JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 33, No. 1, January–February 1996

Code Validation for High-Speed Flow Simulation Over Satellite Launch Vehicle

J. L. F. Azevedo* and P. Moraes Jr.†
Instituto de Aeronáutica e Espaco, São José dos Campos, SP 12228-904, Brazil
and
C. R. Maliska,‡ C. H. Marchi,§ and A. F. C. Silva¶
Universidade Federal de Santa Catarina, Florianópolis, SC 88040-900, Brazil

The development of an all-speed Euler and/or Navier–Stokes flow simulation code that uses a segregated finite volume algorithm for three-dimensional body-conforming curvilinear coordinates with a collocated variable arrangement is described. The efforts towards the physical validation of this code are the major contribution of the present work. The details of an experimental wind-tunnel investigation for pressure distribution and force measurements for the Brazilian Satellite Launch Vehicle (VLS) are also described. The computational results obtained for subsonic and supersonic flow conditions are in very good agreement with the experimental data. Transonic calculations, however, show poorer agreement with the available data for equivalent and even more refined meshes. The computational results produced by the present codes are already supplying aerodynamic information that is currently being used in the VLS design process.

Nomenclature

$\hat{A}, \hat{B}, \hat{C}$	= inviscid flux Jacobian matrices in generalized coordinates
C_p	= specific heat at constant pressure
Cp	= pressure coefficient
$\bar{E}, \bar{F}, \bar{G}$	= flux vectors in generalized curvilinear coordinates
e	= total energy per unit of volume
e_i	= specific internal energy
J	= Jacobian of the coordinate transformation
L_ξ, L_η, L_ζ	= left-hand-side operators for the central-difference scheme
M	= Mach number
n	= temporal index
p	= pressure
p^ϕ	= pressure-gradient source terms in the segregated all-speed scheme
$\bar{Q}$	= vector of conserved variables in generalized curvilinear coordinates
Re	= Reynolds number
R_ξ, R_η, R_ζ	= right-hand-side operators for the central-difference scheme
S^ϕ	= viscous source terms in the segregated all-speed scheme
t	= time (Cartesian coordinate system)
U, V, W	= contravariant velocity components
u, v, w	= Cartesian velocity components
x, y, z	= Cartesian coordinates
α	= angle of attack
$\alpha_{11}, \alpha_{12}, \ldots, \alpha_{33}$	= metric coefficients in the segregated all-speed scheme
γ	= ratio of specific heats
ϵ_E	= explicit-side artificial dissipation parameter
ϵ_I	= implicit-side artificial dissipation parameter
μ	= viscosity coefficient
ξ, η, ζ	= generalized curvilinear coordinates
$\xi_x, \xi_y, \ldots, \eta_x, \ldots, \zeta_z$	= metric terms
ρ	= density
τ	= time (generalized curvilinear coordinate system)
ϕ	= generic variable in the segregated all-speed scheme
φ	= model roll angle

Introduction

THE first Brazilian Satellite Launch Vehicle (VLS) is of the cluster type with four strap-on boosters around the central core, as shown in Fig. 1. The hammerhead-type fairing is of nonconventional shape, but its use is rather common on satellite launchers to accommodate spacecraft with a diameter larger than the last boosting stage. The prediction of the local aerodynamic characteristics of such a fairing, for shape optimization and design purposes, is typically made primarily through wind-tunnel tests. These tests, however, have high cost and usually require a long turnaround time for the complete cycle, which encompasses test specification, model production, the test itself, and data reduction. Therefore, there is great interest in, and much effort has been put into developing, the capability of simulating such aerodynamic flowfields. Of particular interest is accurately computing the high-speed flow about the hammerhead fairing. Thus far only steady-state cases have been considered.

A wind-tunnel test program was established and performed both for actual design and for code validation purposes. This program has covered a wide range of test parameters such as Mach number, Reynolds number, and angle of incidence. The test program has attempted to cover the relevant aerodynamic regimes for the complete expected flight trajectory of the vehicle. Moreover, the test program was organized in such a way that the experimental data could be readily used for computational fluid dynamics code development and validation studies. In particular, although the configuration tested was inherently complex, care was exercised to avoid booster attachment details and body external protuberances, which would be extremely difficult to simulate numerically without severely penalizing the size of the computational grids.

Presented as Paper 93-3046 at the AIAA 24th Fluid Dynamics Conference, Orlando, FL, June 6–9, 1993; received Oct. 29, 1993; revision received May 20, 1995; accepted for publication June 22, 1995.

*Head, Computational Fluid Dynamics Group. Member AIAA.
†Head, Aerodynamics Subdivision.
‡Professor. Member AIAA.
§Research Associate. Member AIAA.
¶Associate Professor.

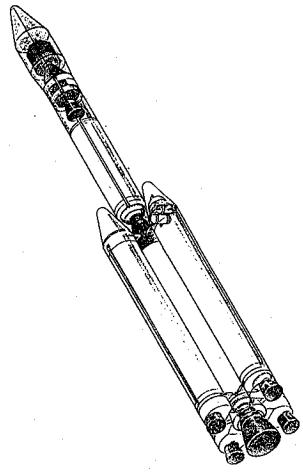


Fig. 1 General configuration of the VLS.

The institutions represented in this paper have been actively involved in the development of high-speed flow simulation codes for several years now. It is hoped that these codes will help to ease the burden presently put on the experimental investigations for the VLS aerodynamic design. However, before these codes can be used in a design environment, they must be thoroughly validated. Moreover, although this requirement is more subjective, project engineers have to develop confidence in the use of the codes for their analysis and design work. Two distinctive lines of work have been pursued. The first one was based on the methods typically used in aerodynamics, in which the governing equations are solved simultaneously with local linearization based on Jacobian matrices. The other was based on segregated solution methods, which were originally developed for low-speed flows and heat-transfer problems. The emphasis in this paper will be on the second class of methods, since a novel methodology for flow simulation at all speeds has been recently developed by some of the present authors.[1–3]

It would be appropriate at this point to discuss some of the motivations for pursuing the all-speed method. A first aspect concerns the need for simulating all phases of flight of a launch vehicle in the lower atmosphere. These vehicles are typically flying at a high speed during most of their atmospheric flight, but there are a few situations in which the flowfield over the vehicle is nearly incompressible. For instance, a vehicle on the launching pad is subjected to lateral winds, and the vortical flow structure created by the vehicle in this situation can be very important for its dynamical stability. Standard compressible-flow codes most certainly would have difficulties in simulating such flowfields. If one considers the effort and the time involved in the development and validation of these flow simulation codes, it is worthwhile investing in the development of a code that would be capable of performing all the required simulations. Moreover, at the time that the development work described in this paper was initiated, there was a fairly large amount of expertise in the Brazilian scientific community on flow simulation methods for low-speed flows, but not for high-speed compressible flows. VLS program managers at the time decided that it was worthwhile investing in a few groups that had demonstrated such expertise, and charging them with the objective of extending their flow simulation methodologies to higher-speed flow regimes. This extension could assume various forms, and one of them is the all-speed method described in this paper.

The present work will briefly describe the experimental investigation and code development. This paper will concentrate on the physical validation of the results produced by the new code against the experimental data. Code-to-code comparisons are also performed, using independent computational results generated by a central-difference, implicit, approximate factorization algorithm, which has been previously tested for similar configurations.[4,5] Further details of the experimental test program are reported in Ref. 6. More details of the numerical formulation in the new segregated, finite-volume code are described in Refs. 1–3 and 7.

Experimental Test Program

The general wind-tunnel test program comprised three different test series, although only the first two are included in the present work. These tests considered the acquisition of global and local vehicle characteristics through the measurement of forces, moments, and local pressures along the vehicle. The third test series considered the simulation of liftoff conditions in a low-speed wind tunnel. The computational simulation of liftoff conditions has not been attempted yet, and therefore the results of this third test series will be of no concern for the present work. The major interest in the present case will be the local pressure measurements reported in Ref. 8.

Pressure measurement tests were performed both in a continuous-type transonic wind tunnel and in a blowdown supersonic tunnel. The continuous transonic tunnel has a test-section size of 1.75×1.77 m, and it uses perforated walls for tests in the transonic speed regime. Tests were conducted in this tunnel for the Mach-number range from 0.5 up to 2.5. Tests in the Mach number range $2.5 \leq M_\infty \leq 3.75$ were performed in the blowdown tunnel. A 1:15-scale smooth model with nonattached boosters was used. The model had approximately 320 pressure taps distributed along the surface of the vehicle's central core and boosters. The model was mounted in the test section using a five-sting support system, which allowed the simulation of the strap-on separation in a static manner. A sketch of the five-sting support system is indicated in Fig. 2. To increase the amount of azimuthal-pressure information without exceeding the available internal space of the model, pressure taps were distributed along various azimuthal positions on both the core and boosters, and the model was tested at two different roll angles, $\varphi = 0$ and 90 deg. This is indicated in Fig. 3, where a front view of the model is sketched. Since all tests were performed for positive and negative angles of attack, this distribution of pressure-tap lines allowed measurements every 30 deg in the azimuthal direction, from leeward to windward, for all bodies involved. For each Mach number tests were performed from −6 to +6 deg in angle of attack at 2-deg intervals.

One of the major thrusts for the specification of the five-sting support was the need for good experimental data for code validation. Since it is very difficult to model computationally all the details of actual booster attachments, it was decided to perform experimental tests without these attachments to provide the computational specialists with clean data with which to compare their results.

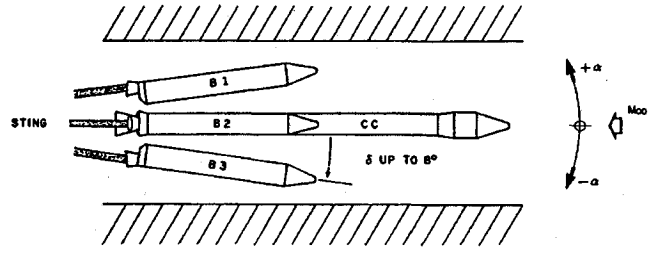
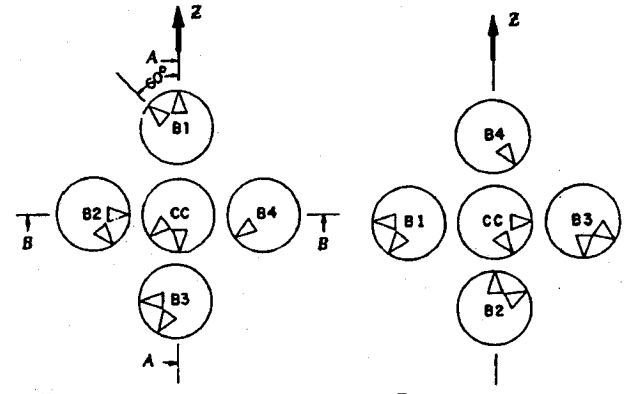


Fig. 2 Sketch of the wind-tunnel model with the five-sting support system.



a) $\Phi = 0$ deg b) $\Phi = 90$ deg

Fig. 3 Front view of the pressure-tap locations: ▷, presure orifices; B, booster; CC, central body; and —, pressure orifice line.

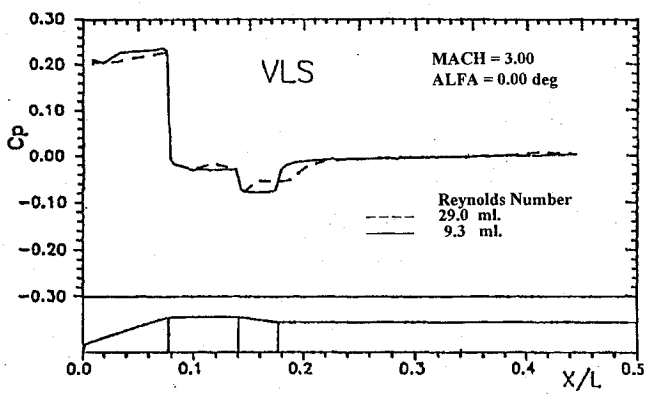


Fig. 4 Reynolds-number effects on vehicle longitudinal pressure coefficient distribution.

Moreover, it is fairly clear that in a code validation effort one should not unnecessarily complicate the geometry, because the grid, and the corresponding computational time, can become very large. The solution to this problem was to support each individual body with its own sting. Certainly, the same system can be used to test the vehicle while simulating the condition of strap-on separation.

Finally, it is worth observing that the wind-tunnel tests have also contemplated the problem of Reynolds-number influence in the flow topology, i.e., the effect of separation and reattachment points in the flowfield. In particular, experimental results have been able to clearly indicate the effect of the Reynolds number in the flow separated region in the boattail. This result is shown in Fig. 4 for $M_\infty = 3$ and zero angle of attack, at Reynolds numbers (based on the model length) of 9.3×10^6 and 29.0×10^6. A very important effect is seen, because flow separation in the boattail can adversely affect the buffeting behavior of the vehicle.

Central-Difference Algorithm

The compressible Euler equations[9] can be written in strong conservation-law form for general three-dimensional, body-conforming, curvilinear coordinates[10] as

$$\frac{\partial \bar{Q}}{\partial \tau} + \frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} + \frac{\partial \bar{G}}{\partial \zeta} = 0 \tag{1}$$

where $\bar{Q}$ is defined as

$$\bar{Q} = J^{-1} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix} \tag{2}$$

$\bar{E}$, $\bar{F}$, and $\bar{G}$ can be written as

$$\bar{E} = J^{-1} \begin{Bmatrix} \rho U \\ \rho u U + p\xi_x \\ \rho v U + p\xi_y \\ \rho w U + p\xi_z \\ (e+p)U - p\zeta_t \end{Bmatrix} \tag{3}$$

$$\bar{F} = J^{-1} \begin{Bmatrix} \rho V \\ \rho u V + p\eta_x \\ \rho v V + p\eta_y \\ \rho w V + p\eta_z \\ (e+p)V - p\eta_t \end{Bmatrix} \tag{4}$$

$$\bar{G} = J^{-1} \begin{Bmatrix} \rho W \\ \rho u W + p\zeta_x \\ \rho v W + p\zeta_y \\ \rho w W + p\zeta_z \\ (e+p)W - p\zeta_t \end{Bmatrix} \tag{5}$$

In the preceding equations, the usual nomenclature is being used: ρ is the density, u, v, and w are the Cartesian components of velocity, and e is the total energy per unit of volume. The equations have been nondimensionalized following the work in Refs. 10 and 11.

The pressure p can be obtained from the equation of state for perfect gases:

$$p = (\gamma - 1)\rho e_i = (\gamma - 1)\left[e - \tfrac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{6}$$

The contravariant velocity components are defined as

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w \tag{7a}$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w \tag{7b}$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \tag{7c}$$

Throughout this work, the curvilinear coordinate system is defined so that ξ is the longitudinal direction, η is the normal direction, and ζ is the circumferential direction. This coordinate system is obtained from the transformation of variables

$$\tau = t \tag{8a}$$

$$\xi = \xi(x, y, z, t) \tag{8b}$$

$$\eta = \eta(x, y, z, t) \tag{8c}$$

$$\zeta = \zeta(x, y, z, t) \tag{8d}$$

The Jacobian of the transformation J can be expressed as

$$J = (x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi)^{-1} \tag{9}$$

Expressions for the various metric relations can be found in Refs. 10 and 11, among other references.

In the present case, the above governing equations were implemented through the use of finite difference methods. The implicit Euler method was used for the time march, and the spatial derivatives were approximated by three-point, second-order central differences. The Beam–Warming implicit approximate factorization scheme[12,13] was used for the solution of the resulting finite difference equations to obtain a cost-efficient algorithm. The resulting scheme is second-order accurate in space, as mentioned, but it is only first-order accurate in time, because of the use of the implicit Euler method.

The factored finite difference equations can be written in the delta form as

$$L_\eta L_\zeta L_\xi \Delta_t \bar{Q}^n = R_\xi + R_\eta + R_\zeta \tag{10}$$

The various operators are defined as

$$L_\xi = I + \Delta t\, \delta_\xi \hat{A}^n - \epsilon_I\, \Delta t\, J^{-1} \nabla_\xi \Delta_\xi J \tag{11a}$$

$$L_\eta = I + \Delta t\, \delta_\eta \hat{B}^n - \epsilon_I \Delta t J^{-1} \nabla_\eta \Delta_\eta J \tag{11b}$$

$$L_\zeta = I + \Delta t\, \delta_\zeta \hat{C}^n - \epsilon_I\, \Delta t\, J^{-1} \nabla_\zeta \Delta_\zeta J \tag{11c}$$

$$R_\xi = -\Delta t\, \delta_\xi \bar{E}^n - \epsilon_E\, \Delta t\, J^{-1} (\nabla_\xi \Delta_\xi)^2 J \bar{Q}^n \tag{11d}$$

$$R_\eta = -\Delta t\, \delta_\eta \bar{F}^n - \epsilon_E\, \Delta t\, J^{-1} (\nabla_\eta \Delta_\eta)^2 J \bar{Q}^n \tag{11e}$$

$$R_\zeta = -\Delta t\, \delta_\zeta \bar{G}^n - \epsilon_E\, \Delta t\, J^{-1} (\nabla_\zeta \Delta_\zeta)^2 J \bar{Q}^n \tag{11f}$$

In the above, δ_ξ, δ_η, and δ_ζ are central-difference operators; ∇_ξ, ∇_η, and ∇_ζ are backward-difference operators; and Δ_ξ, Δ_η and Δ_ζ are forward-difference operators in the ξ, η, and ζ directions, respectively. As an example,

$$\delta_\xi \bar{Q}^n_{i,j,k} = \tfrac{1}{2}\left(\bar{Q}^n_{i+1,j,k} - \bar{Q}^n_{i-1,j,k}\right) \tag{12a}$$

$$\nabla_\xi \bar{Q}^n_{i,j,k} = \bar{Q}^n_{i,j,k} - \bar{Q}^n_{i-1,j,k} \tag{12b}$$

$$\Delta_\xi \bar{Q}^n_{i,j,k} = \bar{Q}^n_{i+1,j,k} - \bar{Q}^n_{i,j,k} \tag{12c}$$

Finally, Δ_t is a forward-difference operator in time given by

$$\Delta_t \bar{Q}^n = \bar{Q}^{n+1} - \bar{Q}^n \tag{13}$$

Artificial dissipation terms have been introduced in the operators described by Eq. (11) to maintain the stability of the numerical solution process. Fourth-order numerical dissipation terms were added to the right-hand-side operators, and second-order terms were used in the left-hand-side operators. From an accuracy standpoint, one would like to also use fourth-order artificial dissipation in the implicit operators. However, computational efficiency constraints prevent such use. The Jacobian matrices $\hat{A}^n$, $\hat{B}^n$, and $\hat{C}^n$ are described in detail elsewhere in the literature (see, for instance, Ref. 11).

Segregated All-Speed Scheme

Preliminary Considerations

Most of the available algorithms for the solution of the Euler, or the Navier–Stokes, equations are only suitable for either compressible or incompressible flows. Although there is no universal agreement as to the cause of this behavior, there are several authors who believe that the key point is associated with the form in which density is treated in the mass conservation equation. In the present work, a numerical method for the solution of three-dimensional, viscous or inviscid, all-speed flows of a perfect gas is considered. The method is developed in the framework of a boundary-fitted, structured, finite-volume spatial discretization, and uses a fully implicit time march procedure. The system of equations is solved in a segregated manner in which one of the dependent variables is assumed to be active in each of the governing equations. All other variables in that equation assume a passive role in the linearization process. Primitive variables are used as dependent variables. Moreover, a collocated variable arrangement is employed which renders the method compact and provides considerable savings in the storage of geometric information concerning the grid system.

The method is derived from the well-known approach used for incompressible flows,[14] whereby the mass conservation equation is transformed into an equation to find the pressure. In this equation, density and velocity components are replaced by relations involving pressure obtained from the equation of state and from approximate forms of the momentum equations, respectively. This allows both velocity and density to remain active in the continuity equation[1] and therefore enables the method to treat both compressible and incompressible flow problems. It should be pointed out that a similar approach has been presented in Ref. 15. In this paper, however, the procedure developed in Refs. 1–3 is employed. Moreover, the present version of the code has incorporated the ability of considering multiple-block grids. The difficulty in discretizing complex computational domains such as the flowfield about the VLS vehicle has indicated the need for such procedure.

Formulation of the Method

The segregated all-speed finite volume scheme as applied to the Navier–Stokes equations will be described in this section. For its application to the Euler equations, one simply has to neglect the viscous terms and make obvious changes on the wall boundary conditions. In the nomenclature that is usually used with segregated finite volume schemes, the Navier–Stokes equations can be written for general, body-conforming, curvilinear coordinates as[16]

$$\begin{aligned}
&\frac{1}{J}\frac{\partial}{\partial t}(\rho\phi)\frac{\partial}{\partial \xi}(\rho U\phi)\frac{\partial}{\partial \eta}(\rho V\phi) + \frac{\partial}{\partial \zeta}(\rho W\phi) \\
&\quad = \Gamma^{\phi}\frac{\partial}{\partial \xi}\left(\alpha_{11}J\frac{\partial \phi}{\partial \xi} + \alpha_{12}J\frac{\partial \phi}{\partial \eta} + \alpha_{13}J\frac{\partial \phi}{\partial \zeta}\right) \\
&\quad + \Gamma^{\phi}\frac{\partial}{\partial \eta}\left(\alpha_{12}J\frac{\partial \phi}{\partial \xi} + \alpha_{22}J\frac{\partial \phi}{\partial \eta} + \alpha_{23}J\frac{\partial \phi}{\partial \zeta}\right) \\
&\quad + \Gamma^{\phi}\frac{\partial}{\partial \zeta}\left(\alpha_{13}J\frac{\partial \phi}{\partial \xi} + \alpha_{23}J\frac{\partial \phi}{\partial \eta} + \alpha_{33}J\frac{\partial \phi}{\partial \zeta}\right) - \hat{p}^{\phi} + \hat{S}^{\phi}
\end{aligned} \tag{14}$$

The general form given in Eq. (14), with the appropriate various source terms, can recover the continuity equation, the three momentum equations, and the energy equation. For that, ϕ must be chosen as 1, u, v, w, and T, respectively. Density and velocity must remain active in the continuity equation[17] to solve both incompressible and compressible flows. Through the use of the SIMPLEC method[18] for the pressure–velocity coupling, the continuity equation is used for the calculation of pressure, the equation of state is used to obtain the density, and the three momentum equations plus the energy equation are used to obtain the other quantities (u, v, w, and T). A collocated variable arrangement is employed in the present work. Further details of the numerical methodology used here can be seen in Refs. 2 and 3.

The solution procedure that will be described next assumes a single grid block. The necessary modifications in this procedure to accommodate a multiblock strategy will be discussed in the next section. Once initial values for the six state variables are known, the solution procedure adopted in the present work takes the following steps: 1) estimation of the u, v, w, p, T, and ρ fields at instant $t + \Delta t$; 2) computation of the coefficients for the three momentum equations, 3) computation of the coefficients for the continuity equation, 4) computation for the source terms for u, v, and w, 5) solution of the momentum equations (this step determines new velocity components u^*, v^*, and w^*, which do not necessarily conserve the mass), 6) evaluation of the contravariant velocity components U^*, V^*, and W^*, 7) computation of the error, or the residue, in the continuity equation using the available contravariant velocity components and density field, 8) evaluation of a correction to the pressure field, using the coefficients in step 3 and the residues determined in step 7, 9) correction of velocity components and densities by the new pressure field (the resulting fields conserve mass), 10) computation of the coefficients and source terms for the energy equations, 11) calculation of a new temperature field, 12) computation of the density as function of pressure and temperature, and 13) return to step 1 and iteration until the steady state is reached.

The solution process, as presented above, does not involve any iteration cycle within each time interval. However, because of the type of linearization adopted and because of the coupling scheme implemented, some steps must be executed more than once within each time step. In the present work, the computations associated with steps 3 through 9 were usually executed twice for each time interval. This inner iteration cycle, which has no meaning for incompressible problems, allows the use of larger time steps. Finally, it should be emphasized that steps 5, 8, and 11 involve the solution of linear systems.

Multiblock Technique Implementation

The multiblock implementation used in the present work was actually developed in two phases. The initial phase assumed that there would be a perfect match of the boundary control volumes on two adjacent grid blocks. Later, this was extended to the general case in which the sizes of the control volumes in two adjacent blocks do not have any specific relation between them. Moreover, it is important to emphasize that the present approach assumes no overlap of adjacent grid blocks. In other words, adjacent grid blocks simply touch each other at their common interface. Discussion of the simpler case is presented first, because it is instrumental in understanding the procedure adopted for the general case. It should also be emphasized that the important point to be discussed is the transferring of information from one grid block to its neighbor, since within each grid block the solution process is as previously described.

The present finite volume approach uses fictitious, or slave, control volumes to implement boundary conditions in each grid block. These control volumes are located outside the computational domain of interest (for each block) and serve the sole purpose of implementing boundary conditions. Hence, the previous statement that grid blocks do not overlap needs some additional explanation. The actual control volumes do not overlap, but the fictitious volumes of one grid block do overlap a few actual volumes of the adjacent grid block. Thus, when solving for a given grid block, the information on its fictitious volumes at an internal grid-block boundary comes from this overlapping. Moreover, this explains the distinction previously made between the two phases of the work reported in this paper. In the initial phase, there is a perfect match between the fictitious

volumes of one grid block with the actual boundary volumes of the adjacent one. In the general case, such nice coincidence does not occur, and one must use some form of interpolation to obtain the properties of the fictitious control volume from those of the actual control volumes in the adjacent block that have some overlap with the fictitious volume.

The procedure for information transfer in the case of a perfect match between adjacent control volumes (initial phase) is based solely on the direction of the flow at the interface. If the contravariant velocity component at the interface is positive (that is, the flow is leaving the grid block being currently solved at that interface), property values at the interface are set equal to their values at the center of the interior volume. Therefore, in this case, the solution in the current grid block is independent of the solution in the adjacent grid block. However, since the numerical method being used needs equations for the calculation of the fictitious volume, the properties of the fictitious volume are simply set equal to those of the interior volume. The opposite situation occurs when the contravariant velocity component at the interface is negative. In this case, the interface is an entrance boundary in which all flow variables are prescribed. The difference between this boundary and, for example, a freestream boundary is that the properties do not receive constant values. However, their values are determined from the corresponding current values of the properties in the control volume of the adjacent grid block. Hence, u, v, w, T, ρ, and p of the fictitious volume corresponding to the grid block being solved are set equal to their respective values in the control volume of the adjacent block. Moreover, properties at the interface itself are also calculated using the values of the control volume at the adjacent block, and of course, the metric terms of the interface.

When the control volumes on both sides of the interface do not have the same dimensions, it is necessary to determine which real control volumes of the adjacent block surround the centroid of the fictitious volume of the grid block currently being solved. Once this has been done, the procedure is similar to what we have described for the case with coincident volumes. The difference, now, is that properties at the centroid of the fictitious volume must be determined by an appropriate interpolation of the values associated with the centroids of the control volumes of the adjacent block that have some overlap with the fictitious volume considered. If the contravariant velocity component at the interface is positive, interface properties are set equal to their corresponding values at the interior volume. If the contravariant velocity component at the interface is negative, one must first perform the interpolation previously discussed, and then determine the interface values following a similar procedure as discussed for the case of coincident volumes.

It is important to observe that, in the three-dimensional case and considering the general case discussed, typically there are eight control volumes involved in this interpolation process. Hence, the computational time involved in finding the control volumes that surround the centroid of the fictitious volume and performing the trilinear interpolation described can be significant. Therefore, an approximate procedure for transferring information was also implemented in which the property values at the centroid of the fictitious volume are taken to be equal to their corresponding values at the actual volume of the adjacent block whose centroid is the closest to the centroid of the fictitious volume. This is clearly introducing an approximation, which, however, becomes less serious as the mesh is refined. Tests performed comparing the correct treatment of the interface and the approximate one here described have indicated that, in most cases, the results are extremely similar. Obviously, the approximate treatment has a lower computational cost.

The algorithm for the numerical solution of the problem with the multiblock implementation is actually quite similar to the one that has been previously described. The difference is that the advance of the iteration scheme in time is done by performing a few iterations in each block, transferring the information on this block to its neighbors, and then moving on to the next block until the complete computational domain is advanced the same number of iterations. Then we return to the first block and repeat the whole procedure until a convergence criterion is satisfied for all blocks. It was found that three to five iterations in each block, before transferring the information to the adjacent blocks, seems to be a good compromise. It is worth mentioning that this multiblock approach is extremely useful not only for the case of very complex configurations, but also for the case in which the available computational resources do not have enough central memory to accommodate the complete grid (or all the grid blocks) in core. Hence, only the grid block can be worked on in central memory. Therefore, from a computational efficiency point of view, it can be extremely helpful to perform several iterations on one grid block before having to write this block's information on disk and reading in the next block's information. Of course, this can only be done if this procedure does not seriously deteriorate the overall convergence rate of the algorithm. Finally, the authors refer the interested reader to the work in Ref. 7 for further details of the present implementation of the multiblock technique.

Comparison of Computational Performance

The results to be reported in the present paper were all fully converged to engineering accuracy. However, the convergence criterion varied between the two codes discussed. The quantity monitored in the central-difference code to determine convergence is the maximum density residue. Experience has shown that a four-order-of-magnitude drop in the maximum density residue is enough for convergence to plotting accuracy, and this has been the criterion adopted for the central-difference code calculations. Typically, some 2500–3000 iterations are required to achieve convergence with this code for a transonic case. The convergence criterion adopted for the segregated finite volume code monitors the normalized maximum pressure variation in the field between two consecutive iterations. Typically, one requires this variation to be less than 10^{-4} before accepting convergence. Transonic calculations usually need of the order of 300 iterations to achieve convergence according to the above criterion.

The codes discussed in this paper were run on a Convex C-210 computer and on an HP-Apollo DN-1000 workstation. The CPU-time comparisons to be presented will refer to Convex running times. The central-difference code runs at 7.99×10^{-5} s per iteration per grid point for a Euler calculation. The segregated all-speed code, already with the multiblock implementation, runs at 24.8×10^{-5} s per iteration per grid point for an inviscid calculation. Therefore, a typical transonic case needs of the order of 32 min to converge with the all-speed code running in Euler mode and using a $60 \times 24 \times 18$ point grid. For a similar grid, the central-difference code would need approximately 100 min to achieve convergence. Therefore, it is clear that the segregated all-speed code is at least 3 times faster than the central-difference code for a comparable problem, besides having the multiblock capability that the central difference code does not have. Moreover, the segregated all-speed code can also be run in Navier–Stokes mode, though its computational performance is then somewhat degraded, as one should expect. For a mesh that has 1.5 times more grid points than the previously cited mesh, which is essentially a comparable mesh except with further refinement in the wall-normal direction to capture viscous terms, the all-speed code needs approximately 2.5 h to achieve convergence for a laminar calculation. All codes were run using double precision in the Convex.

Finally, the subject of computational accuracy must be addressed. For finite volume codes, the order of accuracy of the solutions is mostly associated with the form in which properties are interpolated at volume interfaces for the flux calculations. The present segregated all-speed code uses an adaptive expression for this interpolation, which is dependent on the cell Peclet number. Therefore, this interpolation essentially yields a second-order accurate formula in viscous-dominated regions, whereas it reduces to a fully upwind interpolation, and hence first-order accuracy, in convection-dominated regions. For the high-speed flows considered in the present work, the all-speed method should be first-order accurate throughout most of the flowfield. However, it seems to produce very good results for subsonic and supersonic flow cases, as the forthcoming results show. The central-difference code is second-order accurate in space everywhere. Both codes are first-order accurate in time. However, since only steady-state calculations are sought, this is of no concern.

Some Validation Results

The calculations performed for the VLS so far have only considered the central body. Therefore, none of the computational results to be presented here will include the effect of the boosters. When comparing these computations with the experimental data, precautions were taken to avoid comparisons in regions in which the properties at the central body are affected by the presence of the boosters. Hence, the computations are mainly concerned with the forebody portion of the vehicle.

Figure 5 presents pressure coefficient distributions for the VLS obtained with the centered finite difference algorithm previously discussed. These results are presented for comparison purposes, since the major interest in this work is to discuss results using the segregated finite volume algorithm with the multiblock implementation in comparison with experimental data. As indicated in Fig. 5, the freestream conditions considered in these computations were $M_\infty = 0.5$ and 3.0. The two cases assumed a 0-deg angle of attack. The computational meshes used in these calculations were generated algebraically for one axisymmetric longitudinal plane, and then rotated 360 deg around the body. Therefore, the finite difference calculations use periodic boundary conditions in the circumferential direction. This is in contrast with the finite volume calculations, to be presented next, which use symmetry boundary conditions in the circumferential direction. The computational meshes used for these simulations had 63 × 34 × 26 points in the longitudinal, normal, and circumferential directions, respectively. It is clear from Fig. 5 that both subsonic and supersonic cases are in good agreement with the experimental data. In the subsonic case, we also observe an overexpansion of the flow in the forward cone–cylinder intersection. Experience with axisymmetric calculations has indicated that this has a tendency to occur with Euler simulations with the present method, but that the overexpansion typically disappears when viscous terms are included in the formulation.

Pressure distribution comparisons for the segregated finite volume algorithm were performed in the Mach-number range $0.5 \leq M_\infty \leq 3.0$, and considering angles of attack of 0, 2, and 6 deg. Some representative results of the calculations performed with this code are presented in Figs. 6–8. The leeside pressures are shown in these figures. Similar agreement is obtained for other azimuthal planes. The computational meshes used for these calculations had 48 × 70 × 12 control volumes in the longitudinal, normal, and circumferential directions, respectively. It should be emphasized that the computational domain in the longitudinal direction extends further downstream in the finite difference calculations, which justifies the difference in the number of longitudinal grid points from that in the present calculations. On the other hand, the computations presented in Figs. 6–8 show results for viscous simulations, which accounts for the marked increase in the number of grid points in the wall-normal direction. Good agreement is seen between computational and experimental results for both subsonic ($M_\infty = 0.5$) and supersonic ($M_\infty = 3.0$) flow cases. Moreover, it is evident from Figs. 6 and 8 that the relevant flow features are correctly captured by the simulation.

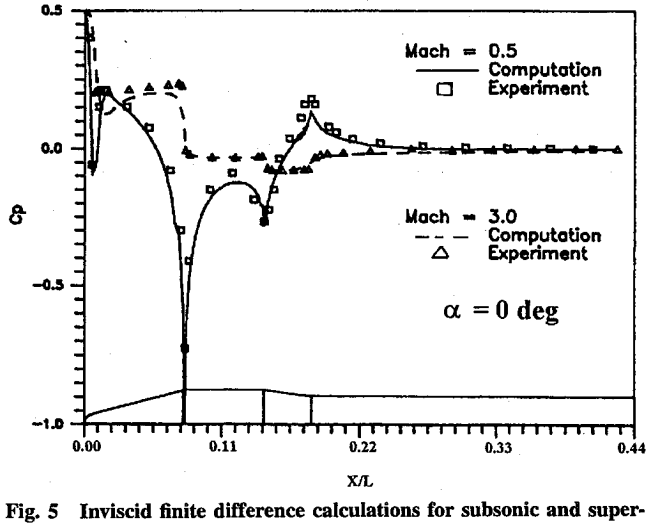


Fig. 5 Inviscid finite difference calculations for subsonic and supersonic freestream conditions.

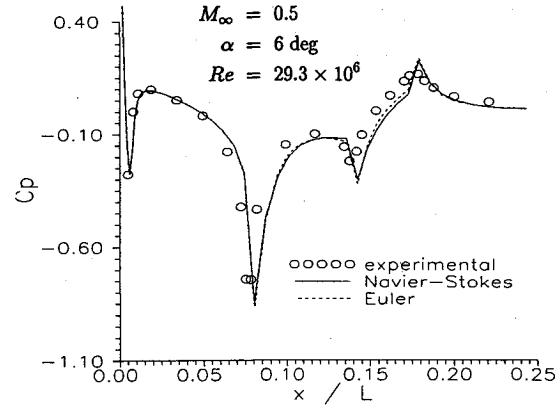


Fig. 6 Leeside pressure coefficient distributions.

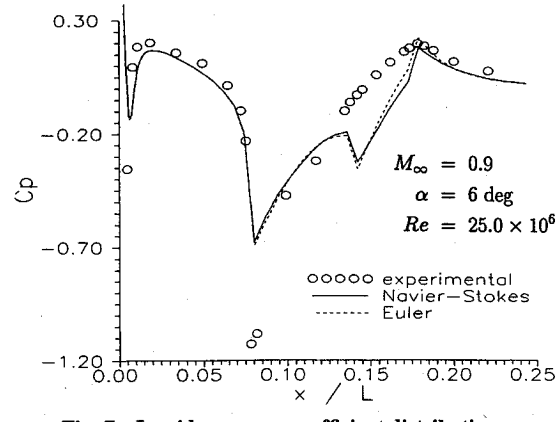


Fig. 7 Leeside pressure coefficient distributions.

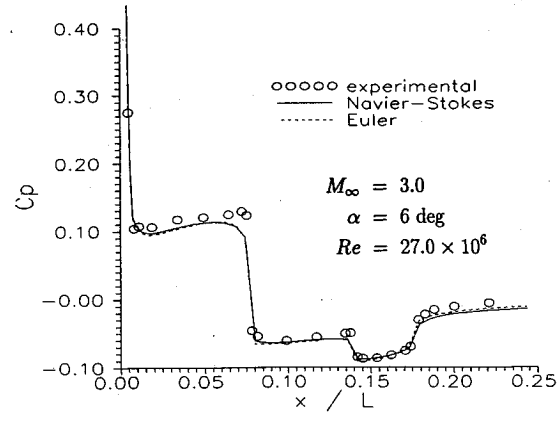


Fig. 8 Leeside pressure coefficient distributions.

The agreement in the transonic case, shown in Fig. 7, is much poorer. Of special concern is the fact that near $x/L = 0.15$, which is the location of the forebody cylinder–boattail intersection, both viscous and inviscid simulations predict a local expansion of the flow, whereas the experimental results show no indication of such behavior. Videotapes of the wind-tunnel tests indicate that for $M_\infty = 0.9$ there is a rather strong shock wave impinging upon the vehicle's forebody cylindrical section. This effect is clearly indicated by the experimental pressure coefficient distribution shown in Fig. 7. Therefore, the current explanation for the difference in pressure distribution behavior around $x/L = 0.15$ is that the present implementation of the code is unable to capture the flow separation that occurs at the impingement point of this transonic shock. Hence, the computation can "see" the expansion at the forebody cylinder–boattail intersection, whereas the actual flow is separated and does not see the expansion.

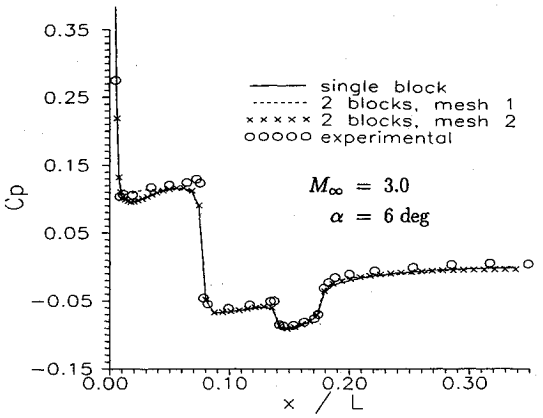


Fig. 9 Comparison of leeside pressure coefficient distributions with multiblock implementation.

It should be emphasized that the present simulations consider laminar flow, since a turbulence model has not been implemented into the code. Moreover, the experimental tests were performed for a Reynolds number of the order 10^7. Hence, there is no question that the experiment has turbulent flow. For this reason, the validation effort in the present work has concentrated in the comparison of the pressure distributions. As the results indicate, the implemented capability seems to provide good agreement with the experimental data, except for the transonic flow case. The implementation of appropriate turbulence models into the present codes is an activity that the authors will pursue in the future to enhance the computational simulation capability for these large launchers, but that is beyond the intended scope of the present work. Nevertheless, at the present stage of development, the code already gives very useful information for preliminary design studies.

Inviscid results using the multiblock technique are presented in Fig. 9 for freestream Mach number 3.0 with 6-deg angle of attack. Two different computational meshes were used for these simulations, besides the single block grid. Each of these meshes had two blocks: an inner block close to the body surface and an outer block. Mesh 1 had $60 \times 10 \times 18$ volumes in the inner block and $16 \times 14 \times 18$ volumes in the outer block. Mesh 2 had $60 \times 14 \times 18$ in the inner block and $60 \times 10 \times 18$ in the outer block. The single block grid is formed simply by the addition of these two grid blocks in a single mesh. The leeside pressure distributions shown in Fig. 9 indicate that the multiblock calculations agree well with the single block solution. The differences observed in mesh 1 in the forebody cone region is caused by a block interface being located in a region of very high gradients. As would be expected, this interface seems to be causing a degradation in the accuracy of the information transferring from one block to the other. The experimental results shown for comparison in Fig. 9 correspond to flow at $Re = 27.0 \times 10^6$. However, the computational results shown in this case consider only the Euler equations.

Concluding Remarks

The development of an all-speed Euler and/or Navier–Stokes flow simulation code that uses a segregated finite volume algorithm for three-dimensional body-conforming curvilinear coordinates with a collocated variable arrangement has been described. The efforts towards the physical validation of this code are the major contribution of the present work. The details of an experimental wind-tunnel investigation for pressure distribution and force measurements for the Brazilian VLS are also described. The results obtained for subsonic and supersonic flow conditions are in very good agreement with the experimental data. Transonic calculations, however, show much poorer agreement with data for equivalent and even more refined meshes. Recent calculations for transonic nozzle applications indicate that the flow solution methodology used in the present code needs severe refinement in the longitudinal direction to truly capture the rapid flow gradients present in transonic flow conditions, at least as compared to conventional central-difference type algorithms. Despite some of its difficulties with transonic flow, the flow simulation capability implemented into the present code already gives very useful information for preliminary design studies, which is actually being used at present in the VLS design process. Moreover, efforts are also under way to use the present multiblock implementation to provide the first three-dimensional computational results on the VLS vehicle together with its first-stage boosters.

References

[1]Maliska, C. R., and Silva, A. F. C., "A Boundary-Fitted Finite Volume Method for the Solution of Compressible and/or Incompressible Fluid Flows Using both Velocity and Density Corrections," *Finite Elements in Fluids*, edited by T. J. Chung and G. Karr, Huntsville Press, 1989, pp. 257–266.

[2]Marchi, C. H., Maliska, C. R., and Bortoli, A. L., "The Use of Co-Located Variables in the Solution of Supersonic Flows," *Proceedings of the 10th Brazilian Congress of Mechanical Engineering* (Rio de Janeiro, Brazil), Vol. 1, 1989, pp. 157–160.

[3]Marchi, C. H., Maliska, C. R., and Silva, A. F. C., "A Boundary-Fitted Numerical Method for the Solution of Three Dimensional All Speed Flows Using Co-Located Variables," *Proceedings of the 3rd Brazilian Thermal Sciences Meeting* (Itapema, SC, Brazil), Vol. 1, 1990, pp. 351–356.

[4]Azevedo, J. L. F., "Aerodynamic Flow Simulation Using a Finite Difference Method," *Proceedings of the 2nd National Meeting of the Thermal Sciences* (Águas de Lindóia, SP, Brazil), 1988, pp. 3–6.

[5]Azevedo, J. L. F., Zdravistch, F., and Silva, A. F. C., "Implementation and Validation of Euler Solvers for Launch Vehicle Flows," *Proceedings of the 4th International Symposium on Computational Fluid Dynamics* (Davis, CA), Vol. 1, 1991, pp. 42–47.

[6]Moraes, P., Jr., and Neto, A. A., "Aerodynamic Experimental Investigation of the Brazilian Satellite Launch Vehicle (VLS)," *Proceedings of the 3rd Brazilian Thermal Sciences Meeting* (Itapema, SC, Brazil), Vol. 1, 1990, pp. 211–215.

[7]Marchi, C. H., Maliska, C. R., and Silva, A. F. C., "Numerical Solution of Flows over Complex Geometries Using the Multiblock Technique," *Proceedings of the 4th Brazilian Thermal Science Meeting* (Rio de Janeiro, Brazil), 1992, pp. 353–356 (in Portuguese).

[8]Moraes, P., Jr., and Neto, A. A., "High Speed Aerodynamic Wind Tunnel Tests for the VLS," Instituto de Aeronáutica e Espaço, Report NT-036/ETP-AED/89, São José dos Campos, SP, Brazil, March 1989 (in Portuguese).

[9]Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, McGraw–Hill, New York, 1984.

[10]Pulliam, T. H., and Steger, J. L., "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, No. 2, 1980, pp. 159–167.

[11]Pulliam, T. H., and Steger, J. L., "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms," AIAA Paper 85-0360, Jan. 1985.

[12]Beam, R. M., and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," *Journal of Computational Physics*, Vol. 22, Sept. 1976, pp. 87–110.

[13]Beam, R. M., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier–Stokes Equations," *AIAA Journal*, Vol. 16, No. 4, 1978, pp. 393–402.

[14]Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, McGraw–Hill, New York, 1980.

[15]Karki, K. C., and Patankar, S. V., "Pressure-Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations," *AIAA Journal*, Vol. 27, No. 9, 1989, pp. 1167–1174.

[16]Maliska, C. R., Silva, A. F. C., and Marchi, C. H., "Co-Located Variable, Three-Dimensional, Euler and Navier–Stokes Formulation for All Speed Flows," Instituto de Aeronáutica e Espaço, Rept. NT-093/ETP-AED/91, São José dos Campos, SP, Brazil, March 1991 (in Portuguese).

[17]Van Doormaal, J. P., "Numerical Methods for the Solution of Incompressible and Compressible Fluid Flows," Ph.D. Thesis, Univ. of Waterloo, ON, Canada, 1985.

[18]Van Doormaal, J. P., and Raithby, G. D., "Enhancements of the Simple Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, Vol. 7, Feb. 1984, pp. 147–163.

Jerry Allen
Associate Editor