AIAA JOURNAL
Vol. 41, No. 10, October 2003

Verification and Validation for Laminar Hypersonic Flowfields, Part 1: Verification

Christopher J. Roy,* Mary A. McWherter-Payne,† and William L. Oberkampf‡
Sandia National Laboratories, Albuquerque, New Mexico 87185

Numerical simulations are performed for Mach 8 laminar flow of a calorically perfect gas over a spherically blunted cone. Code verification calculations are conducted to provide confidence that there are no coding mistakes and include comparisons to highly accurate inviscid benchmark solutions as well as code-to-code comparisons. Special attention is paid to the numerical accuracy of the solutions by carefully monitoring iterative convergence errors and by conducting an extensive grid convergence study. Nonmonotonic convergence of the surface pressure and drag are observed with mesh refinement. The source of this nonmonotonicity is explored in detail. The standard method for determining the spatial order of accuracy is shown to be inadequate for the numerical algorithm employed, and an alternative method is proposed. The overall discretization error of the fine grid surface pressure distributions is estimated to be below 0.4%, with the maximum errors found at the sphere–cone tangency point. With the accuracies demonstrated, it is recommended that the present computations can be used as a numerical benchmark solution for code verification.

Nomenclature

f	=	solution variable
g_i	=	ith-order error term coefficient
h	=	measure of the grid cell spacing
M	=	Mach number
N	=	total number of grid cells
Pr	=	Prandtl number, 0.71
p	=	pressure, N/m^2, or spatial order of accuracy
R_N	=	nose radius, 0.00508 m
r	=	grid refinement factor, h_{k+1}/h_k
T	=	temperature, K
t	=	time, s
W	=	molecular weight, 28.013 kg/kmol for N_2
x	=	axial coordinates, m
y	=	radial coordinate, m
α, β	=	constants
γ	=	ratio of specific heats
ε	=	iterative convergence error
ε_{21}	=	solution difference between mesh levels 2 and 1, $f_2 - f_1$
ε_{32}	=	solution difference between mesh levels 3 and 2, $f_3 - f_2$
Λ	=	iterative convergence parameter

Subscripts and Superscripts

exact	=	exact (or best estimate) value
k	=	mesh level
n	=	iteration number
RE	=	Richardson extrapolation

Received 29 July 2002; revision received 26 March 2003; accepted for publication 4 April 2003.

*Senior Member of Technical Staff, P.O. Box 5800; currently Assistant Professor, Department of Aerospace Engineering, Auburn University, 211 Aerospace Engineering Building, Auburn University, AL 36849-5338; cjroy@eng.auburn.edu. Senior Member AIAA.

†Principal Member of Technical Staff, Mail Stop 0825, P.O. Box 5800. Senior Member AIAA.

‡Distinguished Member of Technical Staff, Mail Stop 0828, P.O. Box 5800. Associate Fellow AIAA.

Introduction

WITH advances in computing power, engineers increasingly rely on modeling and simulation for the design, analysis, and certification of engineering systems. Thus, there is a need to increase the confidence in these simulations, especially in high-risk areas such as aviation, nuclear power generation, and nuclear weapons systems. Verification and validation provides the primary means by which the overall accuracy of computational simulations can be assessed.

To develop a computational model, one must first define a conceptual model of the physical system. Verification is the process of both determining that a model implementation accurately represents the developer's conceptual description and assessing how accurately this conceptual model is solved.[1] Validation, as defined in Ref. 1, is "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model." Simply put, verification asks the mathematical question, Are we solving the equations right?, whereas validation asks the physical question, Are we solving the right equations?

Verification can be separated into two parts, code verification and solution verification. Code verification is used to find coding errors in the discrete solution to a given set of governing equations and boundary conditions. Code verification can be assessed by comparison to exact analytical solutions, the method of manufactured solutions,[2,3] comparison to highly accurate numerical benchmark solutions, and code-to-code comparisons. The first two approaches are rigorous code verification procedures, especially when the order of accuracy of the numerical method is verified. The last two approaches are less rigorous and can be classified as confidence-building approaches, where other tests such as quantifying the error (or difference) may be used for code-to-code comparisons. In this paper, code verification is addressed through a comparison to highly accurate numerical benchmark results for inviscid flow and through code-to-code comparisons.

Solution verification (or numerical error assessment) is concerned with quantifying the numerical error of a given simulation and should ideally take place after code verification has been completed. Solution verification should be performed for each application of the code that is significantly different than previous applications. For steady-state problems, the two main aspects of solution verification are iterative convergence and grid convergence. The former deals with the marching of a solution in pseudotime toward a steady state, whereas the latter addresses the adequacy of the mesh on which the discrete equations are being solved. The spatial order of accuracy

is also an important metric for assessing the errors due to spatial resolution. This paper places a strong emphasis on solution verification. In particular, issues dealing with numerical schemes that have mixed-order spatial accuracy will be addressed.

The goal of the current study is to assess the numerical accuracy of axisymmetric simulations for Mach 8 flow past a spherically blunted cone, the configuration studied under the Joint Computational/Experimental Aerodynamics Program (JCEAP) at Sandia National Laboratories. The Reynolds number is sufficiently low so that the flow remains laminar, and the flow of a calorically perfect gas is assumed. In the companion study,[4] validation comparisons are made between the numerical solutions and the experimental data for surface pressure reported in Refs. 5 and 6. However, before model validation can take place, the numerical accuracy of the simulations must be quantified.

The remainder of this paper is organized as follows: A brief description is given of the computational tool and the flowfield conditions. Next, code verification studies are presented and include comparisons to highly accurate inviscid benchmark solutions as well as code-to-code comparisons. A comprehensive analysis of the numerical accuracy of the simulations is then presented, including a discussion of the iterative and spatial errors and the nonmonotonic solution behavior as the grid is refined. Finally, error estimates are given for the surface pressure distributions using the mixed-order method.

Flowfield Model

The computational fluid dynamics code used herein is SACCARA, the Sandia advanced code for compressible aerothermodynamics research and analysis. The SACCARA code was developed from a parallel distributed memory version[7,8] of the INCA code,[9] originally written at Amtec Engineering. The SACCARA code is used to solve the Navier–Stokes equations for conservation of mass, momentum, and energy in axisymmetric form. Prior code verification studies with SACCARA include code-to-code comparisons with other Navier–Stokes codes (see Refs. 10 and 11) and with the direct simulation Monte Carlo method.[12] The governing equations are discretized using a cell-centered finite volume approach. The convective fluxes at the interface are calculated using the Steger–Warming[13] flux vector splitting scheme. Second-order reconstructions of the interface fluxes are obtained via MUSCL extrapolation.[14] The viscous terms are discretized using central differences. A flux limiter is employed that reduces the spatial discretization to first order in regions of large second derivatives of pressure and temperature. This limiting is used to prevent oscillations in the flow properties at shock waves. The use of flux limiting results in a mixture of first- and second-order accuracy in space. The ramifications of the mixed-order scheme on the grid convergence behavior will be discussed in detail.

The SACCARA code employs a massively parallel distributed memory architecture based on multiblock structured grids. The point-implicit solver is a lower–upper symmetric Gauss–Seidel scheme based on the works of Yoon and Jameson,[15] Yoon and Kwak,[16] and Peery and Imlay,[17] which provides for scalability up to thousands of processors.[18] The simulations presented herein were run using a single 400-MHz processor of a Sun Enterprise 10000 shared-memory machine. The only exception was the finest mesh level (960×960 cells), which was domain decomposed and run in parallel on 50 processors of the Accelerated Strategic Computing Initiative Red Teraflop machine.

Flowfield Conditions

The problem of interest is the Mach 8 perfect gas flow of nitrogen ($\gamma = 1.4$) over a spherically blunted cone. The cone half-angle is 10 deg, and the model has a total length of 0.2639 m and a nose radius of 0.00508 m. A sample flowfield mesh is shown in Fig. 1 along with an enlargement of the spherical nose region. The freestream boundary conditions given in Table 1 are applied at the outer boundary, symmetry is applied at $y = 0$, and a no-slip boundary condition is employed at the vehicle surface with a constant wall temperature of 316.7 K as recommended in Ref. 19. These conditions correspond to the validation experiments conducted by Oberkampf et al.,[5,6] which include surface pressure data.

Table 1 Flowfield conditions for sphere–cone geometry

Flow parameter	Value
Freestream Mach number	7.841
Stagnation pressure	2.4724×10^6 N/m^2
Stagnation temperature	632.8 K
Freestream static pressure	286.8 N/m^2
Freestream static temperature	47.7 K
Freestream unit Reynolds number	6.88×10^6/m
Wall temperature	316.7 K

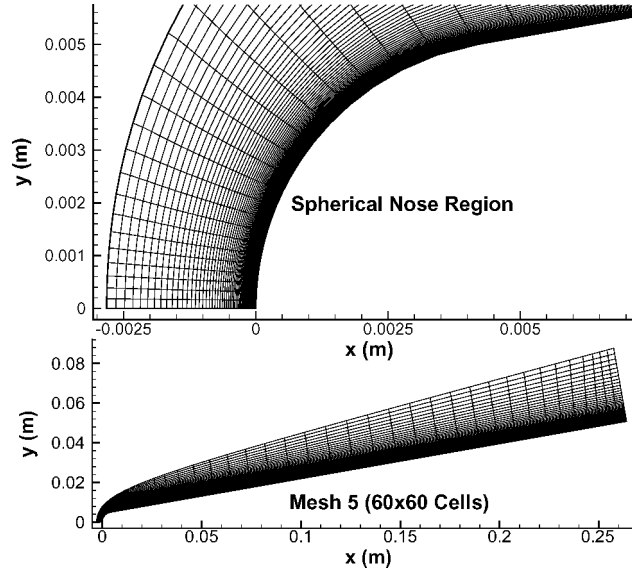

Fig. 1 Sample flowfield mesh for axisymmetric calculations.

The flow in the base region is not computed due to the large computational expense; thus, a supersonic outflow boundary condition, that is, zero gradient extrapolation, is applied at the outflow plane. The grid lines that intersect with the body are normal to the surface and employ hyperbolic tangent functions to ensure smooth variations in grid spacing for adjacent cells. For the finest mesh, the height of the first cell at the wall is approximately 5.0×10^{-7} m at the stagnation point and 3.0×10^{-6} m at the aft corner. The spacing along the surface varies from 1.0×10^{-5} m near the stagnation point to 2.5×10^{-4} m at the end of the body. The coarser meshes, mesh 2 (480×480 cells)–mesh 7 (15×15 cells), are specified by successively eliminating every other grid line, that is, grid halving. All calculations used the freestream properties as initial conditions over the entire domain.

Code Verification

Benchmark Inviscid Solutions

Inviscid solutions with the SACCARA code were computed using the grids already discussed and then compared with two highly accurate numerical benchmark results. The first set of benchmark results are for the inviscid flow of a perfect gas ($\gamma = 1.4$) over a sphere and were provided by Carpenter of NASA Langley Research Center (private communication, June 2000). These calculations employed a high-order, shock-fitting Chebyshev collocation spectral method to achieve accuracies on the order of eight significant figures. (See Refs. 20 and 21 for more details.) The second set of inviscid benchmark calculations are for the Mach 8 perfect gas flow ($\gamma = 1.4$) over a 10-deg half-angle spherically blunted cone. These calculations used shock-fitting finite-difference methods and employed temporal marching in the subsonic region and a space marching procedure in the supersonic region.[22,23]

Surface pressure distributions over the spherical nose tip are given in Fig. 2 for the SACCARA code, the Chebyshev collocation

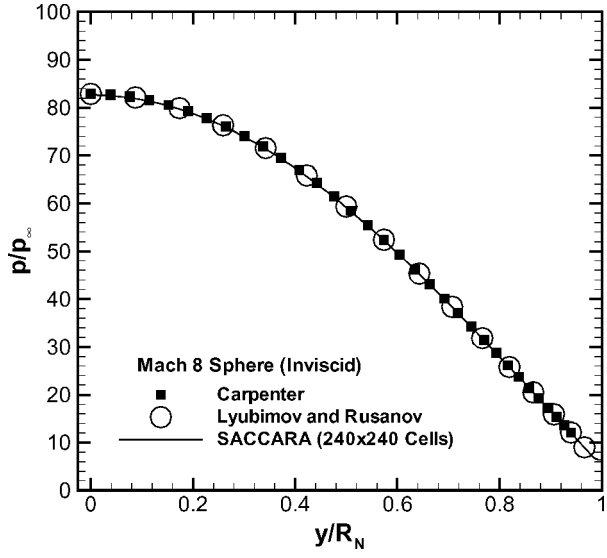

Fig. 2 Comparison of SACCARA pressure distribution with benchmark inviscid solutions (Ref. 23; Carpenter, private communication, June 2000) on the spherical nose tip.

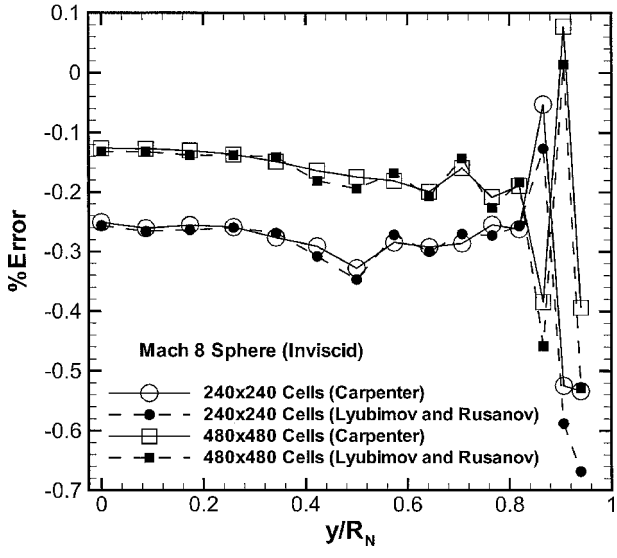

Fig. 3 Error in SACCARA pressure distribution relative to the benchmark inviscid solutions (Ref. 23; Carpenter, private communication, June 2000) on the spherical nose tip.

benchmark solution, and the finite difference benchmark solution. Good agreement is seen between the three methods. The error in the inviscid SACCARA solution with respect to each of the benchmark solutions is shown in Fig. 3 for two mesh levels, 240×240 and 480×480 cells. This error is defined by

$$\text{error}\,(\%) = [(f - f_{\text{benchmark}})/f_{\text{benchmark}}] \times 100 \qquad (1)$$

where f refers to a SACCARA solution and $f_{\text{benchmark}}$ refers to one of the two benchmark solutions. These two benchmark numerical solutions for surface pressure are in excellent agreement on the spherical nose but show some minor differences near the sphere–cone tangency point. The effects of grid refinement are to reduce the error by a factor of two on the finer mesh, indicating first-order spatial accuracy, which is lower than the nominal accuracy of the scheme (second order). This reduction in the observed order of accuracy will be explored in detail in the "Solution Verification" section. The increase in error near $y/R_N = 0.9$ is due to the lack of axial clustering at the sphere–cone tangency point in the SACCARA solution. Whereas the surface location and slope are continuous in this region, the curvature is discontinuous and, as a result, requires further axial clustering to achieve small errors. The magnitude of the errors in surface pressure away from this region are approximately 0.3 and 0.15% for the 240×240 and 480×480 cell meshes, respectively.

Table 2 Comparison of SACCARA results to Lyubimov and Rusanov[23] normalized surface pressures (p/p_∞) for a 10-deg sphere–cone

x/R_N	120×120 cells	240×240 cells	480×480 cells	Lyubimov and Rusanov[23]
1	6.248	6.293	6.279	6.258
2	4.959	4.974	4.990	4.982
3	4.133	4.146	4.155	4.156
4	3.645	3.656	3.667	3.663
5	3.366	3.378	3.380	3.376
10	3.030	3.032	3.035	3.050
15	3.215	3.216	3.215	3.242
20	3.466	3.468	3.468	3.496
25	3.679	3.680	3.680	3.705
30	3.830	3.830	3.830	3.850
40	3.998	3.999	3.999	4.014
50	4.049	4.052	4.053	4.060

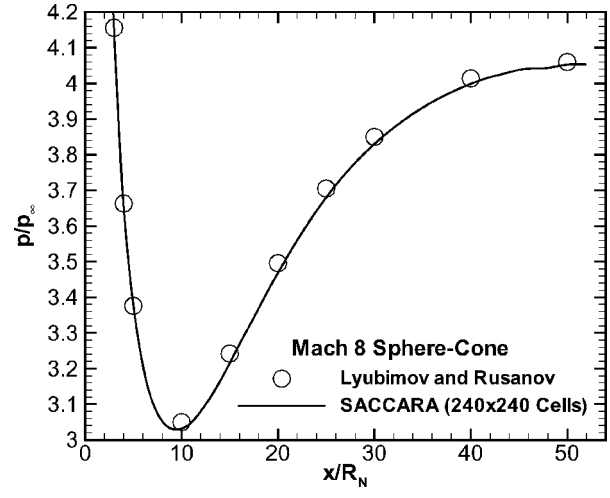

Fig. 4 Comparison of SACCARA pressure distribution with benchmark inviscid solution[23] on the conical afterbody.

The finite difference benchmark solution of Lyubimov and Rusanov also provides flow properties on the conical portion of the body, although the estimated error in the benchmark solution on the cone are larger than those on the sphere.[22,23] A comparison of the inviscid SACCARA solution for surface pressure on the cone is shown in Fig. 4, with good qualitative agreement seen. An expanded scale is used on the y axis to emphasize differences in the solutions. The pressure values for three grid levels using the SACCARA code are given in Table 2 along with the data from the finite difference benchmark solution. In this case, the SACCARA solutions do not appear to converge to the benchmark solution. Possible sources of error in the benchmark calculations include interpolation error between the time-marching scheme (subsonic region) and the space-marching scheme (supersonic region) and the increase in error as the space-marching method proceeds downstream. Both of these error sources are discussed in detail in Refs. 22 and 23. Note that only four significant figures are provided for this benchmark solution, with errors estimated to be as large as ± 0.001 by the authors.

Navier–Stokes Code-to-Code Comparisons

Comparisons have been made using the thin-layer Navier–Stokes code NSEQ[24,25] and the parabolized Navier–Stokes code SPRINT[26,27] to increase the confidence in the laminar SACCARA simulations. Both of these codes have been used extensively at Sandia National Laboratories for the analysis of hypersonic flows. The SPRINT code uses NSEQ to provide initial conditions for the

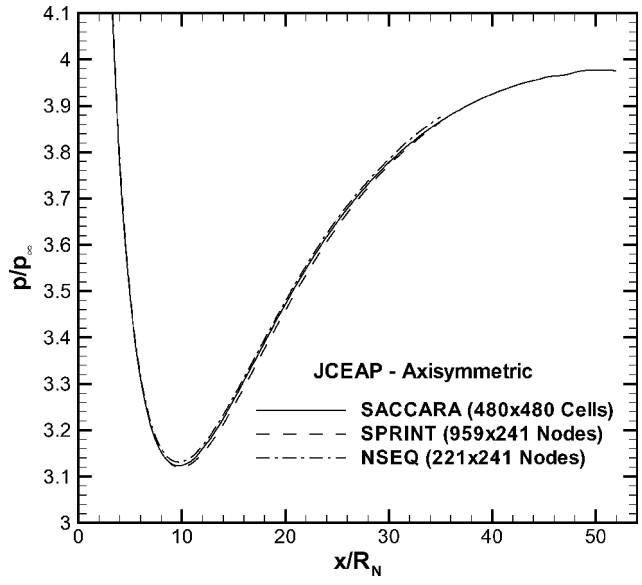

Fig. 5 Code-to-code comparison of SACCARA surface pressure distributions with SPRINT and NSEQ.

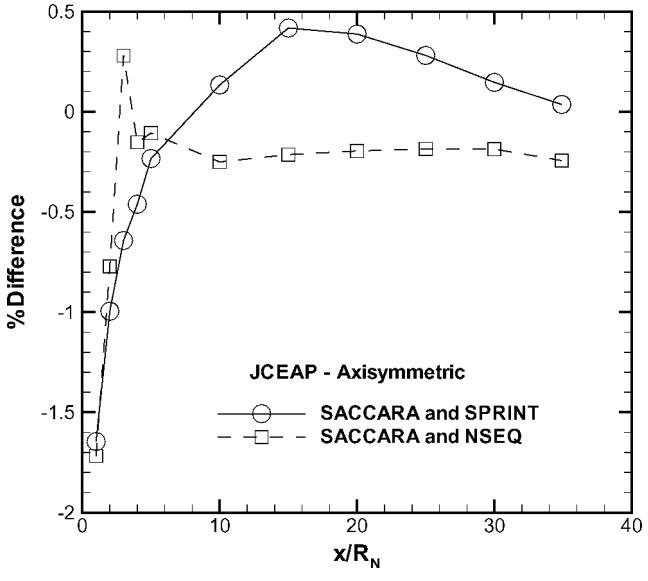

Fig. 6 Percent difference in surface pressure distributions between SACCARA (480 × 480 cells), SPRINT (959 × 241 nodes), and NSEQ (221 × 241 nodes).

space-marching procedure. These two formulations employ shock fitting and use a finite difference scheme, whereas the SACCARA code uses a finite volume shock-capturing approach.

Figure 5 shows results for surface pressure, with good agreement shown between the three codes. Quantitative differences between these three codes are given in Fig. 6. Again, the largest differences are seen near the sphere–cone tangency point ($x/R_N \approx 1$). Away from this region, the differences between the SACCARA code and NSEQ are fairly constant at approximately 0.25%, whereas the differences between SACCARA and SPRINT show more variation and peak near 0.4%. Consistent with the inviscid space-marching procedure discussed in the preceding section, the SPRINT solution also suffers from interpolation error during initialization and error accumulation at the downstream locations. The large number of points used in the axial direction for SPRINT are required because the space-marching procedure is nominally first-order accurate in the marching direction. The benchmark inviscid solutions, the code-to-code comparisons, and the previously published results[10,12] all provide a level of confidence that the SACCARA code is free from coding errors in the exercised portions of the code.

Solution Verification

Iterative Convergence Error

The standard method for assessing iterative convergence is to monitor the L_2 norms of the residuals for the governing equations over the entire domain. The residuals are formed by simply substituting the current numerical solution into the discretized form of the steady-state equations, which should approach zero as iterative convergence is achieved. A solution can be considered fully iteratively converged, within the precision of the computer used, when the residuals are reduced to machine zero (approximately 15 orders of magnitude for a double-precision computer). However, the practice of monitoring iterative convergence does not necessarily provide information on the iterative error in a given flowfield quantity. Furthermore, for engineering calculations, it is not always necessary, or even possible, to converge the solution to machine zero.

For the axisymmetric, laminar, perfect gas flowfield examined herein, the residuals for each of the governing equations were reduced by approximately 14 orders of magnitude on all grid levels. The reduction of the residuals to machine zero provides confidence that the iterative errors in the solution variables are small; however, it does not provide quantitative estimates of the iterative errors. To assess the actual iterative convergence errors in the surface pressure, the method developed in Ref. 28 is employed for the fine grid calculations. This method is repeated for completeness.

The accuracy of a given flowfield variable f relative to the steady-state value is determined by expressing the numerical solution at time t^n as

$$f(t^n) = f^n = f_{\text{exact}} + \varepsilon^n \tag{2}$$

The exact steady-state value is f_{exact} and the convergence error at time t^n is ε^n. The iterative convergence error of the SACCARA code has generally been observed to have an exponential decrease in time, which gives the following variation as the solution approaches a steady state:

$$\varepsilon^n = \alpha e^{-\beta t^n} \tag{3}$$

where α and β are constants. Equations (2) and (3) may be combined and rewritten as

$$\beta t^n = \ell n \alpha - \ell n \left(f^n - f_{\text{exact}}\right) \tag{4}$$

Equation (4) is evaluated at three time levels, $(n-1)$, n, and $(n+1)$, and the three relations are used to eliminate α and obtain

$$\beta(t^n - t^{n-1}) = \ell n \left[\frac{\left(f^{n-1} - f_{\text{exact}}\right)}{\left(f^n - f_{\text{exact}}\right)}\right]$$

$$\beta(t^{n+1} - t^n) = \ell n \left[\frac{\left(f^n - f_{\text{exact}}\right)}{\left(f^{n+1} - f_{\text{exact}}\right)}\right]$$

If the time increments are equal, then $(t^n - t^{n-1}) = (t^{n+1} - t^n)$ and the preceding equations become

$$\left(f^{n-1} - f_{\text{exact}}\right)\left(f^{n+1} - f_{\text{exact}}\right) = \left(f^n - f_{\text{exact}}\right)^2$$

The exact steady-state value is solved for in the preceding equation, which gives

$$f_{\text{exact}} = \frac{f^n - \Lambda^n f^{n-1}}{1 - \Lambda^n}, \qquad \text{where} \qquad \Lambda^n = \frac{(f^{n+1} - f^n)}{(f^n - f^{n-1})} \tag{5}$$

The iterative convergence error becomes

$$\varepsilon^n = -\frac{(f^{n+1} - f^n)}{(1 - \Lambda^n)}$$

and the percent convergence error relative to the exact steady-state value becomes

$$\%\ \text{error of}\ f^n = -100\left[\frac{f^{n+1} - f^n}{f^n - \Lambda^n f^{n-1}}\right] \tag{6}$$

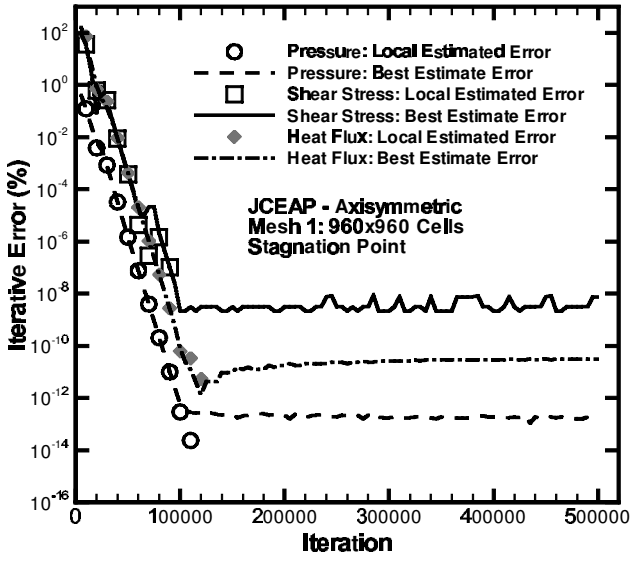

Fig. 7 Iterative convergence error for the 960 × 960 cell parallel SACCARA simulation: stagnation point.

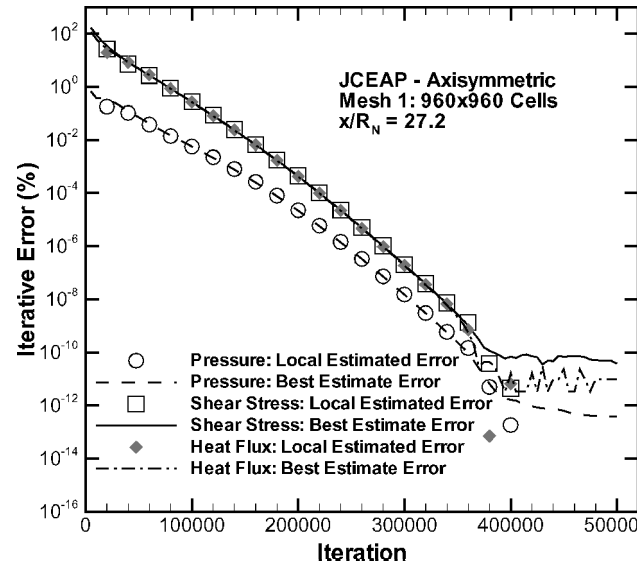

Fig. 8 Iterative convergence error for the 960 × 960 cell parallel SACCARA simulation: $x/R_N = 27.2$.

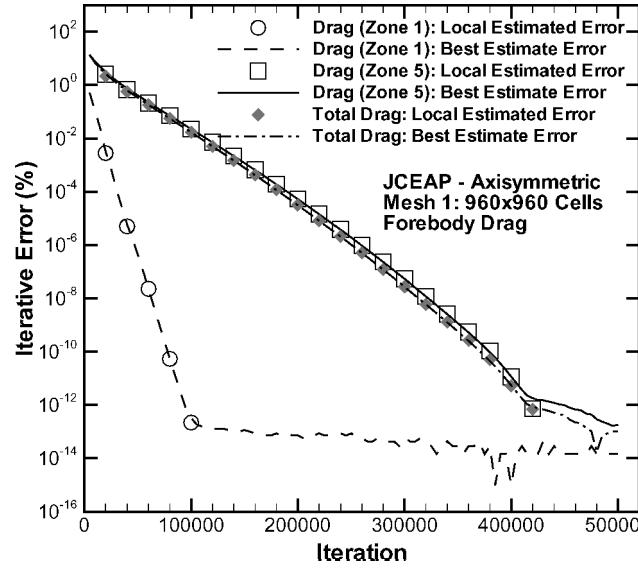

Fig. 9 Iterative convergence error of forebody drag for the 960 × 960 cell parallel SACCARA simulation.

Similar results were independently developed by Ferziger and Peric[29,30] for determining the convergence error of the numerical iterative solution of difference equations, but their results have been obtained with a different approach. In their work, the parameter Λ^n is the spectral radius (or the magnitude of the largest eigenvalue) of the iteration matrix. If the eigenvalues are complex, then the present approach is not appropriate. The complex eigenvalue case has been considered by Ferziger and Peric in Ref. 30.

The local iterative error estimates for various surface quantities at the stagnation point are presented in Fig. 7 along with the best estimates. The local error estimates obtained from Eq. (6) based on time levels $(n-1)$, n, and $(n+1)$ are indicated by the symbols. The curves in Fig. 7 represent the percent error obtained from the best estimate of the exact solution given by Eq. (5). This best estimate is determined from the final three iteration levels of the solution before machine zero is reached. These estimates indicate that the surface pressure, shear stress, and heat flux at the stagnation point all converge to machine zero at roughly 100,000 iterations. The pressure and heat flux exhibit approximately a 13 order of magnitude drop from the initial error values, whereas the surface shear stress achieves only an 11 order of magnitude drop. The limited convergence of the stagnation point shear stress is possibly because the exact value at this location is zero and, thus, is more susceptible to roundoff errors. Whereas the required number of iterations is high, the diagonal point-implicit scheme, which has excellent parallel scalability, requires essentially the same computational cost per iteration as an explicit scheme.

The iterative convergence behavior at a location halfway down the body $(x/R_N = 27.2)$ is presented in Fig. 8. The surface properties converge to machine zero by approximately 400,000 iterations. At this location, all of the surface properties exhibit a 13 order of magnitude drop in error. The larger number of iterations for this downstream location is indicative of the hyperbolic nature of the problem, where iterative convergence errors in the upstream regions essentially serve as varying boundary conditions for the downstream locations.

The iterative convergence behavior for the forebody drag (excluding the base region, which was not simulated) is shown in Fig. 9. The total forebody drag converges in a manner similar to the downstream surface pressure, which is not surprising because the pressure drag makes up more than 90% of the total drag. Also shown in Fig. 9 is the convergence of the drag contributions coming from the nose region (zone 1) and the aft region (zone 5). Zones 1–5 are each made up of 192 axial and 96 radial cells adjacent to the surface and arise from the parallel-domain decomposition (used for the finest mesh only). The slower convergence of the aft region is again indicative of the hyperbolic nature of the flow. The iterative errors in the surface pressure are below $10^{-12}\%$ and are much smaller than the spatial errors, as will be demonstrated in the next section.

Grid Convergence Error

Richardson Extrapolation

The Richardson extrapolation procedure is a technique by which two discrete solutions on different grid levels are used to obtain a solution extrapolated to zero mesh size. These Richardson extrapolated values can be used as a more accurate solution, or more important, as an approximation to the exact continuum solution, which can then be used to obtain error estimates in the discrete solutions. See Chap. 5 of Ref. 3 for a thorough discussion of the basic Richardson extrapolation technique. The underlying assumption in the Richardson extrapolation procedure is that the discrete solutions f on mesh level k have a series representation in powers of the mesh size, that is,

$$f_k = f_{\text{exact}} + g_1 h_k + g_2 h_k^2 + \mathcal{O}\left(h_k^3\right) \tag{7}$$

In Eq. (7), f_{exact} is the exact continuum solution, g_1 and g_2 are coefficients for the first- and second-order terms, respectively, and h_k is some measure of the grid spacing on mesh level k. The required conditions for applying general Richardson-type extrapolations are

that the observed order of the scheme is known and that the grids are sufficiently refined to be in the asymptotic grid convergence range, that is, the higher-order terms in Eq. (7) are small.

For a numerical method with second-order spatial accuracy, the coefficient g_1 is zero. If two second-order solutions are available, one on mesh 1 (fine mesh) and one on mesh 2 (coarse mesh), then Eq. (7) can be solved using these two discrete solutions to obtain a third-order (fourth-order if central differences are employed) accurate estimate of f_{exact}, that is,

$$f_{\text{exact}} \cong f_{\text{RE}} = f_1 + \frac{f_1 - f_2}{r^2 - 1} \tag{8}$$

where f_{RE} is the Richardson extrapolated estimate. Furthermore, if a grid refinement factor of two is used, that is, grid halving/doubling, then Eq. (8) reduces to the standard second-order Richardson extrapolation expression:

$$f_{\text{RE}} = f_1 + (f_1 - f_2)/3 \tag{9}$$

Again, the assumptions that go into using Eq. (9) are that the scheme is second-order accurate, the solutions are in the grid asymptotic range, and the grid refinement factor is two.

Solutions were obtained for seven grid refinement levels, from mesh 1 (960×960 cells) to mesh 7 (15×15 cells), with each successive grid level determined by eliminating every other grid line in each coordinate direction, that is, grid halving. The standard second-order Richardson extrapolation method was then applied using the two finest mesh levels to obtain a nominally third-order accurate estimate of the exact solution. Figure 10 shows the normalized surface pressures for the seven grid levels along with the Richardson extrapolation results. Differences between the finer grid solutions and the extrapolated results are not discernible from Fig. 10. The limits of the pressure axis have been chosen to highlight the differences in the solutions. In general, the Richardson extrapolation values can be used to obtain error estimates on the various grid levels; however, the assumptions of second-order accuracy and asymptotic grid convergence must first be verified.

Order of Accuracy

An additional solution can be used to verify the spatial order of accuracy of the numerical scheme. The standard method[3,31] for determining the order of accuracy is to assume that there is a single dominant error term of order p, that is,

$$f_k = f_{\text{exact}} + g_p h_k^p + \mathcal{O}\left(h_k^{p+1}\right) \tag{10}$$

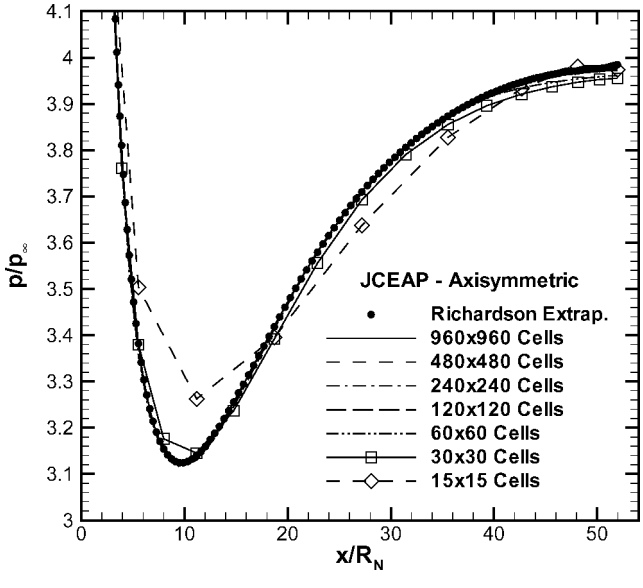

Fig. 10 Surface pressure distributions for the 10-deg half-angle sphere-cone simulations using seven mesh levels.

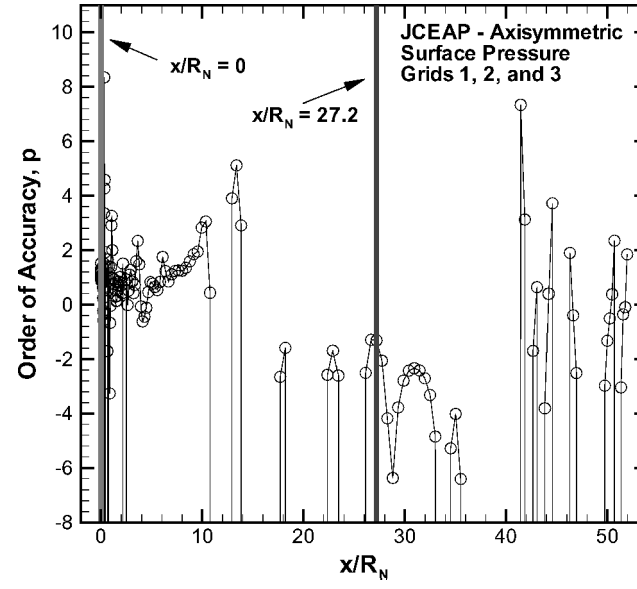

Fig. 11 Order of accuracy of the surface pressure solutions using the three finest meshes.

When three discrete solutions with a constant grid refinement factor ($r = h_2/h_1 = h_3/h_2$) are used, the system of equations found from Eq. (10) can be solved for the order of accuracy to give the following relationship:

$$p = \frac{\ln(\varepsilon_{32}/\varepsilon_{21})}{\ln(r)} \tag{11}$$

where ε_{21} and ε_{32} represent differences between the discrete solutions:

$$\varepsilon_{21} = f_2 - f_1, \qquad \varepsilon_{32} = f_3 - f_2 \tag{12}$$

The order of accuracy as determined by Eq. (11) is presented in Fig. 11 for the surface pressure distributions using the three finest mesh levels. The solution accuracy varies locally from negative values to values as large as eight. Assessment of the order of accuracy from these types of plots is not possible; furthermore, Eq. (11) is undefined when the three pressure solutions are nonmonotone, that is, local maxima or minima exist in the surface pressure vs grid spacing curve. Also shown in Fig. 11 are two locations that will be used for additional analysis ($x/R_N = 0$ and $x/R_N = 27.2$). The results shown in Fig. 11 imply that one of the assumptions used in developing Eq. (11) is not valid.

One assumption to examine is the assumption of a constant grid refinement factor. A stretched, curvilinear mesh was used, so that the mesh spacing is not uniform in physical space; however, the governing equations are transformed into a computational space with fixed spacing. The assumption of a constant grid refinement factor should, thus, be valid. The transformation itself can introduce error into the discrete solutions, but this error is expected to be very small because smooth hyperbolic tangent grid point distributions were used for clustering near to the surface. In addition, the maximum ratio of adjacent cell sizes (cell stretching factors) for the finest mesh were 1.007 normal to the body and 1.03 along the body.

The remaining two assumptions are that there is a single dominant error term and that the discrete solutions are in the asymptotic grid convergence range. These two assumptions are related because in the true asymptotic grid convergence range, that is, as $h \to 0$, the lowest-order error term that is nonzero will be the dominant error term. The Steger–Warming[13] upwind scheme used in the simulations employs a flux limiter that reduces the spatial accuracy of the scheme from second-order to first-order in regions of large second derivatives. For the finer meshes, the flux limiter is expected to be activated only at the shock wave. Thus, although the numerical scheme is nominally second-order accurate, there are regions in the domain

where the scheme will be first-order accurate, resulting in a mixed first- and second-order scheme. The order of accuracy calculation from Eq. (11) (and shown in Fig. 11) is not appropriate when the first- and second-order error terms are of the same magnitude.

Further insight into the described behavior can be gained by assuming that both first- and second-order error terms are present.[11,32] The series representation for the discrete solution from Eq. (7) is again assumed; however, now both the first-order (g_1) and second-order (g_2) terms will be retained. Three solutions are required and take the following form:

$$f_1 = f_{\text{exact}} + g_1 h_1 - g_2 h_1^2 + \mathcal{O}\left(h_1^3\right)$$

$$f_2 = f_{\text{exact}} + g_1 h_2 - g_2 h_2^2 + \mathcal{O}\left(h_2^3\right)$$

$$f_3 = f_{\text{exact}} + g_1 h_3 - g_2 h_3^2 + \mathcal{O}\left(h_3^3\right) \tag{13}$$

If the three solutions (f_1, f_2, and f_3) are known along with the three mesh spacing values (h_1, h_2, and h_3), then Eq. (13) forms a linear system that may be solved for the first- and second-order error coefficients (g_1 and g_2) and the third-order accurate estimate of the exact solution f_{exact}. If we arbitrarily set $h_1 = 1$, then the solution to this linear system gives

$$g_1 \cong \frac{-\varepsilon_{32} + r^2 \varepsilon_{21}}{r(r-1)^2} \tag{14}$$

$$g_2 \cong \frac{\varepsilon_{32} - r\varepsilon_{21}}{r(r+1)(r-1)^2} \tag{15}$$

$$f_{\text{exact}} \cong f_1 + \frac{\varepsilon_{32} - (r^2 + r - 1)\varepsilon_{21}}{(r+1)(r-1)^2} \tag{16}$$

where again a constant mesh refinement factor r is assumed.

The mixed-order method has been applied to the surface pressure solutions on the seven mesh levels. Figure 12 shows the behavior of the error in the surface pressure at the stagnation point. The error is calculated using the third-order accurate estimate for f_{exact} from Eq. (16) using the finest three mesh levels and can be written as

$$\text{spatial error}\,(\%) = [(f_k - f_{\text{exact}})/f_{\text{exact}}] \times 100 \tag{17}$$

Because of the highly refined nature of the finest three grids, these spatial error estimates are expected to be very close to the true discretization error. The spatial error estimates are plotted vs $h = (N_1/N_k)^{1/2}$, where N_1 is the total number of cells on mesh 1 (the fine mesh) and N_k is the number of cells on mesh k. Because a grid refinement factor of two (grid halving in each coordinate direction) was used, the discrete solution points fall at 1, 2, 4, 8, 16, 32, and 64 (from finest to coarsest).

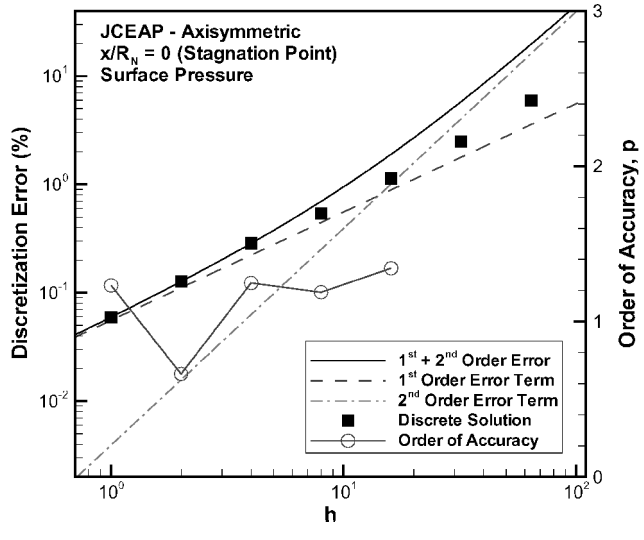

Fig. 12 Magnitude of the error components in surface pressure at $x/R_N = 0$: stagnation point.

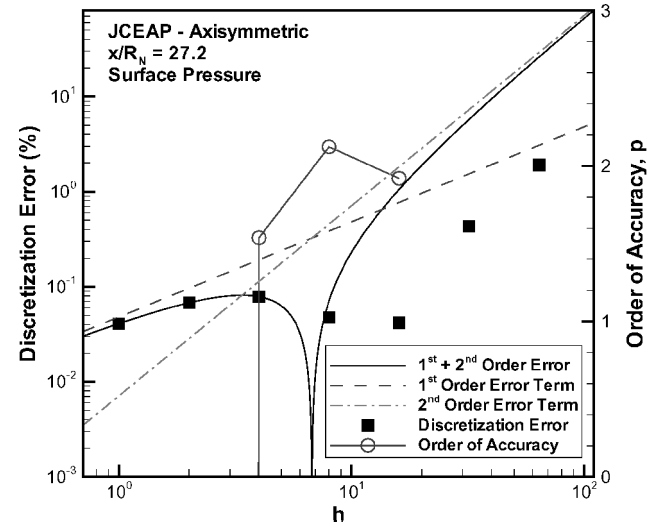

Fig. 13 Magnitude of the error components in surface pressure at $x/R_N = 27.2$.

Also shown in Fig. 12 are the normalized magnitudes of the first- and second-order error terms, respectively,

$$|g_1 h/f_{\text{exact}}| \times 100, \qquad \left|g_2 h^2 \big/ f_{\text{exact}}\right| \times 100 \tag{18}$$

along with the normalized magnitude of their sum

$$\left|\left(g_1 h + g_2 h^2\right) \big/ f_{\text{exact}}\right| \times 100 \tag{19}$$

The first-order error term has a slope of unity on the log–log plot (Fig. 12), whereas the second-order error term has a slope of two. The magnitude of the sum of the two terms (solid line) is forced to pass through the points associated with meshes 1–3 because these discrete solutions are used in the solution to Eqs. (14–16). First-order accuracy is seen in the fine grid solutions, whereas the coarse grid solutions begin to exhibit a second-order behavior. Indeed, the discretization error on the coarser grids begins to approach the second-order slope. In this case, the first- and second-order coefficients (g_1 and g_2) have the same sign, so that the magnitude of the sum of the error terms is larger than each of the individual error terms. Also shown in Fig. 12 is the observed order of accuracy p as calculated from Eq. (11). Because the order of accuracy requires three discrete solutions, results are only available for meshes 1–5. For this case, the observed order of accuracy is well defined and varies around unity ($p = 1$).

The error in surface pressure at a location halfway down the body ($x/R_N = 27.2$) is given in Fig. 13. As was seen in Fig. 12, the solutions display first-order grid convergence for the finer grids and second-order convergence for the coarser grids. In this case, however, the first- and second-order error coefficients are of opposite sign, giving error cancellation at the crossover point ($h \approx 7$). The nonmonotone behavior predicted from the mixed-order error analysis (using the three finest mesh solutions only) is qualitatively seen in the discretization error estimates on the coarser meshes. The fact that the pressure does not converge monotonically results in singular behavior for the standard method for calculating the observed order of accuracy (also shown in Fig. 13) because the argument of the natural logarithm in Eq. (11) becomes negative.

The error in the forebody drag, a global quantity, is shown in Fig. 14. Again, the first- and second-order error terms are of opposite sign, resulting in error cancellation at $h \approx 4$. Because there is a discrete solution at $h = 4$, the effects of the error cancellation are quite pronounced in this case, with the error in the drag for the 240×240 cell grid almost two orders of magnitude lower than on the 960×960 grid. Figure 14 clearly demonstrates how nonmonotonic grid convergence can be caused by error cancellation for

Table 3 Normalized surface pressure p/p_∞ for $p_\infty = 286.8$ N/m^2 (percent error)

Mesh level	Stagnation point	Sphere–cone juncture	$x/R_N = 27.2$
15×15	(−5.9%)	(52.4%)	(−1.9%)
30×30	77.66557 (−2.5%)	9.370761 (23.3%)	3.692487 (−0.43%)
60×60	78.74681 (−1.1%)	8.315920 (9.4%)	3.706927 (−0.041%)
120×120	79.22133 (−0.54%)	7.858402 (3.4%)	3.710240 (0.048%)
240×240	79.42112 (−0.28%)	7.723745 (1.6%)	3.711381 (0.079%)
480×480	79.54735 (−0.13%)	7.658460 (0.74%)	3.710982 (0.068%)
960×960	79.60108 (−0.060%)	7.629082 (0.36%)	3.709991 (0.041%)
First- and second-order extrapolation	79.64854	7.601881	3.708474

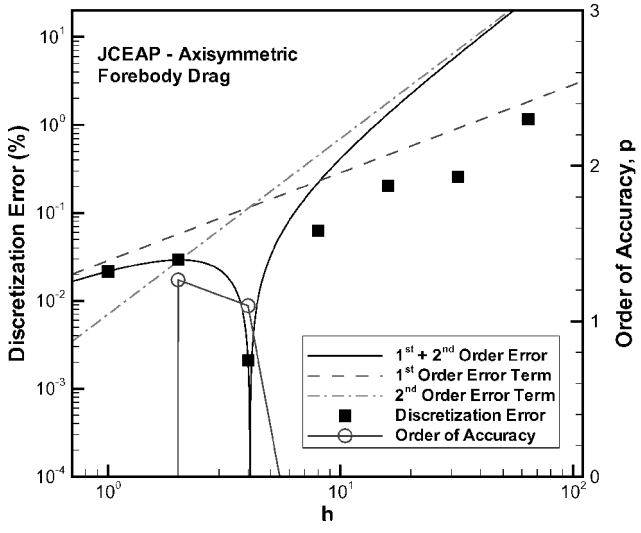

Fig. 14 Magnitude of the error components in forebody drag, neglecting base drag.

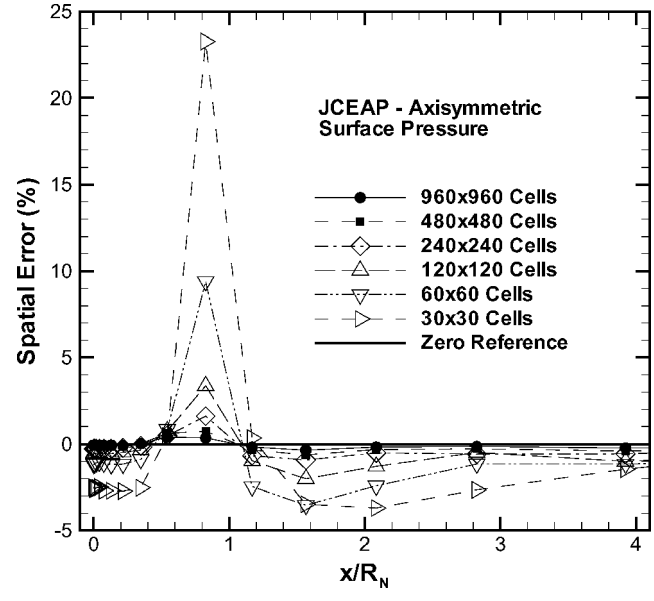

Fig. 15 Error in the surface pressure distributions in the nose region using six mesh levels.

mixed-order schemes. Moreover, nonmonotonic grid convergence greatly complicates the process of error assessment.

The assumption that both the first- and second-order error components are important is supported by the qualitative agreement between the coarse grid error estimates (solid lines) and the discrete errors (symbols) shown in Figs.12–14. That the order of accuracy tends toward first order as the grid is refined is not a new finding. Carpenter and Casper[33] showed that all shock-capturing schemes reverted to first order behind the shock for sufficiently refined meshes. Two differences between their approach and the current work are that they employed higher-order methods (third- and fourth-order methods) and that they did not use a flux limiting procedure. The fact that Carpenter and Casper saw a reduction to first order without using flux limiters is surprising and implies that the current shock-capturing schemes are only capable of transferring information in a first-order manner through discontinuities, at least in two dimensions and higher. See Ref. 33 for more details.

The capturing of discontinuities, for example, shock waves, without oscillation requires a reduction in the local spatial accuracy of a numerical scheme to first order.[34] The prevention of oscillations is especially critical for chemically reacting flows, where nonphysical temperature extrema can strongly affect the chemistry. For the Mach 8 sphere–cone simulations presented herein, the first-order behavior at the shock wave leads to the presence of a first-order error component (however small) everywhere downstream due to error "pollution." As the mesh spacing is refined and $h \to 0$, this first-order error component must eventually dominate. The standard method for assessing the order of spatial accuracy given in Eq. (11) is inadequate when the first- and second-order error terms are of the same magnitude. In a strict sense, the asymptotic grid convergence regime occurs when there is a single dominant error term as $h \to 0$, which for this case is first order. Downstream of the shock wave, the coefficient on the first-order error term g_1 is small, with the magnitude possibly related to the proximity to the discontinuity. In these regions, a second-order asymptotic region may exist, which corresponds to the local discretization error. Once sufficient grid refinement is performed, the errors from the discontinuity become significant, thus, resulting in a first-order asymptotic region. For practical purposes, the second-order asymptotic range should be sufficient for engineering calculations; however, the error and order of accuracy analyses must take into account that both first- and second-order error terms may be present. The effects of the first-order pollution error from the discontinuity could be mitigated by clustering to the shock; however, no attempt to provide such clustering was made in the current work.

Error Assessment

The error of the surface pressure distributions relative to the third-order accurate estimate from Eq. (16) is presented in Fig. 15 for the nose and the beginning of the conical region. The errors are largest at the sphere–cone juncture ($x/R_N \approx 0.83$), which indicates that additional grid refinement is required at geometric boundaries with discontinuous surface curvature. The errors on the conical portion of the body are given in Fig. 16 for the region in which experimental data are available ($6 < x/R_N < 46$). The spatial errors in this region are below 0.06% for mesh 1 and 0.11% for mesh 2. The comparisons to the experimental data are presented in the companion paper.[4]

The numerical errors at the stagnation point, the sphere–cone tangency point, and $x/R_N = 27.2$ are summarized in Table 3. The nonmonotonic grid convergence behavior observed at $x/R_N = 27.2$ in Fig. 13 is clearly evident in Table 3. With the exception of the sphere–cone juncture point, the spatial errors on the three finest meshes are all below 1%. For the two finest meshes, the maximum numerical errors in the surface pressure are 0.36% (mesh 1) and 0.74% (mesh 2) and are considered sufficiently small to qualify as numerical benchmark solutions for code verification.

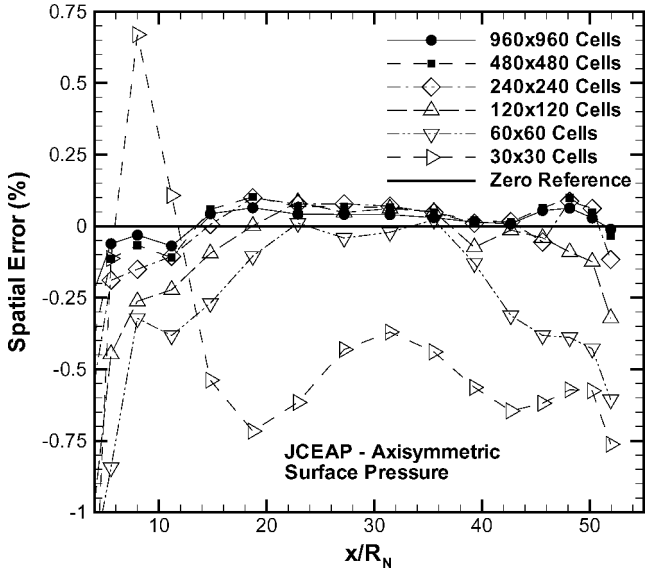

Fig. 16 Error in the surface pressure distributions along the cone using six mesh levels.

Conclusions

Code verification efforts were performed including comparisons to inviscid benchmark solutions and a code-to-code comparison. These activities give increased confidence that the SACCARA code is free from coding errors. Simulations have been conducted for the laminar, perfect gas flow over a 10-deg half-angle sphere–cone. A method for monitoring the iterative convergence error during a calculation was presented. Application of this technique to the fine grid calculation was used to obtain iterative convergence of the surface pressure down to machine zero, or 10^{-12}% error. Solutions on seven mesh levels were obtained to assess the adequacy of the computational meshes and to gain insight into the grid convergence behavior. Nonmonotonic convergence of the surface pressure and forebody drag was observed and was found to be related to the presence of both first- and second-order terms in the discretization error. The grid convergence errors for surface pressure were estimated to be below 0.36 and 0.74% for meshes 1 and 2, respectively. These numerical errors are sufficiently small to qualify as numerical benchmark solutions. Detailed surface and field files for this numerical benchmark solution are available from the first author.

In general, it is desirable to use numerical schemes that reduce to first-order accuracy through discontinuities such as shock waves to prevent numerical oscillations. The first-order behavior at the shock wave leads to the presence of a first-order error component (however small) everywhere downstream. Because the mesh spacing is sufficiently refined, this first-order error component must eventually dominate. The standard method for assessing the order of spatial accuracy was shown to be inadequate when the first- and second-order error terms were of similar magnitude. An alternative method was applied for analyzing the convergence behavior of mixed first- and second-order schemes. This method allows solution nonmonotonicity due to the cancellation of first- and second-order error terms. Although this alternative method requires only three grid solutions, the authors recommend that a fourth mesh level be computed to verify that the error behaves as predicted.

Acknowledgments

This work was supported by Sandia National Laboratories and the Department of Energy's Accelerated Strategic Computing Initiative. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under Contract DE-AC04-94AL85000. The authors thank Frederick Blottner of Sandia National Laboratories for his many helpful discussions on the accuracy of numerical simulations. We also thank Mark Carpenter of NASA Langley Research Center for providing the inviscid benchmark solution using the Chebyshev collocation spectral method.

References

[1]*Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, G-077-1998, AIAA, Reston, VA, 1998, p. 3.

[2]Salari, K., and Knupp, P., "Code Verification by the Method of Manufactured Solutions," Sandia National Labs., Rept. SAND2000-1444, Albuquerque, NM, June 2000.

[3]Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998.

[4]Roy, C. J., Oberkampf, W. L., and McWherter-Payne, M. A., "Verification and Validation for Laminar Hypersonic Flowfields, Part 2: Validation," *AIAA Journal*, Vol. 41, No. 10, 2003, pp. 1944–1954.

[5]Oberkampf, W. L., Aeschliman, D. P., Henfling, J. F., and Larson, D. E., "Surface Pressure Measurements for CFD Code Validation in Hypersonic Flow," AIAA Paper 95-2273, June 1995.

[6]Oberkampf, W. L., Aeschliman, D. P., Henfling, J. F., Larson, D. E., and Payne, J. L., "Surface Pressure Measurements on a Hypersonic Vehicle," AIAA Paper 96-0669, Jan. 1996.

[7]Wong, C. C., Soetrisno, M., Blottner, F. G., Imlay, S. T., and Payne, J. L., "PINCA: A Scalable Parallel Program for Compressible Gas Dynamics with Nonequilibrium Chemistry," Sandia National Labs., Rept. SAND 94-2436, Albuquerque, NM, April 1995.

[8]Wong, C. C., Blottner, F. G., Payne, J. L., and Soetrisno, M., "Implementation of a Parallel Algorithm for Thermo-Chemical Nonequilibrium Flow Solutions," AIAA Paper 95-0152, Jan. 1995.

[9]"INCA User's Manual," Ver. 2.0, Amtec Engineering, Inc., Bellevue, WA, 1995.

[10]Payne, J. L., and Walker, M. A., "Verification of Computational Aerodynamic Predictions for Complex Hypersonic Vehicles using the INCA™ Code," AIAA Paper 95-0762, Jan. 1995.

[11]Roy, C. J., McWherter-Payne, M. A., and Oberkampf, W. L., "Verification and Validation for Laminar Hypersonic Flowfields," AIAA Paper 2000-2550, June 2000.

[12]Roy, C. J., Gallis, M. A., Bartel, T. J., and Payne, J. L., "Navier–Stokes and DSMC Simulations for Hypersonic Laminar Shock-Shock Interaction Flows," AIAA Paper 2002-0737, Jan. 2002.

[13]Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite Difference Methods," *Journal of Computational Physics*, Vol. 40, 1981, pp. 263–293.

[14]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136.

[15]Yoon, S., and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier–Stokes Equations," AIAA Paper 87-0600, Jan. 1988.

[16]Yoon, S., and Kwak, D., "Artificial Dissipation Models for Hypersonic External Flow," AIAA Paper 88-3708, July 1988.

[17]Peery, K. M., and Imlay, S. T., "An Efficient Implicit Method for Solving Viscous Multi-Stream Nozzle/Afterbody Flow Fields," AIAA Paper 86-1380, June 1986.

[18]Payne, J. L., and Hassan, B., "Massively Parallel Computational Fluid Dynamics Calculations for Aerodynamics and Aerothermodynamics Applications," *Proceedings of the 1998 HPCCP/CAS Workshop*, NASA CP-1999-208757, 1999, pp. 111–116.

[19]Walker, M. M., and Oberkampf, W. L., "Joint Computational/Experimental Aerodynamics Research on a Hypersonic Vehicle, Part 2: Computational Results," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2010–2016.

[20]Kopriva, D. A., "Spectral Methods for the Euler Equations: The Blunt-Body Problem Revisited," *AIAA Journal*, Vol. 29, No. 9, 1991, pp. 1458–1462.

[21]Carpenter, M. H., Atkins, H. L., and Singh, D. J., "Characteristic and Finite-Wave Shock-Fitting Boundary Conditions for Chebyshev Methods," *Transition, Turbulence, and Combustion*, edited by M. Y. Hussaini, T. B. Gatski, and T. L. Jackson, Vol. 2, Kluwer Academic, Norwell, MA, 1994, pp. 301–312.

[22]Lyubimov, A. N., and Rusanov, V. V., "Gas Flows Past Blunt Bodies, Part I: Calculation Method and Flow Analysis," NASA TT F-714, Feb. 1973.

[23]Lyubimov, A. N., and Rusanov, V. V., "Gas Flows Past Blunt Bodies, Part II: Tables of the Gasdynamic Functions," NASA TT F-715, Feb. 1973.

[24]Blottner, F. G., and Larson, D. E., "Navier–Stokes Code NS3D for Blunt Bodies, Part I: Analysis, Results, and Verification," Sandia National Labs., Rept. SAND 88-0504/1, Albuquerque, NM, March 1988.

[25]Blottner, F. G., "Accurate Navier–Stokes Results for the Hypersonic Flow over a Spherical Nosetip," *Journal of Spacecraft and Rockets*, Vol. 27, No. 2, 1990, pp. 113–122.

[26]Walker, M. A., "SPRINTRUN: A User Friendly Input Processor for the SPRINT Code," Sandia National Labs., Rept. SAND 89-0625, Albuquerque, NM, March 1990.

[27]Stalnaker, J. F., Nicholson, L. A., Hanline, D. S., and McGraw, E. H., "Improvements to the AFWAL Parabolized Navier–Stokes Code Formulation," U.S. Air Force Wright Aeronautical Lab., TR-86-3076, Wright–Patterson AFB, Dayton, OH, Sept. 1986.

[28]Roy, C. J., and Blottner, F. G., "Assessment of One- and Two-Equation Turbulence Models for Hypersonic Transitional Flows," *Journal of Spacecraft and Rockets*, Vol. 38, No. 5, 2001, pp. 699–710; also AIAA Paper 2000-0132, 2000.

[29]Ferziger, J. H., and Peric, M., *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1996, pp. 115–126.

[30]Ferziger, J. H., and Peric, M., "Further Discussion of Numerical Errors in CFD," *International Journal for Numerical Methods in Fluids*, Vol. 23, No. 12, 1996, pp. 1263–1274.

[31]de Vahl Davis, G., "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution," *International Journal for Numerical Methods in Fluids*, Vol. 3, No. 3, 1983, pp. 249–264.

[32]Roy, C. J., "Grid Convergence Error Analysis for Mixed-Order Numerical Schemes," *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 595–604; also AIAA Paper 2001-2606, June 2001.

[33]Carpenter, M. H., and Casper, J. H., "Accuracy of Shock Capturing in Two Spatial Dimensions," *AIAA Journal*, Vol. 37, No. 9, 1999, pp. 1072–1079.

[34]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. I. The Quest of Monotonicity," *Lecture Notes in Physics*, Vol. 18, Springer-Verlag, Berlin, 1973, pp. 163–168.

G. V. Candler
Associate Editor