicit nature of the equations. Although we have chosen the one-sided method to parallelize our code, the choice is based on its simplicity, which means that the method could be applied to realistic geometries. As we show later, the procedure also tends to have a superior parallel performance. The parallel tridiagonal solvers, whose algorithmic details have been reported by the authors [11], are an alternative to the one-sided method. Unlike the latter, they recover the single-domain results, provided a conforming mesh system is used. Examples of parallel diagonal solvers include the transposed[18], the pipelined[19] (or Parallel Thomas Algorithm, PTA) and Sun's[20] distributed (Parallel Diagonal Dominant, PDD) methods. The comparative performance of these methods are shown in Table 1 (a) (CPU times) and in Table 1 (b) (speedup). The results were generated for a kernel problem for the inversion of a tridiagonal system of equations with $N = 400$ (i.e., total number of nodal points in the $x-$ or derivative–direction), $N_1 = 400$ (i.e., number of nodal points in the vertical or vector direction). N_2 in the Table 1 (a) is the overlap depth in terms of grid points. The numbers in parenthesis in Table 1 (b) are the values actually observed (measured) in our numerical experiments on the IBM SP2 machine at Cornell University. Note that the domain is decomposed only in the $x-$direction for the results in these tables.

The performance data for the IBM SP2 system are: start-up latency, $\alpha = 55\mu$s, point-to-point communication, $1/\beta$=17.5 MWords/s, and time to perform one floating point operation, $\gamma = 1/65$ μs. Note that $\beta = 1/17.5$ μs, is the time required to send double-precision data. Thus, $\alpha/\beta \approx 966$, which is a large number (compared to unity), indicating that it is costly to initiate the process of sending a message (big or small) in this system and that messages should be bundled. The system is rated at 266 Mflops/s at peak performance, although the measured values are 65 Mflops/s (block tridiagonal matrix calculation) and 85 Mflops/s (multi-grid calculation). In Table 1, P denotes the number of processors, k is the number of groups in the pipelined algorithm; j is the reduced number in Sun's algorithm, which is usually not larger than 10 for the compact difference scheme. Note that the optimal parameter k can be expressed as

$$k = \left\lceil \frac{N_1}{\sqrt{\frac{N_1 * P/N * \nu}{\rho(P-1)}}} \right\rceil , \qquad (11)$$

where $\nu = \frac{\alpha}{g_2}, \rho = \frac{g_1}{g_2}$, and g_1 and g_2 are the forward and backward calculation times for the TDMA per grid point. To produce Tables 1 (a) and 1 (b), we choose Sun's reduced PDD approach [20] to represent the distributed algorithms, which appears to be the most efficient parallel solver in this category, and the unoptimized Povitsky[19] method to represent the pipelined algorithms. The transposed algorithm (not shown) was originally developed by Cai, Ladeinde, and O'Brien[18] for FFT, but has also been applied to the tridiagonal system at hand. The TDMA algorithm in the table is the standard Tridiagonal Thomas Algorithm. From Table 1 (a), one can see that Sun's algorithm incurs a smaller communication cost compared to PTA, and therefore should be preferred on machines capable of handling computations much faster than they do communication.

Although Table 1 shows the one-sided method to be the slowest of the three parallelization strategies, it is still the case that the procedure is relatively very easy to implement. Therefore, its parallel performance

was further investigated with a three-dimensional domain splitting in which each processor computed the grid $60 \times 30 \times 30$ in (x, y, z). This is the useful grid in that it excludes two of the five grid points shared with the neighboring processor in the overlap region on each side of a subdomain. The physical system calculated here is a laminar boundary layer flow. Sequential calculations corresponding to each parallel case are needed in order to calculate speedup. The base sequential mesh is $60 \times 30 \times 30$ or 54×10^3 grid points. The results, which are shown in Figures 1 (a) and 1 (b) are quite interesting, as discussed below. The grid layout in the (x, y, z) directions of the sequential mesh is chosen to mimic the processor decomposition. Thus, for the domain decomposition $2 \times 4 \times 1$, for example, the grid layout for the sequential calculation is $(120, 120, 30)$, or $(60 \times 2, 30 \times 4, 30 \times 1)$, which is 432×10^3 nodal points. Table 2 shows that processor decomposition (e.g., (2,1,1)) versus (1,2,1) does not significantly affect the CPU time performance. Also, for the sequential calculations, the total number of grid points, not the grid layout in (x, y, z), govern the performance. Scalability results are presented in Figure 1 (a), wherein the speedup is plotted against the number of processors; each processor calculates $60 \times 30 \times 30$.

In Figure 1 (b), the CPU time performance is presented as a function of the number of grid points for the sequential calculations. The (ideal) solid line in this figure is based on a linear scaling of the CPU time, using the grid $60 \times 30 \times 30$ (or 54×10^3 grid points) as the base for the extrapolation. It is evident that the observed CPU performance (dashed line) does not scale linearly with the number of grid points. In general, if the CPU time for the sequential calculation of the base grid is T_0, that for the sequential calculation of $54 \times n \times 10^3$ grid points, say T_{sn}, is greater than nT_0, as shown by the larger values of the dashed line data over the corresponding solid line results (Figure 1 (b)). The speedup is $\frac{T_{sn}}{T_0+T_c}$. That is, the speedup can go above n (Figure 1 (a)), depending on the values of $(T_{sn} - nT_0)$ relative to T_c. Note that for all cases, T_0 is the same because, even though the size of the sequential problem (and hence T_{sn}) changes, that in each processor (and hence T_0) is fixed.

4.2 Evaluation of 3D Aeroacoustic Computation

The basic computational schemes implemented in this paper have been rigorously validated for single-domain, but mostly two-dimensional aeroacoustic calculations in previous studies by the authors [7, 21]. This section discusses the validation for three-dimensional aeroacoustic computation.

4.2.1 Spherical Acoustic Pulse on a 3-D Curvilinear Mesh

This validation case considers the propagation of a spherical pulse in a three-dimensional curvilinear mesh. An initial pressure pulse is prescribed by

$$p = p_\infty + \epsilon e^{-\ln 2 \frac{(x^2+y^2+z^2)}{9}},$$

where $\epsilon = 0.01$. In order to examine metric cancellation errors, a three-dimensional curvilinear mesh shown in Figure 4 is generated using the following equations

$$x_{i,j,k} = x_{\min} + \Delta x_0 \left[(i-1) + A_x \sin \frac{n_{xy}\pi(j-1)\Delta y_0}{L_y} \sin \frac{n_{xz}\pi(k-1)\Delta z_0}{L_z} \right]$$

$$y_{i,j,k} = y_{\min} + \Delta y_0 \left[(j-1) + A_y \sin \frac{n_{yz}\pi(i-1)\Delta z_0}{L_x} \sin \frac{n_{yz}\pi(k-1)\Delta z_0}{L_z} \right]$$

$$z_{i,j,k} = z_{\min} + \Delta z_0 \left[(k-1) + A_z \sin \frac{n_{zx}\pi(i-1)\Delta x_0}{L_x} \sin \frac{n_{zy}\pi(j-1)\Delta y_0}{L_y} \right]$$

where

$$i = 1, ..., IL; \quad j = 1, ..., JL, \quad k = 1, ..., KL$$

$$\Delta x_0 = \frac{L_x}{IL-1}; \quad \Delta y_0 = \frac{L_y}{JL-1}; \quad \Delta z_0 = \frac{L_z}{KL-1}.$$

The grid $(IL, JL, KL) = (61, 61, 61)$, $(L_x, L_y, L_z) = (60, 60, 60)$, and $n_{xy} = n_{yz} = ... = 8$. These parameters yield a mesh in which the metric identities are not trivially satisfied.

The pulse propagation problem is computed with RK4 ($\Delta t = 0.004$) using the fourth-order compact differencing, tenth-order filtering scheme with $\alpha_f = 0.49$. The perturbation pressure along the grid line $i = j = 31$ (Figure 3) is compared for the present procedure and the exact solution. The numerical results for both the standard metrics and the conserving form presented above are shown. It is evident that, whereas the standard metrics yield the wrong results, the new metrics give results that are in perfect agreement with the theory. Note that the use of a sixth-order compact

scheme (not shown) displayed reduced sensitivity to the choice of metric evaluation procedure, consistent with [22], where metric cancellation errors were shown to decrease with order of accuracy. Nonetheless, all solutions on this highly distorted mesh were found to be of poor quality if the non-conserving metric evaluation procedures are used. In this case, the freestream preservation errors could pollute the acoustic pressure solutions. The present results clearly demonstrate that high-order compact schemes can be successfully extended to general curvilinear grids, making them suitable for complex aerodynamic configurations.

4.2.2 Three-Dimensional Scattering of Acoustic Pulse from a Cylinder

Another validation of the parallel three-dimensional approach developed in this paper is the scattering of acoustic pulse from a cylinder, the two-dimensional version of which is denoted as Category I, Problem 2 in the second CAA Workshop of [23]. The original problem is extended to three dimensions in this paper, by introducing the z coordinate direction. The additional boundary condition of periodic solutions is imposed in the $z-$direction. The pulse is given by

$$p = p_\infty + \epsilon e^{-\ln 2 \frac{(x-x_c)^2+(y-y_c)^2}{b^2}},$$

with $x_c = 4$, $y_c = 4$, $\epsilon = 0.01$, $b = 0.2$. Along the cylinder surface, the normal velocity is set to zero, whereas the normal gradients of the tangential and axial velocity components, pressure, and density are set to zero. Since the configuration is symmetric, only the upper half of the configuration is considered, and symmetry conditions are applied along the surface $(r, \theta = 0, 180, z)$. Note that a generalized curvilinear coordinate formulation (not a polar coordinate system) is used and the coordinate directions were interchanged in order to test any bias.

A coarse version of the computational mesh is shown in Figure 4, whereas, in Figure 5, the pressure results are compared for the sequential two-dimensional and the present parallel three-dimensional calculations. The agreement between the two sets of results is evident. Figure 6 shows the uniformity of the solution along the z-direction, as one would expect. From the foregoing, one can conclude that the treatment of the interface between subdomains does not degrade the solutions relative to the single domain results.

4.3 Acoustic Scattering by a Complex Configuration

In order to demonstrate the capability of the present method to treat complicated aeroacoustic phenomenon over realistic configurations, we consider the scattering of a spherical pulse by a generic aerospace vehicle (the X24C) for which a body-fitted grid system was readily available. The final problem used to illustrate the application of our procedure is the acoustic scattering by the X24C reentry vehicle. The pulse is specified as

$$p = p_\infty + \epsilon e^{-\ln 2 \frac{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}{b^2}},$$

where

$$p_\infty = \frac{1}{\gamma M_\infty^2}, \epsilon = 0.01, b^2 = 0.1.$$

and

$$x_c = 0.2978, \qquad y_c = 0.2995, \qquad z_c = 0.$$

A sixth-order compact scheme is used in the interior with fourth-order on the boundary. The interior filter is tenth-order, whereas the four nodes in the vicinity of the boundary (including interface boundaries) use filter schemes of orders 2,4, 6, and 8, respectively. The calculations were done with the third-order, implicit Beam-Warming procedure using $\Delta t = 10^{-3}$. Note that the use of RK4 for this problem required a Δt that is two orders of magnitudes smaller than this value. The calculations were done for two grids: $120 \times 80 \times 121$ and $60 \times 40 \times 61$. The domain is decomposed as $2 \times 2 \times 2$ and mapped into eight processors on SGI 2100. The transformed curvilinear coordinates $\xi(i), \eta(j), \zeta(k)$ are aligned with the streamline, body normal, and transverse directions, respectively. Figure 7 shows the surface grid (J=0) while Figure 8 is the outer boundary (J=61). Projected views of four stations (I=15, 30, 45, and 60) are shown in Figure 9. The actual views of the surfaces is shown in Figure 10. The computed pressure on these surfaces are shown in Figure 11. In Figure 12, the projections of four surfaces along the ζ direction are shown. The pressure distribution on these surfaces is shown in Figure 13.

The acoustic simulation exercise for the X24C reentry vehicle, as shown above, is preliminary and has not been examined in detail from a numerical perspective. However, the calculations do not show any unusual behavior for this complicated problem. Therefore, the high-order procedure holds promise for aeroacoustic analysis of complex configurations.

5. Conclusion

In this paper, we have demonstrated the usefulness, through the calculation of realistic aeroacoustic systems, of the high-order compact differencing and filtering schemes developed and implemented in [8]. The nonlinear Euler equations were analyzed in the strong conservation form and in a generalized curvilinear coordinate system. The procedures were carefully implemented to minimize freestream preservation errors and to provide a robust, yet accurate, treatment of outflow radiation conditions. For time integration, both the standard fourth-order Runge-Kutta scheme and third-order Beam-Warming scheme were investigated. The analysis of realistic systems with the developed procedures was made possible by the execution of the resulting code on parallel machines. The computations reported in this paper were done on the multi-processor SGI Origin 2100 computer, with the MPI message passing protocol. The analysis of a spherical pulse on a 3D curvilinear mesh and the 3D scattering of an acoustic pulse from a cylinder was used to validate the accuracy of the parallel computations. Nonetheless, for this complicated geometry, no unusual behavior in the results were observed, especially at the interface between subdomains. A preliminary acoustic simulation for the X24C vehicle is also reported in this paper. When executed on parallel machines, the developed procedures are shown to be effective in the simulation of aeroacoustics on complex geometries.

Acknowledgments

The authors acknowledge the assistance of P. Morgan in providing the generic reentry vehicle grid system. The first two authors will like thank Aerospace Research Corporation, L.I. for supporting this work and for giving the permission to publish it.

References

[1] C. K. W Tam. Computational Aeroacoustics: Issues and Methods. AIAA J., 33(10):1788-1796, 1995

[2] V. L. Wells and R. A. Renault. Computing Aerodynamically Generated Noise. Annual Review of Fluid Mechanics, 29, pp. 161-199, 1997

[3] C. K. W. Tam and J. C. Webb. Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics. Journal of Computational Physics, 107: 262-281, 1993

[4] O. Holberg. Computational Aspects of the Choice of Operator and Sampling Interval for Numerical Differentiation in Large-Scale Simulation of Wave Phenomenon. Geophys. Prospect, 35, pp. 629-655, 1987.

[5] S.K. Lele, Compact Finite Difference Schemes With Spectral-like Resolution, Journal of Computational Physics, 103 (1992) 16-42.

[6] J. Casper. Using High-Order Accurate Essentially Nonoscillatory Schemes for Aeroacoustic Application. AIAA Journal, 34, pp. 244-250, 1994

[7] F. Ladeinde, X. Cai, M. Visbal, and D. Gaitonde. Application of DRP and Compact Schemes to Computational Aeroacoustics on Curvilinear Meshes. AIAA 200-2330, Reno 2000.

[8] M. R. Visbal and D. V. Gaitonde. Computation of Aeroacoustics on General Geometries Using Compact Differencing and Filtering Schemes. AIAA 99-3706, 30th AIAA Fluid Dynamics Conference, Norfolk, VA, 1999.

[9] R. Hixon. Prefactored Compact Filters for Computational Aeroacoustics. AIAA 99-0358. Reno 1999.

[10] R. Hixon, R. R. Mankbadi, and L. A. Povinelli. Very Large Eddy Simulation of Jet Noise. AIAA 2000-2008. 6th Aeroacoustics Conference, Lahaina, Hawaii, 2000.

[11] D. Gaitonde and M.R. Visbal, High-Order Schemes for Navier-Stokes Equations: Algorithm and Implementation into FDL3DI. Technical Report # AFRL-VA-WP-TR-1998-3060, Air Force Research Laboratory, Wright-Patterson AFB, OH. (1998)

[12] F. Ladeinde, X. Cai, M. R. Visbal, and D. V. Gaitonde. Efficiency and Scalability Issues in The Parallel Implementation of Curvilinear High-Order Schemes. AIAA 2000-0276, Reno 2000.

[13] D. Gaitonde, J. S. Shang, and J. L. Young. Practical Aspects of High-Order Accurate Finite Volume Schemes for Electromagnetics. AIAA 97-0363, Reno 1997.

[14] M.R. Visbal and D.V. Gaitonde, High-Order Accurate Methods for Unsteady Vortical Flows on Curvilinear Meshes, Paper AIAA-98-0131, Reno, NV. (1998).

[15] T. H. Pulliam and J. L. Steger. Implicit Finite-Difference Simulation of Three Dimensional Compressible Flows. AIAA Journal 18 (2), pp. 159-167, February 1980

[16] D. J. Fyfe. Economical Evaluation of Runge-Kutta Formulae. Math. Comput. 20, pp. 392-398, 1966.

[17] D. Gaitonde and M.R. Visbal. 1999. Further Development of a Navier-Stokes Solution Procedure Based on Higher-Order Formulas, AIAA Paper 99-0557.

[18] Cai, X., Ladeinde, F. & O'Brien, E. E. 1997. DNS on SP2 With MPI. *Advances in DNS/LES.* Edit. C. Liu & Z. Liu. Greyden Publishing Co., Columbus Ohio., pp. 491-495.

[19] A. Povitsky. Parallel Directionally Split Solver Based on Reformulation of Pipelined Thomas Algorithm. ICASE Report No.98-45.

[20] X.-H. Sun and S. Moitra. A Fast Parallel Tridiagonal Algorithm for a Class CFD Applications. *NASA TP 3585*, 1996.

[21] M. R. Visbal and D. V. Gaitonde. Computation of Aeroacoustics on General Geometries Using Compact Differencing and Filtering Schemes. Accepted for Publication, J. Acoustics. 2001.

[22] Proceedings of the Second Computational Aeroacoustics Workshop on Benchmark Problems. NASA Langley Research Center, Hampton, VA, 1997

Table 1 (a): Theoretical CPU times required by various schemes to invert a tridiagonal system of matrix equations. "j" is the reduced number in Sun's PDD algorithm and "k" is the number of groups in Povitsky's PTA procedure.

Algorithm	Computation	Communication	Idle
TDMA[16]	$N_1(5N-3)\gamma$	0	0
One-Sided[16]	$N_1\left[5(\frac{N}{P}+N_2)-3\right]\gamma$	$(2\alpha+N_1N_2\beta)$	0
PTA[14,16]	$T_1=N_1\left(5\frac{N}{P}-3\right)\gamma$	$T_2=2k\alpha+2N_1\beta$	$(P-1)\left(T_1/k\right)$
PDD[13,16]	$N_1\left(5\frac{N}{P}+3j+1\right)\gamma$	$2\alpha+2N_1\beta$	0

Table 1 (b): Theoretical versus observed CPU times on IBM SP2 taken by various schemes to invert a tridiagonal system of matrix equations. Only the computation task is included in this table (i.e., no communication or idle time) and the numbers have been normalized by the CPU time for the Thomas algorithm. The numbers in parenthesis are the observed (measured) values on the IBM SP2.

Algorithm	$P=2$	$P=4$	$P=8$	$P=16$
TDMA	1	1	1	1
One-Sided	1.954 (1.957)	3.826 (3.802)	7.342 (7.30)	13.585 (13.41)
PTA	1.53 (1.445)	2.29 (1.927)	3.07 (2.179)	3.69 (2.169)
PDD	1.89 (1.762)	3.59 (3.025)	6.52 (5.098)	11.01 (6.78)

Table 2: Performance data for sequential and one-sided parallel scheme calculation of laminar boundary layer. For parallel processing, the size of the grid in each processor is $60\times30\times30$. For the sequential calculations, the grid points in (x,y,z) corresponding to each of the 8 parallel cases are $(60,30,30)$, $(120,30,30)$, $(60,60,30)$, $(240,30,30)$, (120,60,30), (120,120,30), (120,60,60), and (120,120,60). The abbreviations "Proc", "dim" and "seq" in the table denote "processor", "dimension", and "sequential", respectively

# Proc.	1	2	2	4	4	8	8	16
Proc. dim.	(1,1,1)	(2,1,1)	(1,2,1)	(4,1,1)	(2,2,1)	(2,4,1)	(2,2,2)	(2,4,2)
Parallel CPU	41.92	43.82	44.85	45.53	48.43	50.29	53.03	54.37
Sequential Grid	54000	108000	108000	216000	216000	432000	432000	884000
Sequential CPU	42.79	93.82	93.33	192.64	198.04	410.07	419.92	862.86

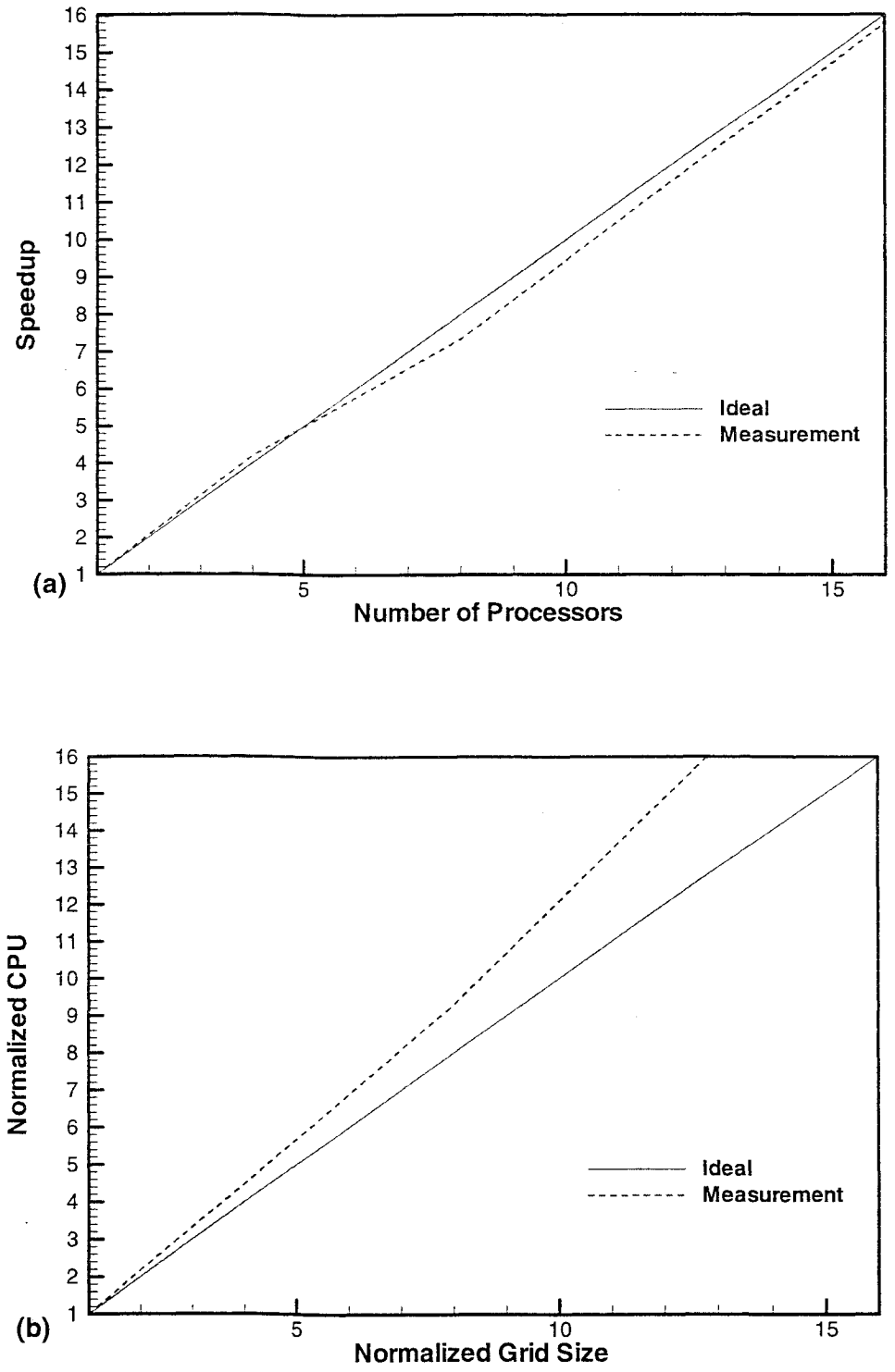


Figure 1. Performance of parallel computations using the one-sided strategy.

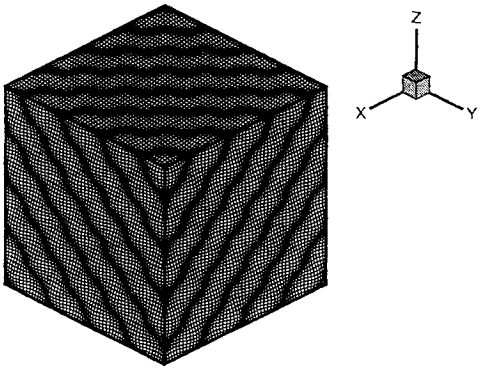


Figure 2: 3D curvilinear mesh model for spherical acoustic pulse.

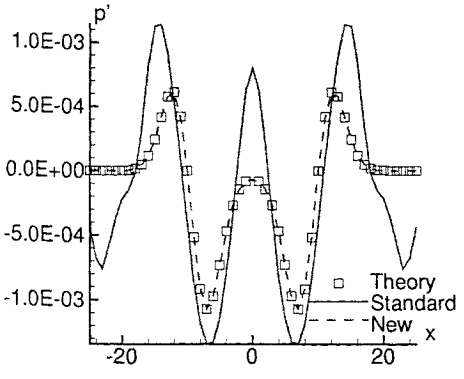


Figure 3: Effect of metric evaluation on computed pressure along line through spherical pulse at t=10.

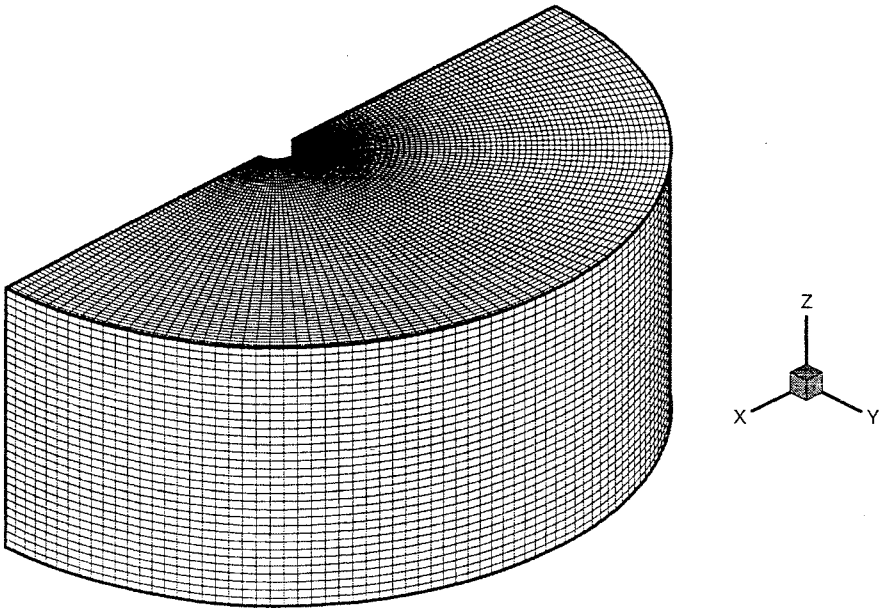


Figure 4. Coarse version of the 3D mesh used for parallel computing scattering of acoustic pulse from a cylinder.

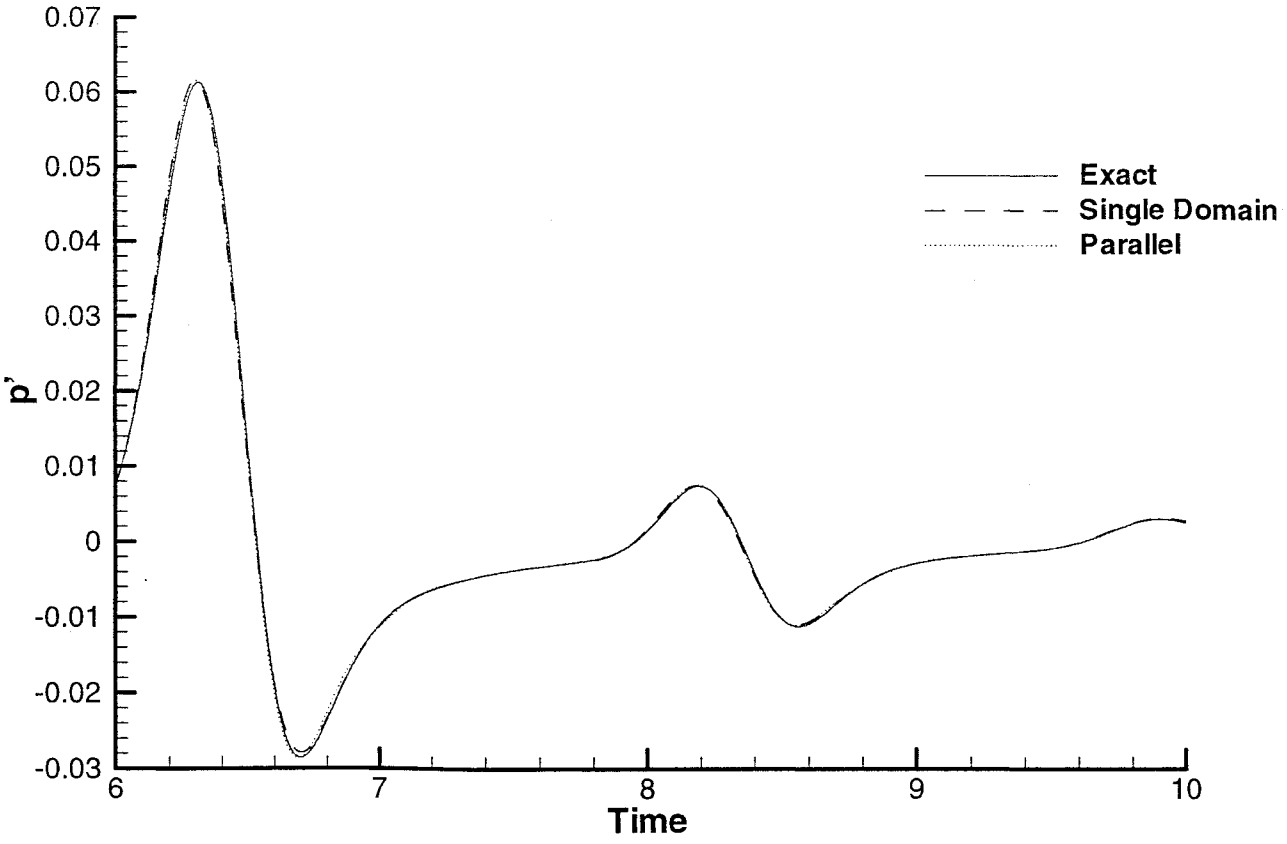


Figure 5. History of pressures at Point (0.,5.,0.) from 2D single-domain and 3D parallel computations.

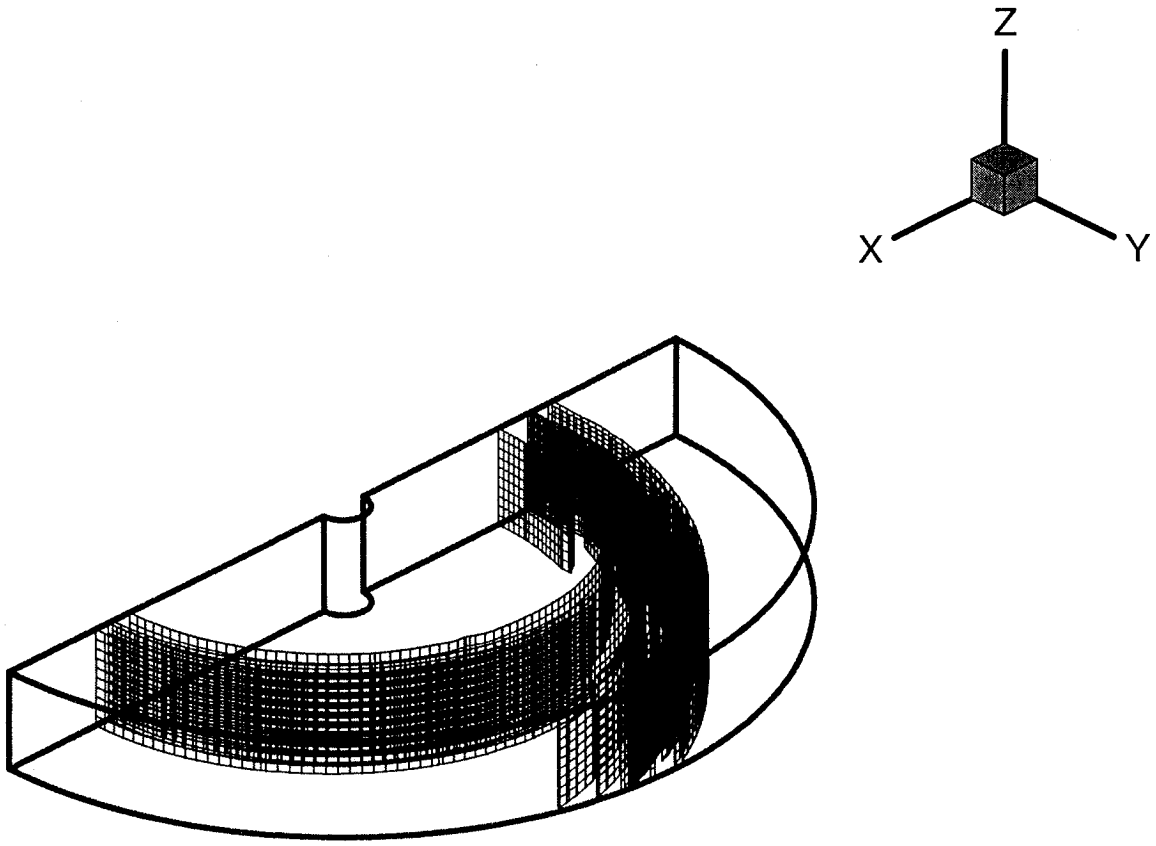
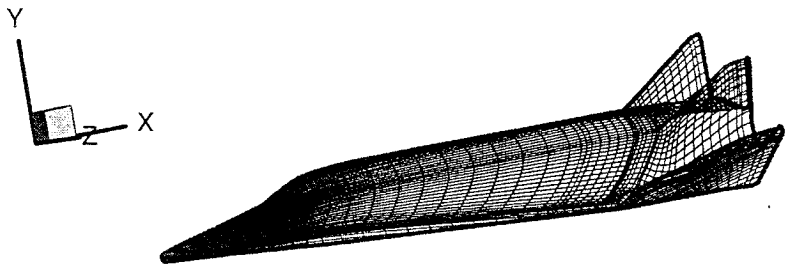
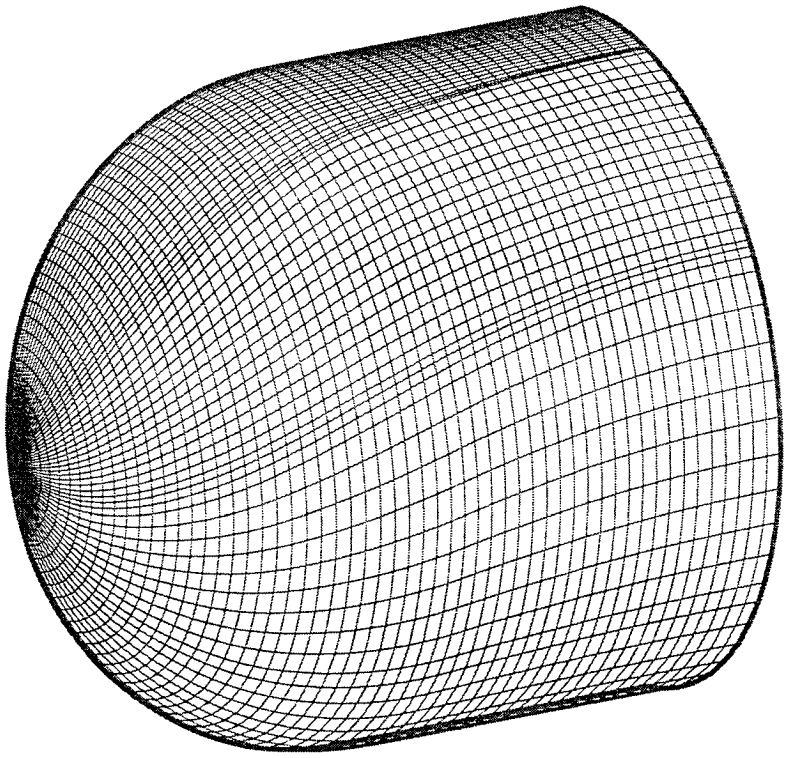


Figure 6. 3D contour maps of pressure distribution for acoustic scattering from a cylinder.



Inner Surface

Figure 7. Coarse grid model of X24C Reentry Vehicle showing the surface mesh (J=0).



Outer Surface

Figure 8. Coarse grid model of X24C Reentry Vehicle showing the outer surface mesh (J=40).

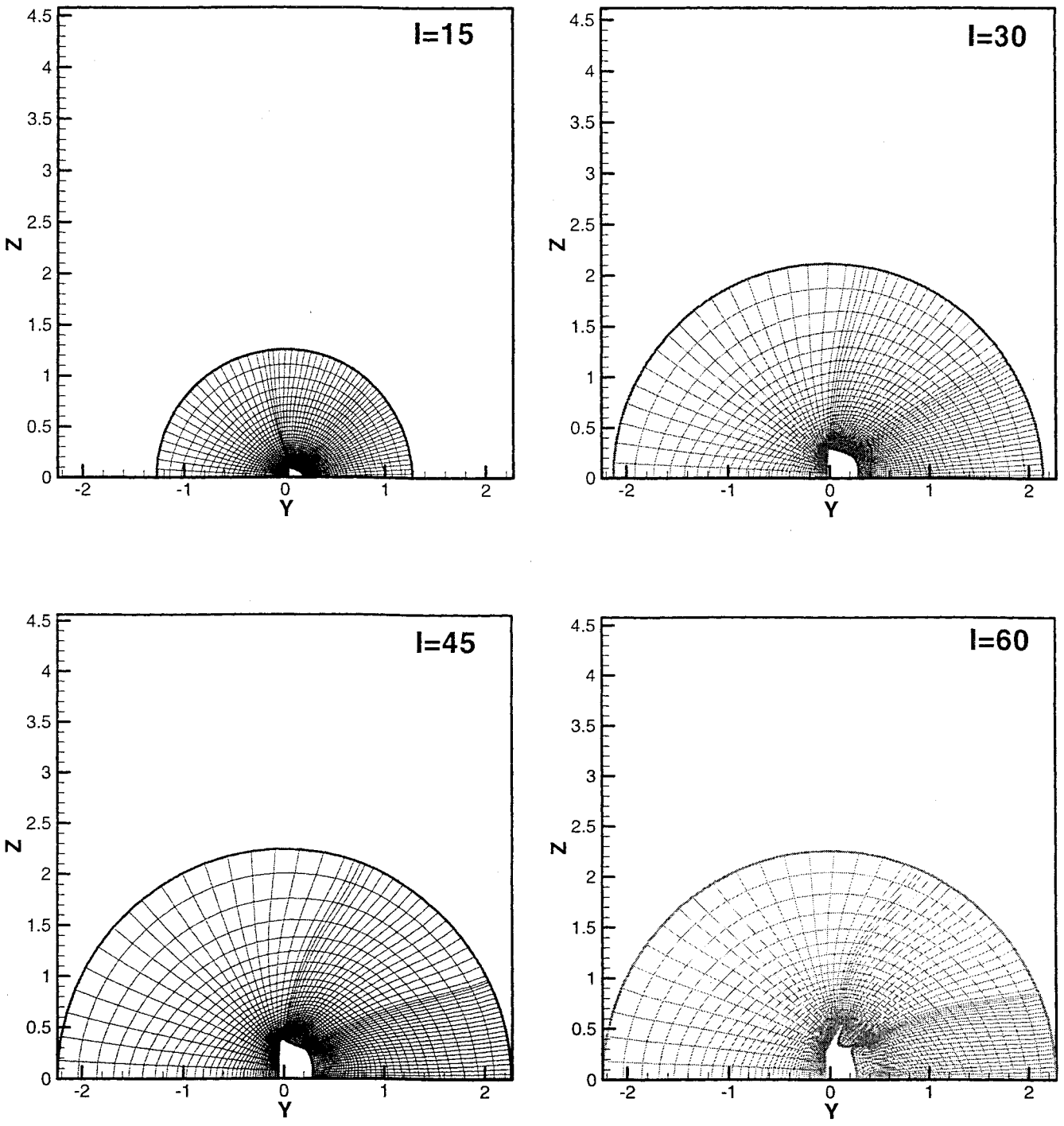


Figure 9. Cuts along the x-direction of the model X24C Reentry Vehicle. Projected views are shown here.

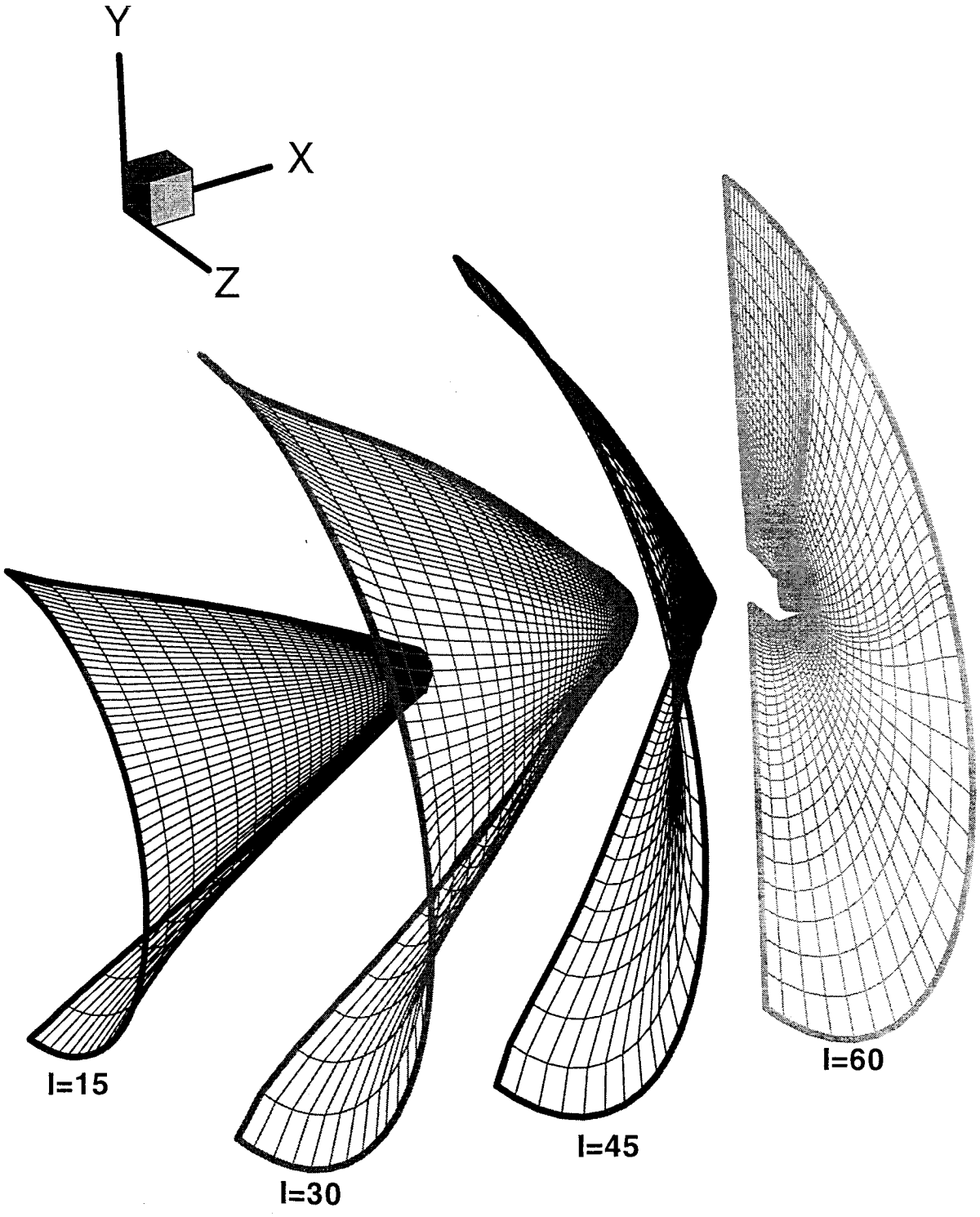


Figure 10. Cuts along the x-direction of the model X24 Reentry Vehicle. The actual surfaces are shown.

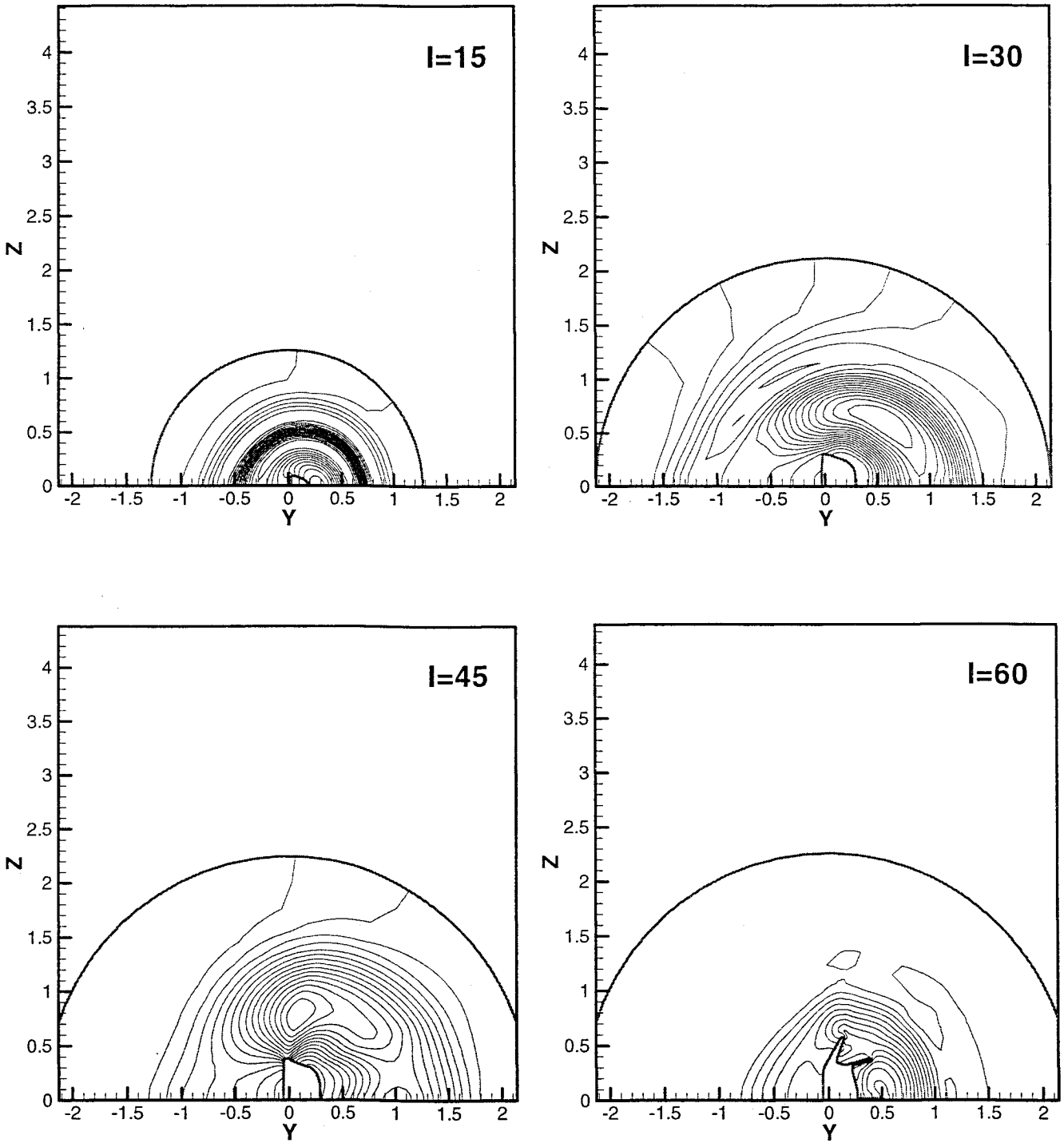


Figure 11. Acoustic pressure distribution in various surfaces along the X (axial) direction of the model X24C Reentry Vehicle.

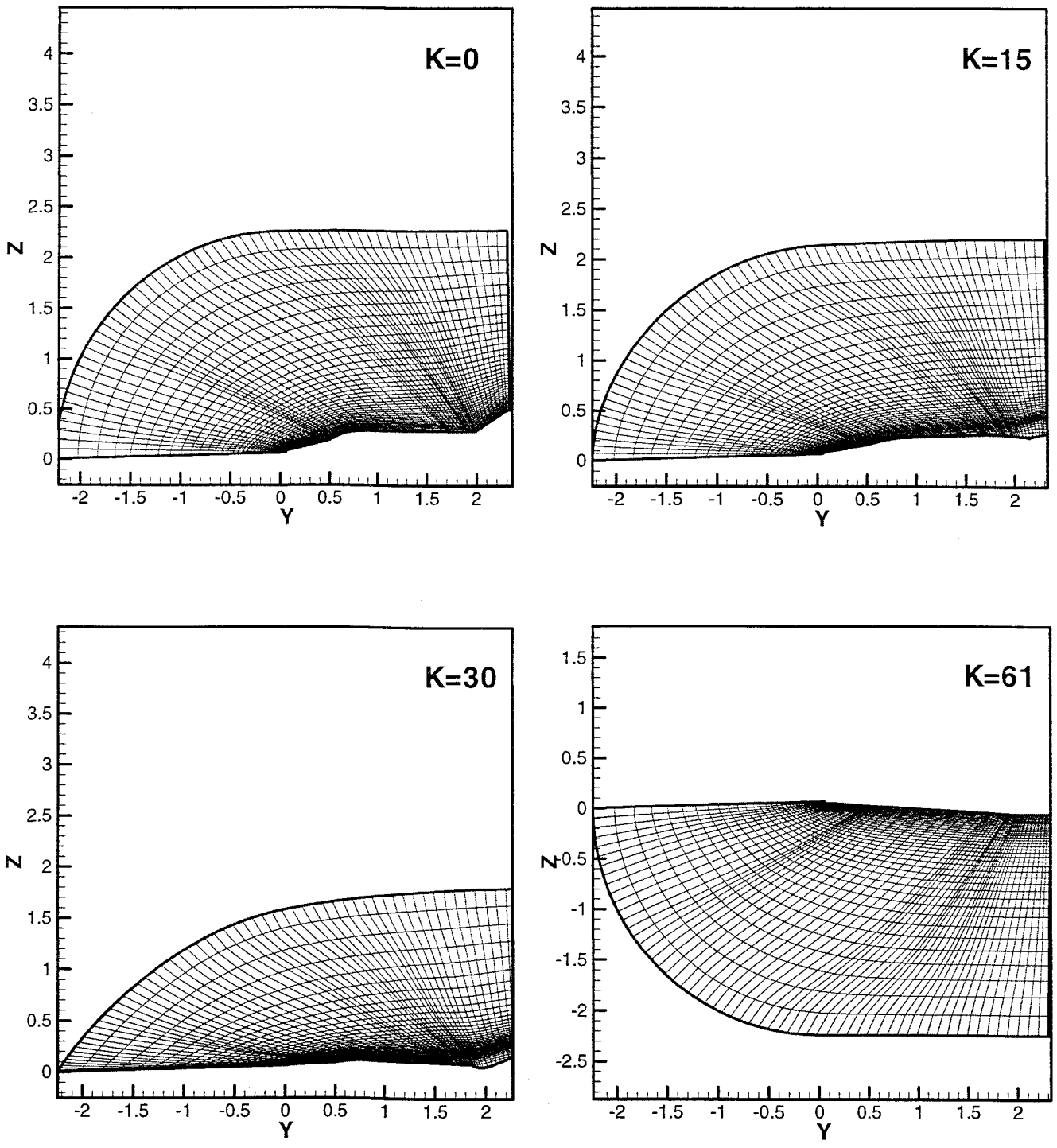


Figure 12. Cuts along the the Z (circumferential) direction of the model X24C Reentry Vehicle. Projected views are shown

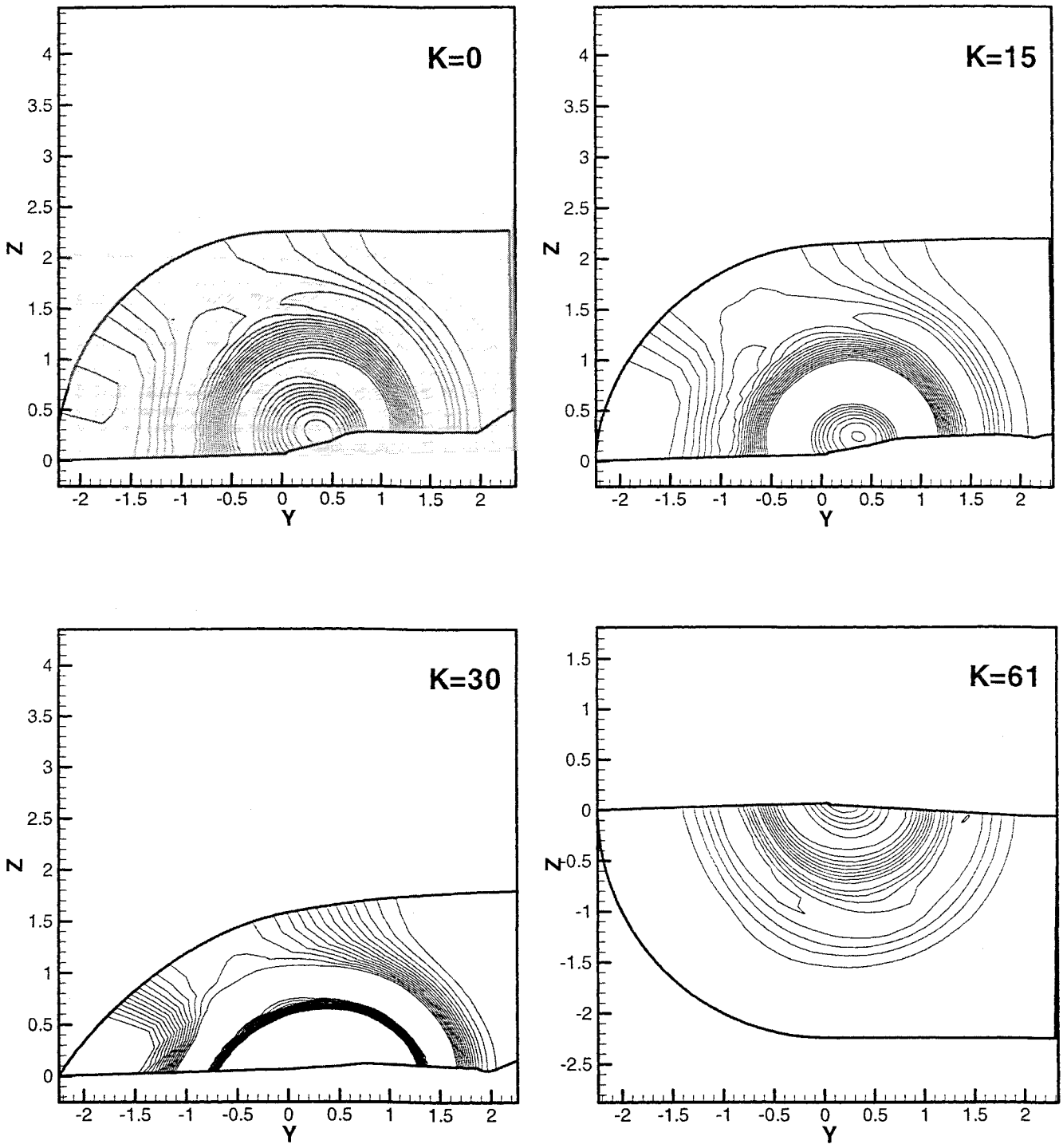


Figure 13. Acoustic pressure distribution in various surface along the Z (circumferential) direction of the model X24C Reentry Vehicle.