NASA/TM–2011-217181

Inflow/Outflow Boundary Conditions with Application to FUN3D

Jan-Reneé Carlson
Langley Research Center, Hampton, Virginia

October 2011

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at 443-757-5803

- Phone the NASA STI Help Desk at 443-757-5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/TM–2011-217181

Inflow/Outflow Boundary Conditions with Application to FUN3D

Jan-Reneé Carlson
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

October 2011

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

Several boundary conditions that allow subsonic and supersonic flow into and out of the computational domain are discussed. These boundary conditions are demonstrated in the FUN3D computational fluid dynamics (CFD) code which solves the three-dimensional Navier-Stokes equations on unstructured computational meshes. The boundary conditions are enforced through determination of the flux contribution at the boundary to the solution residual. The boundary conditions are implemented in an implicit form where the Jacobian contribution of the boundary condition is included and is exact. All of the flows are governed by the calorically perfect gas thermodynamic equations. Three problems are used to assess these boundary conditions. Solution residual convergence to machine zero precision occurred for all cases. The converged solution boundary state is compared with the requested boundary state for several levels of mesh densities. The boundary values converged to the requested boundary condition with approximately second-order accuracy for all of the cases.

1 Introduction

THE solution of a partial differential equation or a system of partial differential equations requires a statement of the dependent variables along the boundaries of the solution space. When physically relevant solutions of the equations are sought, the imposition of the boundary conditions should solve the correct numerical problem without adversely affecting solution stability.

The choice between an explicit or implicit implementation of the boundary conditions in the solution algorithm is dependent on the numerical method that is used for solving the governing equations. Explicit treatments of boundary conditions has occupied much of the literature. The compilation of papers in the proceedings *Numerical Boundary Condition Procedures* [1] contains extensive discussions of the state-of-the-art theories at the time of printing. In particular, Moretti [2], Blottner [3], Pulliam [4], and Thompkins and Bush [5] discuss inflow and outflow boundary conditions for both Navier-Stokes and Euler equations.

Much of that work centered around developing a system of equations, often in terms of characteristic variables, to describe the desired physical state in a consistent form at the boundary. Information is communicated through the boundary interface via wave propagation and convection. The flow of information across the boundary determines the physical conditions to impose. Thus, the boundary conditions must be formulated to keep the solution realizable at the boundary. The interior solution will be consistent with the specified physical state at all of the boundaries once an iteratively converged solution has been achieved.

Figure 1 is a schematic of a boundary where the physical state is on the right side and the solution space is on the left. Also shown are the upstream and downstream propagating Riemann invariants, R^- and R^+. The convention in FUN3D [6] is that the normal vector at a boundary, $\hat{n} = \vec{n}/|\vec{n}|$, is outwardly directed from the interior. The velocity perpendicular to the boundary, $U_\perp$, is defined by the scalar product of the local velocity vector with the boundary normal, $U_\perp = \overrightarrow{U} \cdot \hat{n}$, where $\overrightarrow{U} = u\hat{i} + v\hat{j} + w\hat{k}$.

Designated by the sign and magnitude of $U_\perp$, the flow conditions at the boundary are listed in Table 1. The Euler equations have five eigenvalues—three that are associated with convective waves, λ_{2-4}, and two that are associated with acoustic waves, $\lambda_{1,5}$. A positive eigenvalue corresponds to a wave that is entering the domain and that conveys physical information specified from the outside (i.e., the boundary condition). A negative eigenvalue is a wave that is leaving the solution domain. As discussed in Kim et al. [7], various conditions for a boundary are summarized in Table 1. For example, for the specification

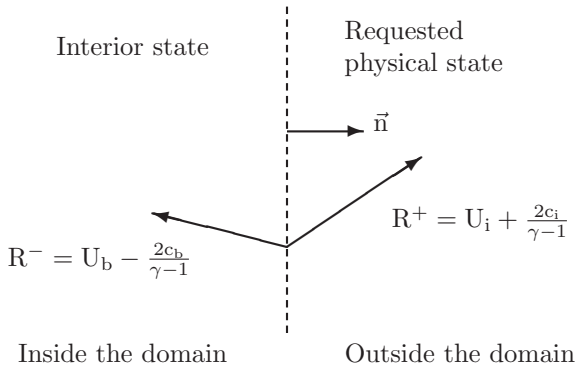


Figure 1. **Characteristics at a boundary.**

of a *subsonic inflow* boundary with the use of primitive variables, density and velocity or density and pressure are combinations that can be used to completely define the condition. In practice, many methods are used to enforce a particular condition at a boundary interface. In this work, the boundary conditions are enforced weakly, that is, by calculating and applying the flux contribution of the boundary condition to the solution residual rather than setting the physical value of the boundary condition.

The details of each boundary condition are discussed in a generic framework in section 2; and a discussion of the discrete, computational method used for evaluation of the boundary conditions in FUN3D follows in section 3. Three test geometries are described in section 4. Computations that demonstrate the use of the boundary conditions are given in section 5.

Table 1. **Flow Conditions at a Boundary**

Case	**Condition**	$U_\perp$	$\lvert U_\perp \rvert$	**Eigenvalue**					**Specify**
				λ_1	λ_2	λ_3	λ_4	λ_5	
1	Subsonic inflow	<0	$<c$	<0	>0	>0	>0	>0	p_t, T_t, α, β^*
2	Subsonic outflow	>0	$<c$	<0	<0	<0	<0	>0	p
3	Supersonic inflow	<0	$>c$	>0	>0	>0	>0	>0	All
4	Supersonic outflow	>0	$>c$	<0	<0	<0	<0	<0	None

*Alternatively ρ and $\overrightarrow{U}$ or ρ and p, but not $\overrightarrow{U}$ and p.

2 Boundary Conditions

To evaluate the fluxes at the boundary, information from the interior of the domain is combined with the physical constraints of the problem. As shown in Figure 2, the flux at the boundary depends on a combination of the left and right states ($\boldsymbol{q}_L$ and $\boldsymbol{q}_R$) and the direction, $\hat{n}$, and area, A, of the boundary ($\boldsymbol{\mathcal{F}} = f(\hat{n}, A, \boldsymbol{q}_L, \boldsymbol{q}_R)$). The left state is equal to the interior state, $\boldsymbol{q}_i$, and the right state is a function of the interior state, $\boldsymbol{q}_i$, the free-stream, $\boldsymbol{q}_\infty$, and/or user-specified parameters, $\boldsymbol{\mathcal{B}}$, depending on the boundary condition.

The linearization of the flux function for fixed grids, is written as the change in the flux function, $\boldsymbol{\mathcal{F}}$, with respect to the state vector, $\boldsymbol{q}$, on the left side of the boundary:

$$\left.\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{q}}\right|_L = \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{q}_L} + \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{q}_R}\frac{\partial \boldsymbol{q}_R}{\partial \boldsymbol{q}_L} \tag{1}$$

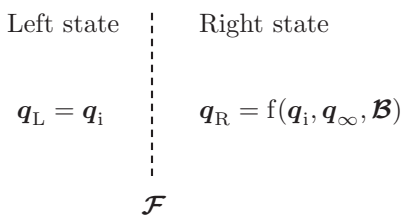


Figure 2. **Flux and boundary states.**

The first two derivatives, $\partial\boldsymbol{\mathcal{F}}/\partial\boldsymbol{q}_{\mathrm{L}}$ and $\partial\boldsymbol{\mathcal{F}}/\partial\boldsymbol{q}_{\mathrm{R}}$, are the Jacobians for the numerical flux function used in the spatial differencing. FUN3D provides multiple options for the numerical flux function. The results in this work use Roe's approximate flux scheme and the Jacobian terms of Roe's scheme were handcoded term-by-term. The third term, $\partial\boldsymbol{q}_{\mathrm{R}}/\partial\boldsymbol{q}_{\mathrm{L}}$, is determined exactly by using automatic differentiation via operator overloading of the equations in the boundary condition functions in Griewank and Corliss [8, 9]. This differentiation technique, which is based on the chain rule, permits the partial derivatives of the boundary condition variables with respect to the interior variables to be calculated and propagated through each of the boundary condition function calls and results in the Jacobian term of the boundary conditions.

Depending upon the specific application, the four inflow/outflow conditions listed in Table 1 (that is, subsonic inflow/outflow and supersonic inflow/outflow) can be formed in different ways. Several methods for determining the boundary state, $\boldsymbol{q}_{\mathrm{R}}$, are discussed in sections 2.1 through 2.9. For each boundary condition, the format of the introductory table is as follows: the type of condition, the specified conditions, variables that are extrapolated from the interior, and variables that are updated as a consequence of applying the boundary condition. Words expressed with the `fixed-width font` indicate code variables that occur within FUN3D.

Variables with dimensions are denoted by an overhead ˜ symbol. Non-dimensional variables are denoted without any symbol overhead. Freestream values are denoted by the subscript ∞. The equations relating the dimensional to the non-dimensional flow-field variables are as follows:

$$\tilde{\rho} = \rho\tilde{\rho}_{\infty}, \quad \tilde{\mathrm{u}} = \mathrm{u}\,\tilde{\mathrm{c}}_{\infty}, \quad \tilde{\mathrm{v}} = \mathrm{v}\,\tilde{\mathrm{c}}_{\infty}, \quad \tilde{\mathrm{w}} = \mathrm{w}\,\tilde{\mathrm{c}}_{\infty}, \quad \tilde{\mathrm{p}} = \mathrm{p}\,\tilde{\rho}_{\infty}\tilde{\mathrm{c}}_{\infty}^2 \tag{2}$$

$$\tilde{\mathrm{e}} = \mathrm{e}\,\tilde{\rho}_{\infty}\,\tilde{\mathrm{c}}_{\infty}^2, \quad \tilde{\mathrm{c}} = \mathrm{c}\,\tilde{\mathrm{c}}_{\infty}, \quad \tilde{\mathrm{T}} = \mathrm{T}\,\tilde{\mathrm{T}}_{\infty} = \mathrm{c}^2\,\tilde{\mathrm{T}}_{\infty} \tag{3}$$

$$\rho_{\infty} = 1, \quad \mathrm{p}_{\infty} = 1/\gamma, \quad \mathrm{c}_{\infty} = 1, \quad \mathrm{T}_{\infty} = 1 \tag{4}$$

$$\mathrm{u}_{\infty} = \mathrm{M}_{\infty}\cos\alpha_{\infty}\cos\beta_{\infty}, \quad \mathrm{v}_{\infty} = -\mathrm{M}_{\infty}\sin\beta_{\infty}, \quad \mathrm{w}_{\infty} = \mathrm{M}_{\infty}\sin\alpha_{\infty}\cos\beta_{\infty} \tag{5}$$

The equations in the subsequent sections are written in their non-dimensional form.

2.1 Far-Field Boundary Condition (`farfield_roe`)

Type	**Specify**	**Extrapolate**	**Update**
Outflow	M_{∞}, $[\mathrm{M}_{\infty} \geq 0]$, α_{∞}, β_{∞}	None	All

The far-field boundary condition specifies the free-stream conditions that are calculated from the user input variables M_{∞}, α_{∞}, and β_{∞}.

$$\rho_{\infty} = 1, \quad \mathrm{c}_{\infty} = 1, \quad \mathrm{p}_{\infty} = \frac{\rho_{\infty}\mathrm{c}_{\infty}^2}{\gamma}, \quad \overrightarrow{\mathrm{U}}_{\infty} = \begin{bmatrix} \mathrm{M}_{\infty}\cos\alpha_{\infty}\cos\beta_{\infty} \\ -\mathrm{M}_{\infty}\sin\beta_{\infty} \\ \mathrm{M}_{\infty}\sin\alpha_{\infty}\cos\beta_{\infty} \end{bmatrix} \tag{6}$$

The right-state vector is:

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \rho_\infty \\ \overrightarrow{\mathrm{U}}_\infty \\ \mathrm{p}_\infty \end{bmatrix} \tag{7}$$

2.2 Riemann Invariant Boundary Condition (`riemann`)

Type	Specify	Extrapolate	Update
Inflow/outflow	M_∞ $[\mathrm{M}_\infty \geq 0]$	Entropy (outflow)	ρ, p

The Riemann invariants correspond to the incoming R^- and outgoing R^+ characteristic waves. The invariants determine the locally normal velocity component and the speed of sound. The entropy, s, and the speed of sound, c, are used to determine the density and pressure on the boundary (see Figure 1.) The incoming Riemann invariant uses the far-field conditions, $\boldsymbol{q}_{\mathrm{o}} = \left[\rho_\infty, \overrightarrow{\mathrm{U}}_\infty, \mathrm{p}_\infty\right]^{\mathrm{T}}$.

$$\mathrm{R}^+ \;=\; \mathrm{U_i} + \frac{2\mathrm{c_i}}{\gamma - 1}, \quad \mathrm{R}^- = \mathrm{U_o} - \frac{2\mathrm{c_o}}{\gamma - 1} \tag{8}$$

where

$$\mathrm{U_i} \;=\; \overrightarrow{\mathrm{U}}_{\mathrm{i}} \cdot \hat{\mathrm{n}}, \quad \overrightarrow{\mathrm{U}}_{\mathrm{i}} = \left[\mathrm{u_i}, \mathrm{v_i}, \mathrm{w_i}\right]^{\mathrm{T}}, \quad \mathrm{c_i^2} = \frac{\gamma \mathrm{p_i}}{\rho_{\mathrm{i}}} \tag{9}$$

$$\mathrm{U_o} \;=\; \overrightarrow{\mathrm{U}}_{\mathrm{o}} \cdot \hat{\mathrm{n}}, \quad \overrightarrow{\mathrm{U}}_{\mathrm{o}} = \left[\mathrm{u_o}, \mathrm{v_o}, \mathrm{w_o}\right]^{\mathrm{T}}, \quad \mathrm{c_o^2} = \frac{\gamma \mathrm{p_o}}{\rho_{\mathrm{o}}} \tag{10}$$

$$\mathrm{M_i} \;=\; \frac{|\mathrm{U_i}|}{\mathrm{c_i}} \tag{11}$$

If the flow at the boundary is locally supersonic leaving the domain ($\mathrm{M_i} \geq 1$), then no incoming characteristic waves exist; thus, R^- is set equal to

$$\mathrm{R}^- \;=\; \mathrm{U_i} - \frac{2\mathrm{c_i}}{\gamma - 1} \tag{12}$$

Similarly, if the flow is supersonic entering the domain, then no outgoing characteristic waves exist and R^+ is set equal to

$$\mathrm{R}^+ \;=\; \mathrm{U_o} + \frac{2\mathrm{c_o}}{\gamma - 1} \tag{13}$$

A velocity, $\mathrm{U_b}$, and the speed of sound, $\mathrm{c_b}$, at the boundary are the sum and difference of the invariants:

$$\mathrm{U_b} \;=\; \frac{1}{2}\left(\mathrm{R}^+ + \mathrm{R}^-\right), \quad \mathrm{c_b} = 4\left(\gamma - 1\right)\left(\mathrm{R}^+ - \mathrm{R}^-\right) \tag{14}$$

The velocity that is imposed on the boundary depends on the local direction of flow. If the sign of $\mathrm{U}_\perp$ is positive, then the flow is exiting the computational domain and the entropy is extrapolated from the interior and is used to update the density at the boundary. Conversely, a negative $\mathrm{U}_\perp$ indicates that the flow is entering the computational domain and

the free-stream entropy is used. Summarizing, the velocity and entropy on the boundary are calculated from the following equations.

$$\overrightarrow{\mathrm{U}}_\mathrm{b} = \begin{cases} \overrightarrow{\mathrm{U}}_\mathrm{i} + (\mathrm{U} - \mathrm{U}_\mathrm{i})\hat{\mathrm{n}}, & \text{if } \mathrm{U} > 0 \text{ (outflow)} \\ \overrightarrow{\mathrm{U}}_\mathrm{o} + (\mathrm{U} - \mathrm{U}_\mathrm{o})\hat{\mathrm{n}}, & \text{if } \mathrm{U} \le 0 \text{ (inflow)} \end{cases} \tag{15}$$

$$s_\mathrm{b} = \begin{cases} \frac{\mathrm{c}_\mathrm{i}^2}{\gamma \rho_\mathrm{i}^{\gamma-1}}, & \text{if } \mathrm{U}_\perp > 0 \text{ (outflow)} \\ \frac{\mathrm{c}_\mathrm{o}^2}{\gamma \rho_\mathrm{o}^{\gamma-1}}, & \text{if } \mathrm{U}_\perp \le 0 \text{ (inflow)} \end{cases} \tag{16}$$

The density and pressure on the boundary are then calculated as follows.

$$\rho_\mathrm{b} = \left(\frac{\mathrm{c}_\mathrm{b}^2}{\gamma s_\mathrm{b}} \right)^{1/\gamma - 1}, \quad \mathrm{p}_\mathrm{b} = \frac{\rho_\mathrm{b} \mathrm{c}_\mathrm{b}^2}{\gamma} \tag{17}$$

The right-side vector is

$$\boldsymbol{q}_\mathrm{R} = \begin{bmatrix} \rho_\mathrm{b} \\ \overrightarrow{\mathrm{U}}_\mathrm{b} \\ \mathrm{p}_\mathrm{b} \end{bmatrix}. \tag{18}$$

2.3 Outflow Mach-Number Boundary Condition (`subsonic_outflow_mach`)

Type	Specify	Extrapolate	Update
Outflow	M $[0 \le \mathrm{M} < 1]$	T	ρ, p

The Mach number at the boundary, M_set, is specified for the entire boundary, and the flow is assumed to be adiabatic and isentropic. The local acoustic speed is determined from the temperature of the interior state, $\boldsymbol{q}_\mathrm{i}$, with the local speed of sound determined using Eq. (19).

$$\mathrm{c}_\mathrm{i} = \sqrt{\mathrm{T}_\mathrm{i}}, \quad \mathrm{T}_\mathrm{i} = \frac{\gamma \mathrm{p}_\mathrm{i}}{\rho_\mathrm{i}} \tag{19}$$

For each point on the boundary, the Mach number, M_i, is used to determine a new total pressure, $\mathrm{p}_{\mathrm{t,b}}$.

$$\mathrm{M}_\mathrm{i} = \frac{|\mathrm{U}_\mathrm{i}|}{\mathrm{c}_\mathrm{i}}, \quad \mathrm{U}_\mathrm{i} = \overrightarrow{\mathrm{U}}_\mathrm{i} \cdot \hat{\mathrm{n}} \tag{20}$$

$$\mathrm{p}_{\mathrm{t,b}} = \mathrm{p}_\mathrm{i} \left[1 + \frac{1}{2}(\gamma - 1)\mathrm{M}_\mathrm{i}^2 \right]^{\frac{\gamma}{\gamma-1}} \tag{21}$$

The static pressure, p, is updated using the new total pressure and the set Mach number, M_set

$$\mathrm{p} = \mathrm{p}_{\mathrm{t,b}} \left[1 + \frac{1}{2}(\gamma - 1)\mathrm{M}_\mathrm{set}^2 \right]^{-\frac{\gamma}{\gamma-1}} \tag{22}$$

If the flow is locally supersonic as a result of some transient flow condition, then the boundary condition will reset the local static pressure to the local total pressure. If the specified Mach number is supersonic, then an extrapolation boundary condition should be used. Thus,

$$\mathrm{p}_\mathrm{b} = \begin{cases} \mathrm{p}, & \text{if } \mathrm{M}_\mathrm{i} < 1 \text{ (subsonic)} \\ \mathrm{p}_\mathrm{t}, & \text{if } \mathrm{M}_\mathrm{i} > 1 \text{ (supersonic)} \end{cases} \tag{23}$$

The right-state state is written as:

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \gamma \mathrm{p_b}/\mathrm{T_i} \\ \overrightarrow{\mathrm{U}}_{\mathrm{i}} \\ \mathrm{p_b} \end{bmatrix} \tag{24}$$

Note: Using this condition is inconsistent if the boundary face adjoins a viscous surface. The Mach number is fixed over the entire boundary and precludes the presence of a velocity gradient that would normally be present on the viscous no-slip surface. A constant pressure boundary condition, such as the subsonic outflow boundary, should be used in this situation.

2.4 Pressure Outflow Boundary Condition (`back_pressure`)

Type	Specify	Extrapolate	Update
Outflow	$\mathrm{p}/\mathrm{p}_\infty$ $[\mathrm{p}/\mathrm{p}_\infty > 0]$	$\overrightarrow{\mathrm{U}}$, T	ρ

For a subsonic outflow boundary condition, the static pressure ratio, $\mathrm{p_{set}}/\mathrm{p}_\infty$, is specified, while the velocities and temperature are extrapolated. The boundary flow is assumed to be adiabatic and isentropic. The density is updated from the extrapolated temperature and the requested static pressure. If the flow is supersonic at this boundary, then all quantities are extrapolated. The pressure is calculated as

$$\mathrm{p_b} = \begin{cases} \mathrm{p_{set}}, & \text{if } \mathrm{M_i} < 1 \text{ (subsonic)} \\ \mathrm{p_i}, & \text{if } \mathrm{M_i} \geq 1 \text{ (supersonic)} \end{cases} \tag{25}$$

and the right-side vector is

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \gamma \mathrm{p_b}/\mathrm{T_i} \\ \overrightarrow{\mathrm{U}}_{\mathrm{i}} \\ \mathrm{p_b} \end{bmatrix}, \quad \mathrm{T_i} = \frac{\gamma \mathrm{p_i}}{\rho_{\mathrm{i}}} \tag{26}$$

2.5 Subsonic Outflow Boundary Condition (`subsonic_outflow_p0`)

Type	Specify	Extrapolate	Update
Outflow	$\mathrm{p}/\mathrm{p}_\infty$ $[\mathrm{p}/\mathrm{p}_\infty > 0]$	$\overrightarrow{\mathrm{U}}$, T	ρ

The manner in which this boundary specifies the static pressure ratio is the same as that presented in section 2.4 but with different implementation details. If any reverse flow (i.e., flow into the computational domain) occurs, setting the static pressure at the boundary is numerically ill-posed. This boundary condition will explicitly set the flow to exit the domain. The flow is also forced to remain subsonic by setting the local static pressure to the local total pressure if the local Mach number is greater than one. The velocities, $\mathrm{U_i}$, and the temperature, $\mathrm{T_i} = \gamma \mathrm{p_i}/\rho_{\mathrm{i}}$, are extrapolated from the interior solution. The pressure is

$$\mathrm{p_b} = \begin{cases} \mathrm{p_{set}}, & \text{if } \mathrm{M_i} < 1 \text{ (subsonic)} \\ \mathrm{p_t}, & \text{if } \mathrm{M_i} > 1 \text{ (supersonic)} \end{cases} \tag{27}$$

and the velocity is

$$\overrightarrow{\mathrm{U}}_{\mathrm{b}} = \begin{cases} \overrightarrow{\mathrm{U}}_{\mathrm{i}}, & \text{if } \mathrm{U}_{\perp} > 0 \quad \text{(outflow)} \\ |\mathrm{U}_{\mathrm{i}}|\hat{\mathrm{n}}, & \text{if } \mathrm{U}_{\perp} < 0 \quad \text{(inflow)} \end{cases}, \quad \mathrm{U}_{\perp} = \overrightarrow{\mathrm{U}}_{\mathrm{i}} \cdot \hat{\mathrm{n}} \tag{28}$$

The right-side vector is

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \gamma \mathrm{p_b}/\mathrm{T_i} \\ \overrightarrow{\mathrm{U}}_{\mathrm{b}} \\ \mathrm{p_{set}} \end{bmatrix} \tag{29}$$

2.6 Mass Flow Out Boundary Condition (`massflux_out`)

Type	Specify	Extrapolate	Update
Outflow	$\dot{\mathrm{m}}$ $[\dot{\mathrm{m}} \geq 0]$	$\overrightarrow{\mathrm{U}}$, T	ρ, p

This boundary condition allows for specification of the mass flow out of the computational domain. Adiabatic flow through the boundary is assumed, and the method iteratively modifies the static pressure for the entire boundary to obtain the requested mass flow condition. The choice of equations solved for this boundary condition is from the CFD code VULCAN [10].

The boundary values of momentum thrust Eq.(30), pressure force Eq.(31), and mass flow Eq.(32), are calculated by using the following integrals, where the domain of the integration is the entire boundary face:

$$\mathrm{F_m} = \int_{\mathrm{boundary}} \rho \left(\overrightarrow{\mathrm{U}} \cdot \hat{\mathrm{n}} \right)^2 d\mathrm{A}, \quad \text{(momentum thrust)} \tag{30}$$

$$\mathrm{F_p} = \int_{\mathrm{boundary}} \mathrm{p}\, d\mathrm{A}, \quad \text{(pressure force)} \tag{31}$$

$$\dot{\mathrm{m}} = \int_{\mathrm{boundary}} \rho \left(\overrightarrow{\mathrm{U}} \cdot \hat{\mathrm{n}} \right) d\mathrm{A}, \quad \text{(mass flow)} \tag{32}$$

The static pressure is updated with a form of the continuity equation:

$$\mathrm{p_b} = \frac{1}{\mathrm{A}} \left[\mathrm{F_m} \delta \mathrm{m} + \mathrm{F_p} \right], \quad \delta \mathrm{m} = \left(1 - \frac{\dot{\mathrm{m}}_{\mathrm{set}}}{\dot{\mathrm{m}}} \right) \tag{33}$$

The velocity and the temperature are extrapolated from the interior computational domain so that the right-state vector is:

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \gamma \mathrm{p_b}/\mathrm{T_i} \\ \overrightarrow{\mathrm{U}}_{\mathrm{i}} \\ \mathrm{p_b} \end{bmatrix} \tag{34}$$

2.7 Subsonic Inflow Boundary Condition (`subsonic_inflow_pt`)

Type	Specify	Extrapolate	Update
Inflow	$\mathrm{p_t}/\mathrm{p_\infty}$, $\mathrm{T_t}/\mathrm{T_\infty}$, $\hat{\mathrm{n}}$	p, $\mathrm{R^+}$, $\mathrm{H_t}$	ρ, $\overrightarrow{\mathrm{U}}$

The user can specify the direction of the velocity at the boundary either by the flow angles (α_{set} and β_{set}) relative to the global coordinate axis, or as normal to the boundary, $\hat{\text{n}}_{\text{b}}$.

$$\hat{\text{n}} = \begin{cases} \cos\alpha_{\text{set}} \ \cos\beta_{\text{set}} \ \hat{i} - \sin\beta_{\text{set}} \ \hat{j} + \sin\alpha_{\text{set}} \ \sin\beta_{\text{set}} \ \hat{k} \\ \hat{\text{n}}_{\text{b}} \end{cases} \tag{35}$$

The right-side vector, $\boldsymbol{q}_{\text{R}}$, is determined from the outward propagating invariant, R^+, and a statement of the total enthalpy, H_{t}, at an element face on the boundary. The formulation of this boundary condition is the same as that used in [10], which assumes that the flow through the boundary is adiabatic and isentropic. Following these assumptions, we write

$$\text{H}_{\text{t}_{\text{i}}} \ = \ \frac{\text{p}_{\text{i}}}{\rho_{\text{i}}}\left(\frac{\gamma}{\gamma-1}\right) + \frac{1}{2}\left(u_{\text{i}}^2 + v_{\text{i}}^2 + w_{\text{i}}^2\right) \tag{36}$$

$$\text{R}^+ \ = \ -\text{U}_{\text{i}} - \frac{2\text{c}_{\text{i}}}{\gamma-1} \tag{37}$$

Because the flow is adiabatic, that is, the total enthalpy is conserved across the boundary, we can state that

$$\text{H}_{\text{t}} \ = \ \frac{\text{c}_{\text{b}}^2}{\gamma-1} + \frac{1}{2}\text{U}_{\text{b}}^2 \tag{38}$$

Then, by extrapolating R^- to the boundary, we obtain

$$\text{R}^+ \ = \ -\text{U}_{\text{b}} - \frac{2\text{c}_{\text{b}}}{\gamma-1} \tag{39}$$

By combining Eqs. (39) and (38), we obtain

$$\text{H}_{\text{t}} \ = \ \frac{\text{c}_{\text{b}}^2}{\gamma-1} + \frac{1}{2}\left(\text{R}^+ + \frac{2\text{c}_{\text{b}}}{\gamma-1}\right)^2 \tag{40}$$

Equation (40) is solved for c_{b}, which is the sonic speed at the boundary, by rewriting it as a quadratic equation of the form

$$\left[1 + \frac{2}{(\gamma-1)}\right]\text{c}_{\text{b}}^2 + 2\text{R}^+\text{c}_{\text{b}} + \frac{\gamma-1}{2}\left(\text{R}^{+2} - 2\text{H}_{\text{t}}\right) \ = \ 0 \tag{41}$$

The solution has the form

$$\text{c}_{\text{b}_\pm} \ = \ -\frac{b}{2a} \pm \frac{\sqrt{b^2-4ac}}{2a} \tag{42}$$

where a, b, and c are the coefficients of the quadratic equation (Eq. 41):

$$a \ = \ 1 + \frac{2}{(\gamma-1)}, \quad b = 2\text{R}^+, \quad c = \frac{\gamma-1}{2}\left(\text{R}^{+2} - 2\text{H}_{\text{t}}\right) \tag{43}$$

The physically consistent result is the larger of the two roots and is chosen to update the sonic velocity at the boundary:

$$\text{c}_{\text{b}} \ = \ \max(\text{c}_{\text{b}_+}, \text{c}_{\text{b}_-}) \tag{44}$$

The updated inflow velocity and the Mach number at the boundary are then computed by using

$$\text{U} \ = \ \frac{2\text{c}_{\text{b}}}{\gamma-1} - \text{R}^+, \quad \text{M}_{\text{b}} = \frac{\text{U}}{\text{c}_{\text{b}}} \tag{45}$$

The static pressure and temperature are calculated from the isentropic relations

$$\mathrm{p_b} \;=\; \mathrm{p_{t,set}}\left(1+\frac{\gamma-1}{2}\mathrm{M_b^2}\right)^{\frac{\gamma}{\gamma-1}}, \quad \mathrm{T_b} = \mathrm{T_{t,set}}\left(\frac{\mathrm{p_b}}{\mathrm{p_{t,set}}}\right)^{\frac{\gamma-1}{\gamma}} \tag{46}$$

The primitive variables that are determined from the input boundary conditions are placed in the right state $\boldsymbol{q}_\mathrm{R}$ as

$$\boldsymbol{q}_\mathrm{R} \;=\; \begin{bmatrix} \mathrm{p_b}/R\mathrm{T_b} \\ \mathrm{U}\hat{\mathrm{n}} \\ \mathrm{p_b} \end{bmatrix} \tag{47}$$

2.8 Mass Flow In Boundary Condition (`massflux_in`)

Type	Specify	Extrapolate	Update
Inflow	$\dot{\mathrm{m}}$ [$\dot{\mathrm{m}} \geq 0$], $\mathrm{T_t/T_\infty}$	ρ	p

Mass flow into the computational domain through this boundary condition is updated by adjusting the static temperature, T, at the boundary through a form of the energy equation:

$$\mathrm{c_p T} + \frac{\overline{\mathrm{U}}^2}{2} \;=\; \mathrm{c_p T_{t,set}} \tag{48}$$

The variable, $\mathrm{T_{t,set}}$, in Eq. (48) is the user-specified total temperature at the boundary, and the flow is assumed to be isentropic and adiabatic through the boundary face. An expression for the velocity is derived by dividing the specified mass flow by the density and can be written as

$$\overline{\mathrm{U}} \;=\; \frac{\dot{\mathrm{m}}}{\bar{\rho}\mathrm{A}}, \quad \bar{\rho} = \frac{\int_{\mathrm{boundary}} \rho \; d\mathrm{A}}{\mathrm{A}} \tag{49}$$

A quadratic equation of static temperature is written by combining Eq. (49) with Eq. (48) and using a thermally perfect gas assumption for density in terms of the static pressure and temperature, $\bar{\rho} = \bar{\mathrm{p}}/R\mathrm{T}$:

$$\frac{1}{2}\left(\frac{\dot{\mathrm{m}}R}{\bar{\mathrm{p}}\mathrm{A}}\right)^2 \mathrm{T}^2 + \mathrm{c_p T} - \mathrm{c_p T_{t,set}} = 0 \tag{50}$$

The solution to the resulting equation is a quadratic equation

$$\mathrm{T}_\pm = -\frac{b}{2a} \pm \frac{\sqrt{b^2-4ac}}{2a} \tag{51}$$

where the coefficients of the quadratic equation are

$$a \;=\; \frac{1}{2}\left(\frac{\dot{\mathrm{m}}R}{\bar{\mathrm{p}}A}\right)^2, \quad b = \mathrm{c_p}, \quad c = -\mathrm{c_p T_{t,set}} \tag{52}$$

The larger root is taken to update the static temperature at the boundary as

$$\mathrm{T_b} \;=\; \max(\mathrm{T_+}, \mathrm{T_-}) \tag{53}$$

To lessen transients during the solution startup, the user can ramp the mass inflow, $\dot{\mathrm{m}}$, from zero up to the specified amount by using the iteration parameter `flow_mflux_ramp` as

$$\dot{\mathrm{m}}_{\mathrm{ramp}} = \min\left(1, \frac{\texttt{iteration}}{\texttt{flow\_mflux\_ramp}}\right) \mathrm{m}_{\mathrm{set}} \tag{54}$$

The continuity equation determines the mean velocity at the boundary, and the density is updated from the updated static temperature as

$$\mathrm{U} = \frac{\mathrm{T_b}}{\gamma} \frac{\dot{\mathrm{m}}_{\mathrm{ramp}}}{\bar{\mathrm{p}} A} \tag{55}$$

The right-state vector is:

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \rho_{\mathrm{i}} \\ -\mathrm{U}\hat{\mathrm{n}} \\ \rho_{\mathrm{i}} \mathrm{T_b}/\gamma \end{bmatrix} \tag{56}$$

2.9 Supersonic Inflow Boundary Condition (`fixed_inflow`)

Type	Specify	Extrapolate	Update
Inflow	ρ, $\overrightarrow{\mathrm{U}}$, p	None	All

Pressure, density, and velocity ($\overrightarrow{\mathrm{U}}_{\mathrm{set}}$) are specified for the supersonic inflow boundary condition. The velocity vector is forced to be normal to the boundary. The right-state vector is:

$$\boldsymbol{q}_{\mathrm{R}} = \begin{bmatrix} \rho_{\mathrm{set}} \\ -|\mathrm{U}_{\mathrm{set}}|\hat{\mathrm{n}} \\ \mathrm{p}_{\mathrm{set}} \end{bmatrix} \tag{57}$$

3 Computational Method

3.1 FUN3D Code

FUN3D is an unstructured three-dimensional, implicit, Navier-Stokes code. Roe's flux difference splitting [11] is used for the calculation of the explicit terms. Other available flux construction methods include HLLC [12], AUFS [13], and LDFSS [14]. The default method for calculation of the Jacobians is the flux function of van Leer [15], but the method by Roe and the HLLC, AUFS and LDFSS methods are also available. The use of flux limiters are mesh and flow dependent. Flux limiting options include MinMod [16] and methods by Barth and Jespersen [17] and Venkatakrishnan [18]. Other details regarding FUN3D can be found in Anderson and Bonhaus [6] and Anderson et al. [19], as well as in the extensive bibliography that is accessible at the FUN3D Web site, `http://fun3d.larc.nasa.gov`.

3.2 Boundary Element Discretization

Discretization of the computational volume consists of any combination of tetrahedra, hexahedra, prisms, and pyramids. The faces of the volume elements are either triangles or quadrilaterals. Therefore, the boundaries consist only of either triangular or quadrilateral faces or combinations of both types of face elements. An example of a boundary face that

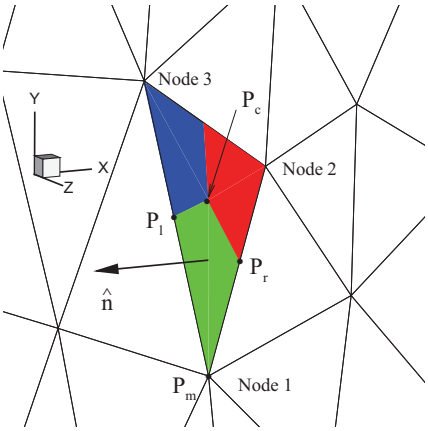


Figure 3. **Perspective view of triangular boundary element in Y-Z plane.**

Table 2. **Boundary Element Node Weight Factors**

Topology	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_3$
Tetrahedral	6/8	1/8	1/8
Hexahedral, prism, pyramid	1	0	0

consists of triangular face elements is shown in the perspective sketch in Figure 3. A space of subelements is formed to calculate the fluxes at the boundary. Three subelements are formed from each of the corner nodes (1, 2, and 3) of the triangular face element. The area and the orientation of the quadrilateral subelements are determined by using the center of the element , P_c; the left and right midpoints, P_l and P_r; and the main node, P_m. The normal vector is the cross product of the two vectors that are formed from the main and center nodes, $\overrightarrow{P_m P_c}$, and the left and right midpoints, $\overrightarrow{P_l P_r}$, so that $\vec{n} = \overrightarrow{P_m P_c} \times \overrightarrow{P_l P_r}$.

The interior, or left state, for the subelement is calculated from an area weighted average of the interior state nodal values, $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and $\boldsymbol{q}_3$. The values of the area weights, $\mathcal{W}$, change with the element topology and are listed in Table 2. A detailed derivation of the boundary node weighting for tetrahedral meshes can be found in appendix A.

$$\boldsymbol{q}_{\mathrm{L}} \;=\; \mathcal{W}_1 \boldsymbol{q}_1 + \mathcal{W}_2 \boldsymbol{q}_2 + \mathcal{W}_3 \boldsymbol{q}_3 \tag{58}$$

A flux function calculates the contribution to the solution residual for the subelement from the interior and right-side (i.e., boundary condition) states. The inviscid flux is added to the solution residual at each solution time step. Roe's approximate Riemann solver was used for the boundary flux contributions to the residual.

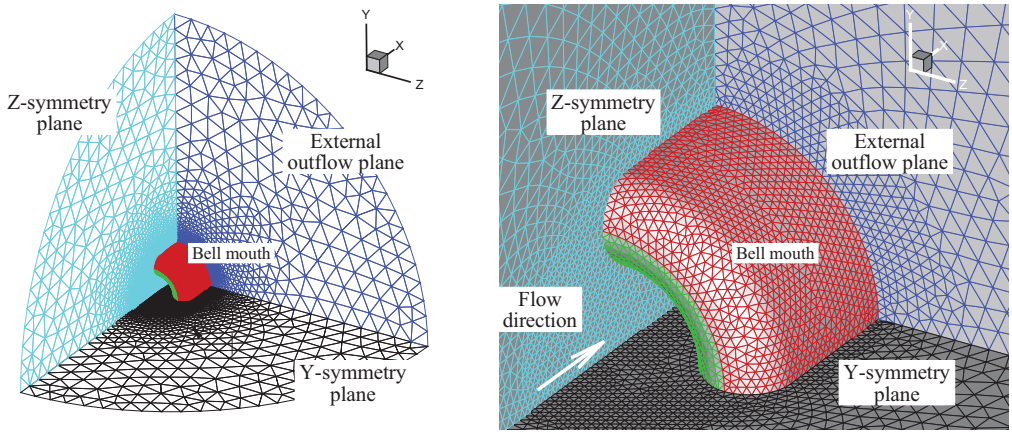
4 Test Case Descriptions

The three geometries that are described in Sections 4.1 through 4.3 are used to demonstrate the inflow and outflow boundary conditions that were described in Section 2. The geometries are representative of a typical physical situation that could be modeled with each of the different boundary conditions. Static pressure and mass flow outflow subsonic conditions are applied to the outflow boundary of the bell-mouth geometry, Section 4.1. Total pressure-total temperature conditions and mass flow inflow conditions are applied to the inflow boundary of the American Society Of Mechanical Engineers (ASME) nozzle geometry,

Section 4.2. Fixed primitive variable conditions that simulate a supersonic inflow are applied to the inflow boundary of the diffuser geometry, Section 4.3.

4.1 Generic Bell-Mouth

The generic bell-mouth configuration is a geometry that transitions an external flow to an internal flow over a range of conditions without flow separation [20]. The bell-mouth geometry is routinely used at experimental facilities as a flow-conditioning device for testing inlets or propulsion simulators. Figure 4(a) shows a perspective view of the geometry with the symmetry planes, the external outflow plane, and the external surface of the bell-mouth. A detailed view of the external bell-mouth surface from the same perspective view is shown

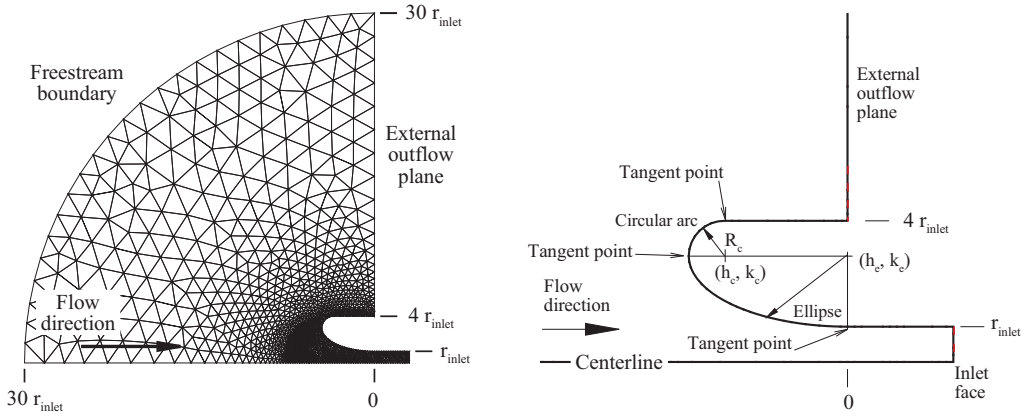


(a) **Perspective view of the geometry.**

(b) **Detailed view of external bell-mouth geometry.**

Figure 4. **Bell-mouth geometry with representative surface mesh.**

in Figure 4(b). The direction of the flow is from left to right. The radius of the inlet is 1.0, and the far-field boundary is 30 inlet radii upstream, as shown in Figure 5(a). The shape

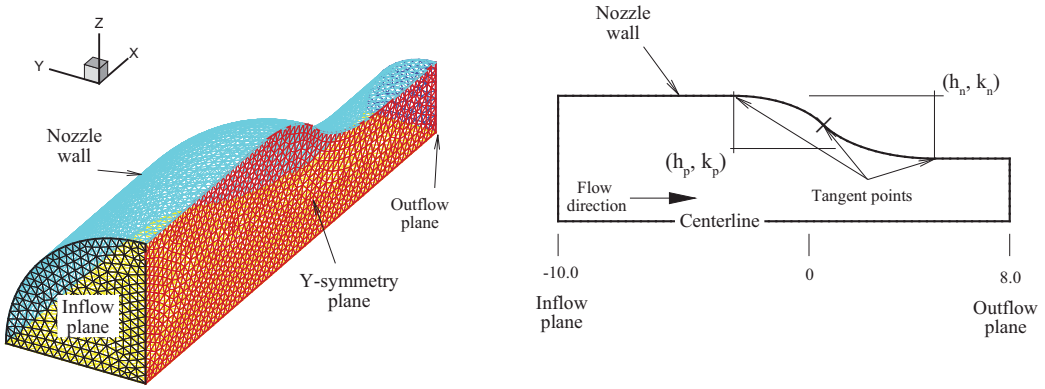


(a) **Symmetry plane view.**

(b) **Geometric definition of the bell-mouth.**

Figure 5. **Bellmouth geometry details.**

of the bell-mouth consists of an elliptic contraction section with a 9:1 capture area. (See



(a) **Perspective view of the nozzle.** (b) **Geometric definition of the nozzle.**

Figure 6. **ASME nozzle geometry.**

Figure 5(b).) The bell fairs to a flat external wall via a circular-arc lip. For this study, the geometry is axisymmetric and is modeled with a quarter-plane symmetric unstructured tetrahedral mesh. The mathematical equation that describes the elliptic portion of the bell-mouth is

$$\frac{(x-\mathrm{h_e})^2}{\mathrm{a_e^2}}+\frac{(y-\mathrm{k_e})^2}{\mathrm{b_e^2}}=1,\quad \mathrm{h_e}=0.0,\quad \mathrm{k_e}=3.0,\quad \mathrm{a_e}=4.5,\quad \mathrm{b_e}=2.0 \tag{59}$$

and the equation that describes the circular-arc fairing is

$$(x-\mathrm{h_c})^2+(y-\mathrm{k_c})^2=\mathrm{R_c^2},\quad \mathrm{h_c}=-\mathrm{a_e}+\mathrm{R_c},\quad \mathrm{k_c}=\mathrm{k_e},\quad \mathrm{R_c}=1.0 \tag{60}$$

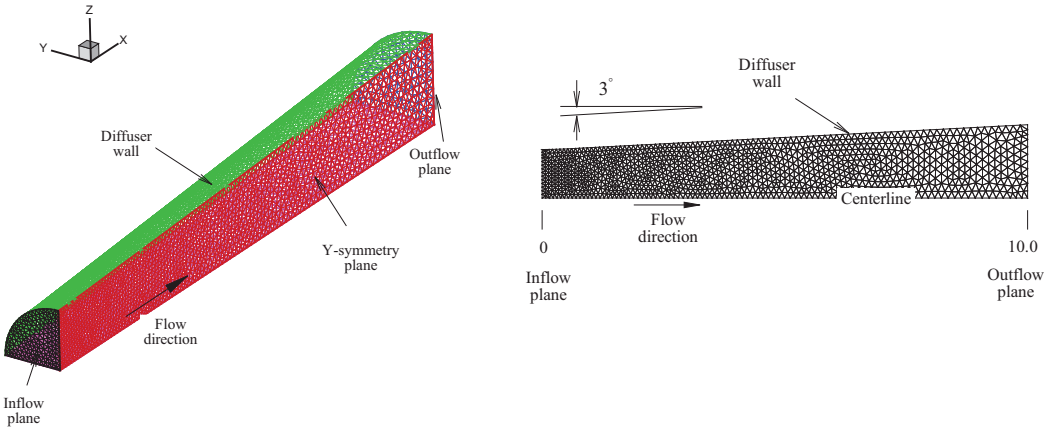
4.2 ASME Flow Calibration Nozzle

The ASME nozzle is a flow standard geometry that is used for calibration of full-pipe flows in experimental facilities [21] and serves as an excellent configuration for validation of internal flow calculations. A perspective view of the mesh is shown in Figure 6(a), and a streamwise cross-sectional view of the nozzle is shown in Figure 6(b). The geometry consists of two constant-area cylindrical sections joined via two elliptically contoured sections. The upstream plenum section has a cross-sectional radius of 5 in., and the downstream nozzle section is a constant-area pipe with a radius of 2.5 in. The plenum entrance is at station $x=-10.0$, with the exit at station $x=8.0$. The fairing tangent point between the two elliptical contours is at $x\approx 0.54$. All of the computations were accomplished by using quarter-plane symmetric tetrahedral meshes. The equation that defines the nozzle wall contour from the middle tangent point to the downstream tangent point is

$$\frac{(x-\mathrm{h_n})^2}{\mathrm{a_n^2}}+\frac{(y-\mathrm{k_n})^2}{\mathrm{b_n^2}}=1,\quad \mathrm{h_n}=5.0,\quad \mathrm{k_n}=2.5,\quad \mathrm{a_n}=5.0,\quad \mathrm{b_n}=2.5 \tag{61}$$

and the equation that defines the nozzle wall from the upstream tangent point to the middle tangent point is

$$\frac{(x-\mathrm{h_p})^2}{\mathrm{a_p^2}}+\frac{(y-\mathrm{k_p})^2}{\mathrm{b_p^2}}=1,\quad \mathrm{h_p}=-3.0,\quad \mathrm{k_p}=2.5,\quad \mathrm{a_p}=3.99,\quad \mathrm{b_p}=2.9 \tag{62}$$



(a) **Perspective view of the diffuser.** (b) **Geometric definition of the diffuser.**

Figure 7. **Conical diffuser geometry.**

Table 3. **Test Case Mesh Densities**

h_N	Bell-mouth	ASME nozzle	Diffuser
	tetrahedra/nodes	**tetrahedra/nodes**	**tetrahedra/nodes**
1.0	51,078 / 10,253	50,309 / 10,202	25,484 / 5,670
0.8	99,125 / 19,326	97,517 / 19,129	52,499 / 10,992
0.6	235,111 / 44,370	231,510 / 43,760	132,693 / 26,452
0.4	775,258 / 141,372	764,440 / 139,535	433,863 / 82,184
0.2	6,648,653 / 1,168,680	6,011,766 / 1,059,652	2,271,735 / 491,521

4.3 Supersonic Diffuser

The third test geometry is a conical diffuser with a wall divergence angle of 3 degrees. The increasing cross-sectional area of the duct maintains the supersonic flow condition downstream of the supersonic fixed inflow boundary. A perspective view of the geometry is shown in Figure 7(a). A streamwise cross-sectional view of the geometry is shown in Figure 7(b).

5 Discussion

The unstructured meshes with tetrahedral cells were generated using the software package VGRID [22]. The number of nodes for the cases is changed through the use of the global parameter, `ifact`. The `ifact` parameter controls the density and number of cells, N, by increasing or decreasing the strength of the sources within the mesh generation data file. The total number of cells (and nodes) for each case are listed in Table 3.

An equivalent mesh size can be related to the mesh density (or number of degrees of freedom) N. The equivalent mesh size should decrease with an increase in the number of cells. In three dimensions, the equivalent mesh size, h_N, should tend to zero as $N^{-1/3}$. The equivalent mesh size can also be related to a characteristic distance defined in terms of the

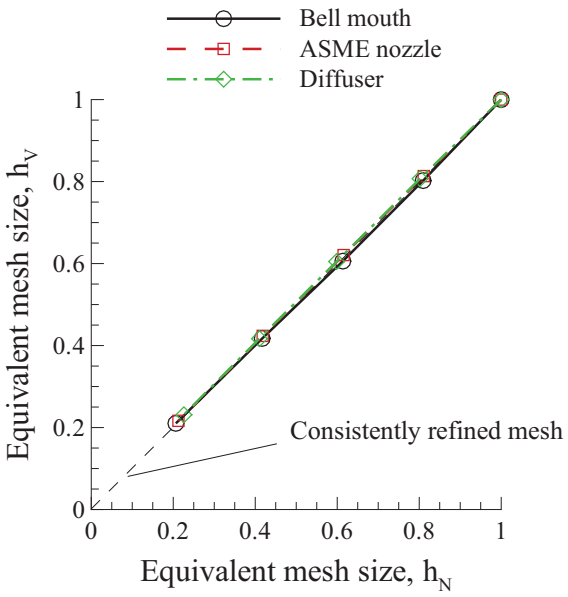


Figure 8. **Mesh characteristics.**

Table 4. **Test Cases**

Boundary condition	**Boundary name**	**Geometry**	**Specified conditions**
Pressure outflow	`back_pressure`	Bell-mouth	p/p_∞
Pressure outflow	`subsonic_outflow_p0`	Bell-mouth	p/p_∞
Mach number outflow	`subsonic_outflow_mach`	Bell-mouth	M
Mass flow outflow	`massflux_out`	Bell-mouth	$\dot{m}$
Pressure subsonic inflow	`subsonic_inflow_pt`	ASME nozzle	p_t/p_∞, T_t/T_∞ , α, β
Mass flow inflow	`massflux_in`	ASME nozzle	$\dot{m}$, T_t/T_∞
Supersonic inflow	`fixed_inflow`	Diffuser	ρ, $\vec{U}$, p

local control volume size, $h_v = \|V^{-1/3}\|$, where the norm used in this study is the L_∞ norm.

A statement of the asymptotic order of error can only be made when consistently refined meshes are used. Consistent mesh refinement is purely geometric occurring when a linear relationship exists between the equivalent mesh size based on degrees of freedom ,h_N, and the equivalent mesh size based on the characteristic distance, h_V. See Thomas *et al.* [23] for additional discussion on consistent refinement. The similarity of the mesh families for each configuration is shown in Figure 8. A consistently refined mesh is shown as a dashed line in the Figure. All three geometries display a linear trend between the equivalent mesh measures, h_N and h_V.

The free-stream Mach number, unless otherwise noted, was 0.2 and the Courant-Friedrichs-Lewy (CFL) number, unless otherwise noted, was ramped from 1 to 100 over 250 iterations. Flux construction was performed by using the approximate Riemann solver by Roe and without any flux limiting. The boundary condition test cases are listed in Table 4. The bell-mouth was used to test the static pressure and the mass flow out boundary conditions. The ASME nozzle was used to test boundary conditions with the total temperature and either total pressure or mass flow set at an inflow boundary. The diffuser geometry was used

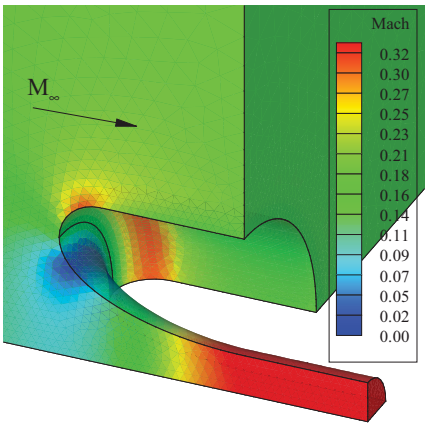


Figure 9. **Mach number contours for the bell-mouth geometry.**

to test the supersonic fixed inflow boundary condition. In the following sections, solution convergence histories for each of the mesh densities are shown for each boundary condition. Also, the convergence of the error in the boundary value is shown for each boundary condition. The error in the boundary condition is the difference between the user-defined boundary condition and the area-weighted average of the boundary solution condition from the code. The area-weighted average for a variable, ϕ, is designated by an overbar $\overline{\phi}$ and is calculated as $\int_{\text{boundary}} \phi \text{dA} / \int_{\text{boundary}} \text{dA}$.

The area-weighted average values for ρ, p, p_t, and mass flow are calculated by the summation equations, Eqs. (63). The average velocity, speed of sound, and Mach number are calculated by Eqs. (64).

$$\overline{\rho} = \frac{1}{\text{A}} \sum_{\text{boundary}} \rho \, \delta\text{A}, \quad \overline{\text{p}} = \frac{1}{\text{A}} \sum_{\text{boundary}} \text{p} \, \delta\text{A}, \quad \overline{\text{p}}_\text{t} = \frac{1}{\text{A}} \sum_{\text{boundary}} \left(\text{p} + \frac{1}{2} \rho \text{U}^2 \right) \delta\text{A} \tag{63}$$

$$\overline{\dot{\text{m}}} = \sum_{\text{boundary}} \rho \text{U}_\perp \delta\text{A}, \quad \overline{\text{U}} = \overline{\dot{\text{m}}} / \overline{\rho}\text{A}, \quad \overline{\text{c}} = \sqrt{\gamma \overline{\text{p}} / \overline{\rho}}, \quad \overline{\text{M}} = \overline{\text{U}} / \overline{\text{c}} \tag{64}$$

where δA is the area of an individual element on the boundary face and A is the boundary area, $\text{A} = \sum_{\text{boundary}} \delta\text{A}$. The error in the boundary condition for the parameter ϕ is then

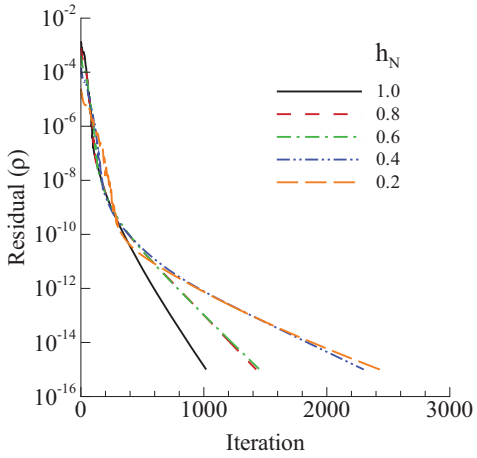
$$\text{Error}_\phi = \phi_{\text{set}} - \overline{\phi} \tag{65}$$

5.1 Bell-Mouth Calculations

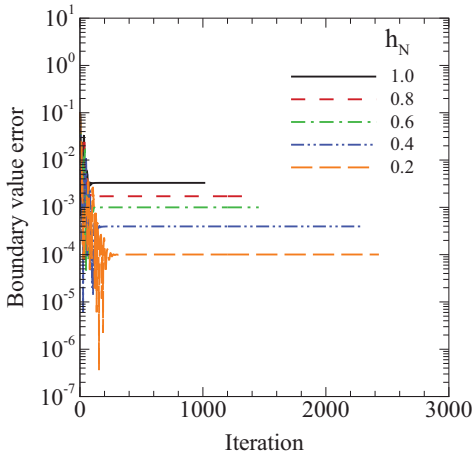
The bell-mouth geometry was used to test the fixed subsonic Mach number, fixed static pressure, and mass flow outflow boundary conditions. Surface Mach number contours on and around the bell-mouth for an outflow mass flow condition of 0.25 and a free-stream Mach number of 0.2 are shown in Figure 9. For this set of conditions, the flow stagnation point is slightly below the leading edge of the bell-mouth, as evidenced by the region of zero Mach number contours. The flow accelerates externally around the bell mouth leading edge to exit the computational domain further downstream.

5.1.1 Static Pressure Outflow (`back_pressure`)

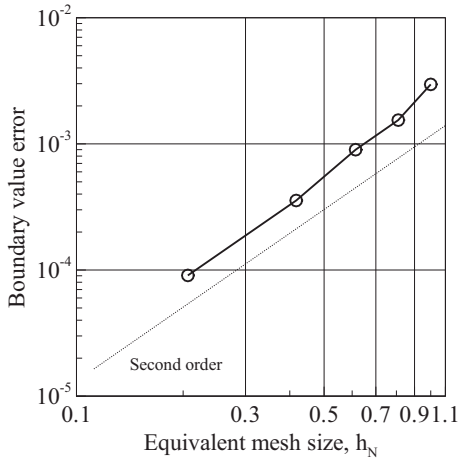
Type	Specify
Outflow	$p/p_\infty = 0.90$, $M_\infty = 0.2$



(a) **Residual history.**



(b) **Boundary value error history.**



(c) **Boundary value error property.**

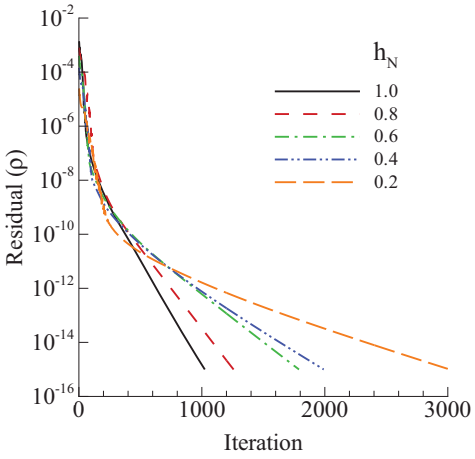
Figure 10. **Static pressure ratio boundary condition,** `back_pressure`**.**

The solution residual history for the fixed static pressure (also known as a back pressure) test is shown in Figure 10(a). Iterative convergence was achieved for all mesh levels for the set condition of $p/p_\infty = 0.9$. The boundary value error typically reached the final level for each equivalent mesh size in fewer than 400 iterations, Figure. 10(b). The final boundary value error is typically achieved before interative convergence of the solution residual. The change in boundary value error with equivalent mesh size after iterative convergence is shown in Figure 10(c). The error in the set static pressure converges as the square of the equivalent

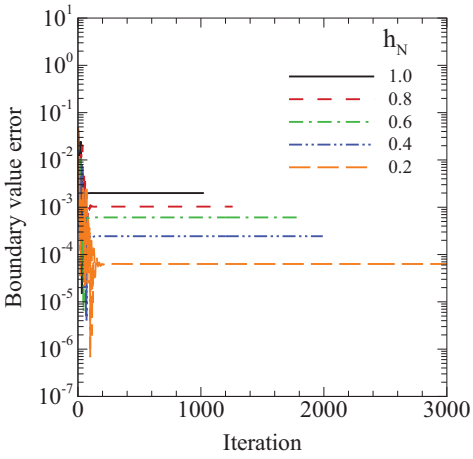
mesh size. The second-order convergence property in the error is indicated by the slope of the dotted line.

5.1.2 Subsonic Static Pressure Outflow (`subsonic_outflow_p0`)

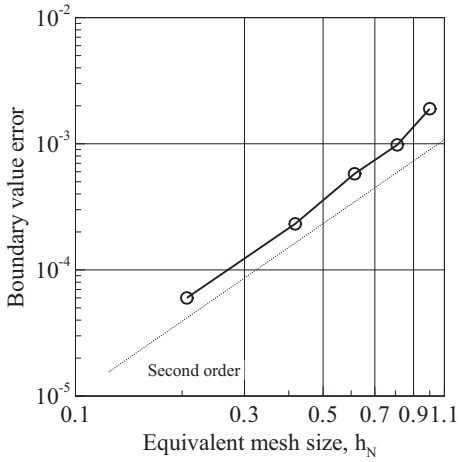
Type	Specify
Outflow	$p/p_\infty = 0.90$, $M_\infty = 0.2$



(a) **Residual history.**



(b) **Boundary value error history.**



(c) **Boundary value error property.**

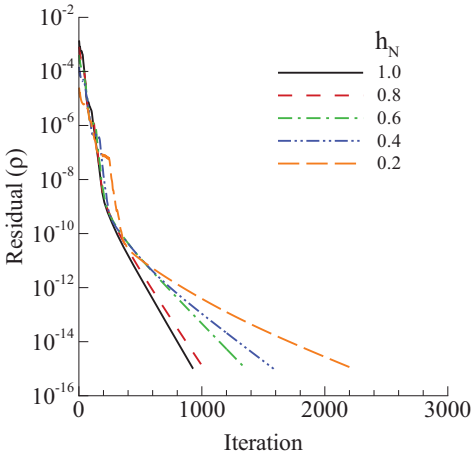
Figure 11. **Subsonic outflow boundary condition,** `subsonic_outflow_p0`**.**

The residual history for subsonic outflow static pressure calculations is shown in Figure 11(a). Similar to the back pressure boundary, all mesh levels achieve iterative convergence for the set condition of $p/p_\infty = 0.9$. The boundary value error achieved the final value for each mesh level typically before 400 iterations, Figure 11(b). The change in boundary value error with equivalent mesh size after iterative convergence is shown in Figure 11(c).

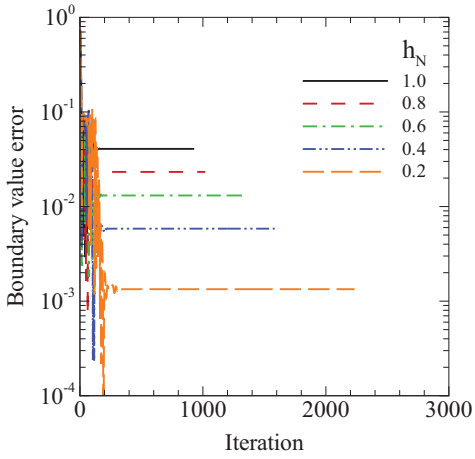
The error in the set static pressure converges as the square of the equivalent mesh size. The second-order convergence property in the error is indicated by the slope of the dotted line.

5.1.3 Mach-number outflow (`subsonic_outflow_mach`)

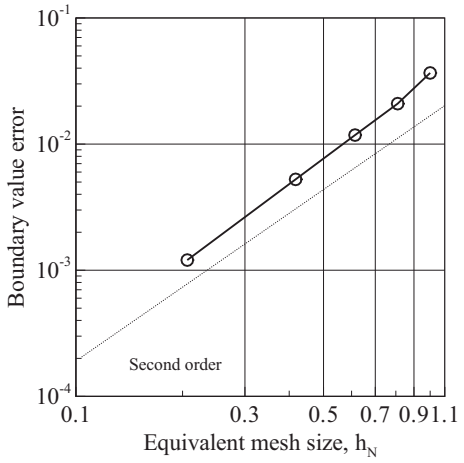
Type	Specify
Outflow	$M_{set} = 0.90$, $M_\infty = 0.2$



(a) **Residual history.**



(b) **Boundary value error history.**



(c) **Boundary value error property.**

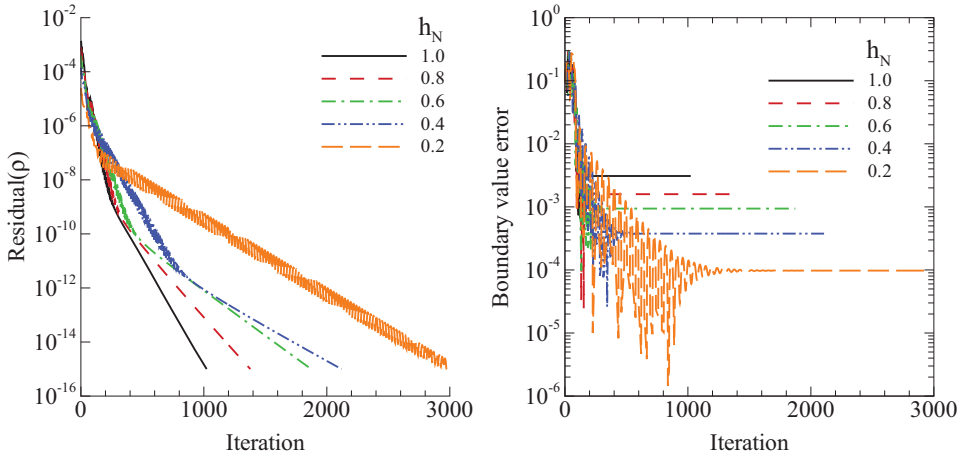
Figure 12. **Mach-number boundary condition,** `subsonic_outflow_mach`**.**

The Mach-number outflow boundary condition adjusts the static pressure to obtain the specified Mach number. The residual history for the fixed subsonic Mach-number outflow boundary condition test is shown in Figure 12(a). Iterative convergence was achieved for all mesh levels for the set condition of $M_{set} = 0.9$. As shown in Figure 12(b), the error in

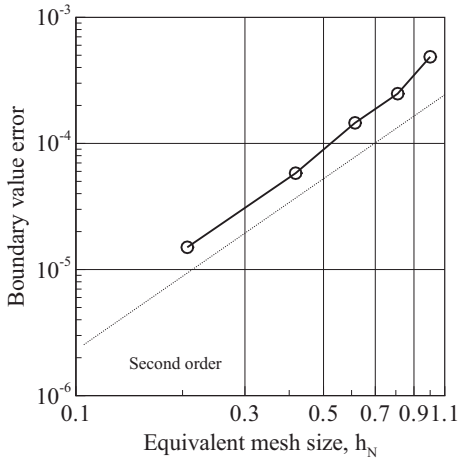
the Mach number reaches the final value within several hundred iterations for this test case. The error in the set Mach number converges as the square of the equivalent mesh size. The second-order convergence property in the error is indicated by the slope of the dotted line (Figure 12(c)). For this configuration, M_{set}= 0.9 is approximately equivalent to a inflow static pressure ratio of $p/p_\infty = 0.61$.

5.1.4 Mass flow outflow (`massflux_out`)

Type	Specify
Outflow	$\dot{m} = 0.25$, $M_\infty = 0.2$



(a) **Residual history.** (b) **Boundary value error history.**



(c) **Boundary value error property.**

Figure 13. **Mass flow out boundary condition,** `massflux_out`**.**

The residual history and the boundary value error history for the fixed mass flow outflow calculation are shown in Figure 13(a). Iterative convergence was achieved for all mesh levels

for the set condition of $\dot{m} = 0.25$. As shown in Figure 13(b), the error in the mass flow reaches the final value within several hundred iterations for most cases. The relaxation of the static pressure to set the mass flow took longer for the equivalent size mesh $h_N = 0.2$ as a result of a lower damping of the pressure oscillations in the flow field. The error in the set mass flow converges as the square of the equivalent mesh size. The second-order convergence property in the error is indicated by the slope of the dotted line (Figure 13(c)). The average values for the Mach number and the static pressure ratio at the boundary for this mass flow setting are approximately 0.33 and 0.95, respectively.

5.2 ASME Nozzle Calculations

Two inflow boundary conditions are evaluated in this section using an ASME flow calibration nozzle. Figure 14 shows Mach-number contours for choked-flow conditions in the nozzle. The direction of flow, from left to right, is denoted by the symbol U and the vector arrow. Typical combinations of inflow and outflow boundaries for an internal flow configuration includes (a) total pressure–total temperature inflow with static pressure outflow; (b) total pressure–total temperature inflow with mass flow outflow; or (c) mass flow inflow and extrapolation outflow. Choice (c) would work for choked flows where the exit Mach number is equal to or greater than 1. The total pressure–total temperature inflow boundary condition is the most appropriate inflow condition to use when the performance of the nozzle (e.g. discharge coefficient, thrust ratio, or thrust coefficient) is to be determined. This set of conditions does not presuppose the mass flow of the nozzle because the performance is a result of geometric characteristics.

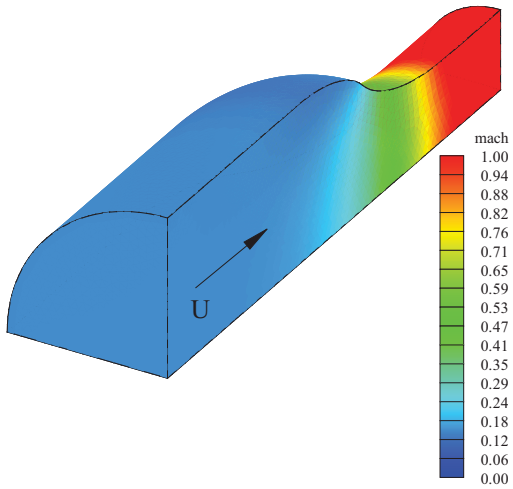
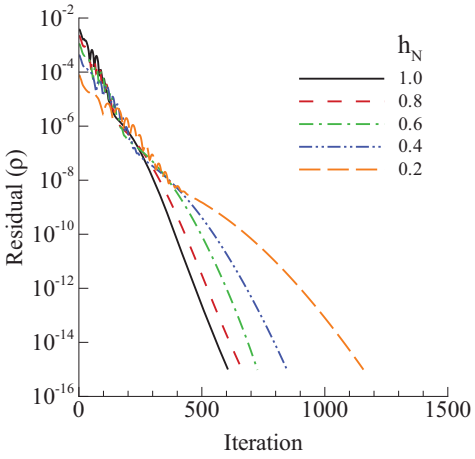


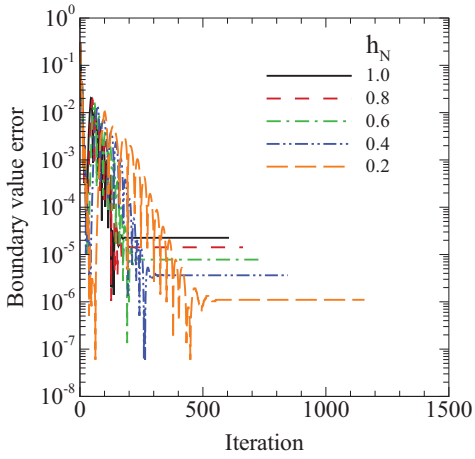
Figure 14. **Mach-number contours for the ASME nozzle test case.**

5.2.1 Pressure Subsonic Inflow (`subsonic_inflow_pt`)

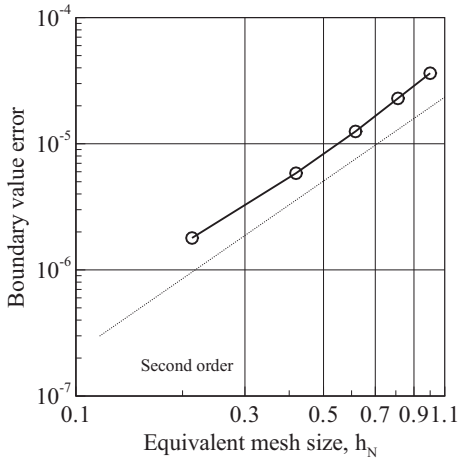
Type	Specify
Inflow	$p_t/p_\infty = 1.6$, $T_t/T_\infty = 1.0$, $\alpha = \beta = 0.0$



(a) **Residual history.**



(b) **Boundary value error history.**



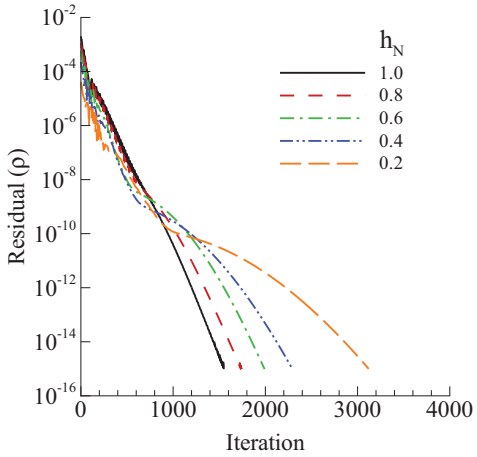
(c) **Boundary value error property.**

Figure 15. **Subsonic inflow (p_t) boundary condition,** `subsonic_inflow_pt`**.**

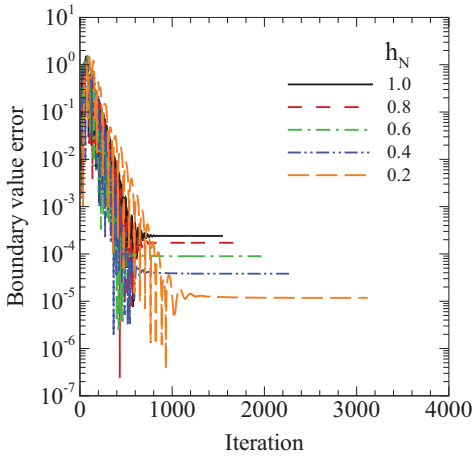
Iterative convergence, shown by the residual history plotted in Figure 15(a), was achieved for all mesh levels for the set condition of $p_t/p_\infty = 1.6$. As shown in Figure 15(b), the error in the total pressure reaches the final value within several hundred iterations for all grids. The error in the set total pressure converges as the square of the equivalent mesh size. The second-order convergence property in the error is indicated by the slope of the dotted line (Figure 15(c)). The average values for the Mach number and the mass flow ratio at the boundary for this total pressure/total temperature setting are approximately 0.14 and 4.4, respectively.

5.2.2 Mass Flow Inflow (`massflux_in`)

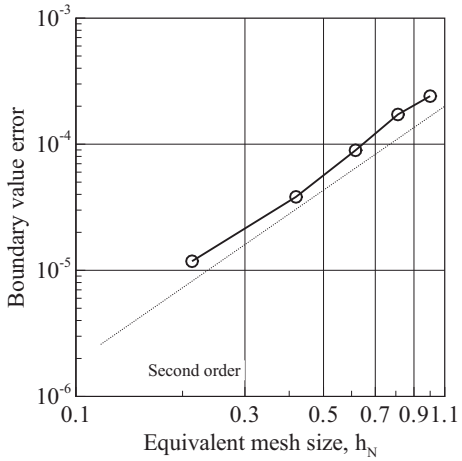
Type	Specify
Inflow	$\dot{m} = 1.0$, $T_t/T_\infty = 1.0$ (unchoked)
Inflow	$\dot{m} = 10.0$, $T_t/T_\infty = 1.0$ (choked)



(a) **Residual history.**



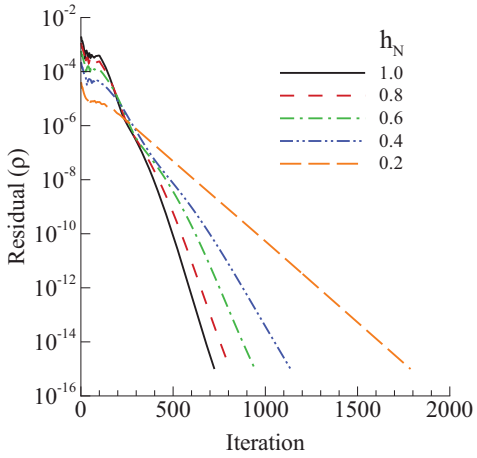
(b) **Boundary value error history.**



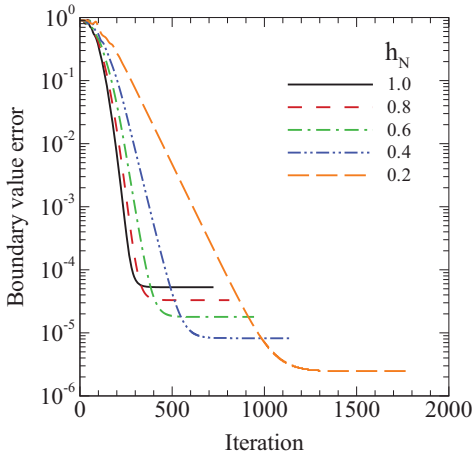
(c) **Boundary value error property.**

Figure 16. **Unchoked mass flow boundary condition, `massflux_in`.**

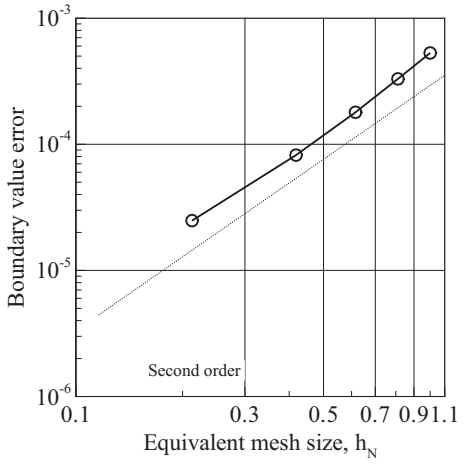
The residual history and the boundary value error history for the mass flow inflow cases are shown in Figure 16 for the unchoked nozzle flow and Figure 17 for the choked nozzle flow condition. Iterative convergence was achieved for all mesh levels for both set conditions of $\dot{m} = 1.0$ and $\dot{m} = 10.0$, as shown in Figures 16(a) and 17(a) respectively. The error in the mass flow reaches the final value in about 1000 iterations for all cases, as shown in Figures 16(b) and 17(b). The boundary value error converges approximately as the square of the equivalent mesh size. The second-order convergence property in the error is



(a) **Residual history.**



(b) **Boundary value error history.**



(c) **Boundary value error property.**

Figure 17. **Choked mass flow boundary condition,** `massflux_in`**.**

indicated by the slope of the dotted line (Figures 16(c) and 17(c)). The average values for the total pressure ratio at the inflow boundary for the unchoked and choked nozzle mass flow conditions are approximately 1.03 and 3.5, respectively.

5.3 Supersonic Diffuser Calculations

A solution using the supersonic fixed inflow boundary condition at the inflow plane of the diffuser is shown in Figure 18. An extrapolation boundary condition was used at the outflow plane. The flow is from left to right, as indicated by the arrow. The flow accelerates from approximately Mach 1.3 at the inflow to Mach 2.3 at the outflow.

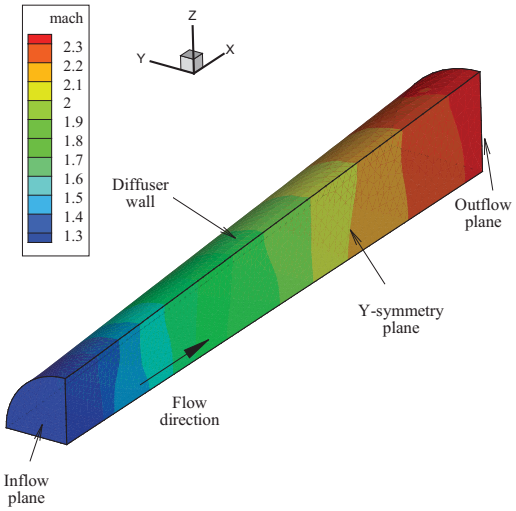


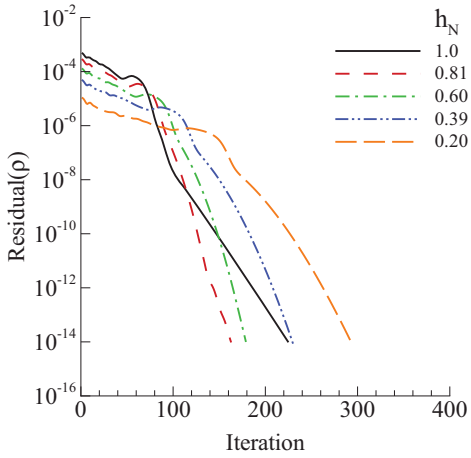
Figure 18. **Mach-number contours for the diffuser test case.**

Type	**Specify**
Inflow	$\rho = 1.2,\ \overrightarrow{\mathrm{U}} = (1.0, 0., 0.),\ \mathrm{p} = 1.0/1.4$

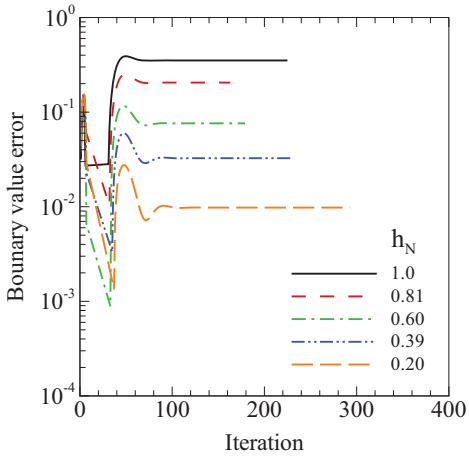
The residual history and the boundary value error history for the fixed supersonic inflow case are shown in Figure 19. Iterative convergence was achieved for all mesh levels for the set conditions within a few hundred iterations. As shown in Figure 19(b), the error in the set velocity achieves the final value for each mesh in less than 100 iterations. The error in the set velocity converges as the square of the equivalent mesh size for the three smallest equivalent mesh sizes. The second-order convergence property in the error is indicated by the slope of the dotted line (Figure 19(c)).

6 Summary

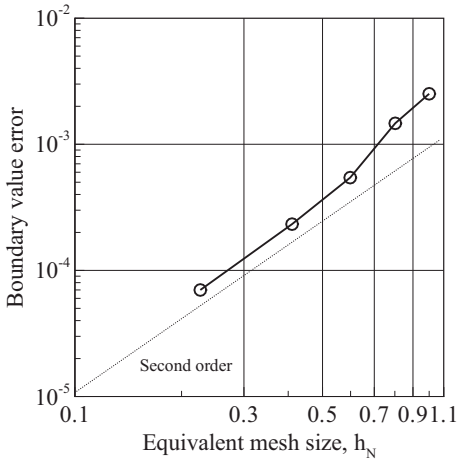
Boundary conditions that allow subsonic and supersonic flow into and out of the computational domain have been derived and implemented in the computational method FUN3D. The boundary conditions are enforced through determination of the flux contribution to the solution residual. Roe's approximate Riemann solver is used to construct the fluxes at the boundary faces. These boundary conditions were implemented in an implicit manner in



(a) **Residual history.**



(b) **Boundary value error history.**



(c) **Boundary value error property.**

Figure 19. **Supersonic inflow boundary condition,** `fixed_inflow`**.**

the FUN3D CFD code and were verified for three generic test geometries. Iterative residual convergence was achieved at all mesh-density levels for all of the conditions that were used in this study. The user-requested condition was achieved in all cases, and error in the boundary condition value decreased in a second-order manner with an increase in mesh density. Boundary value error convergence occurs before iterative convergence and depending on the application, boundary values were within 0.5 percent of the specified parameter value with only a residual reduction of 5 orders of magnitude.

Acknowledegment

The author would like to thank Dr. James Thomas, Dr. Mike Park and Mr. Jeff White of the Computational AeroSciences Branch at NASA Langley Research Center for their many invaluable discussions and contributions to this work.

7 Nomenclature

Roman letters

A	area
c	speed of sound
c_p	specific heat at constant pressure
H	enthalpy
h_N	equivalent mesh size based on degrees of freedom, $N^{-1/3}$
h_v	L_1 norm of $V^{-1/3}$, $\|L_1\| = \|V^{-1/3}\|_1$
$\hat{i}, \hat{j}, \hat{k}$	Cartesian unit vectors axis in physical space
$\dot{m}$	mass flow
M	Mach number
$\vec{n}$	normal vector
$\hat{n}$	unit normal vector, $\vec{n}/\|\vec{n}\|$
N	number of nodes in mesh
p	static pressure
R	real gas constant
R	Riemann invariant
s	entropy
T	temperature
$\overline{U}$	average velocity
$\overrightarrow{U}$	velocity vector, $[u, v, w]^T$
U	velocity magnitude
u,v,w	Cartesian velocity components
V	primal volume of elements in mesh
x, y, z	Cartesian coordinates in physical space

Subscripts

b	boundary state
∞	free-stream state
i	internal state
L	left state

o	outer state
$\perp$	normal to boundary element
R	right state
set	user-requested condition
t	total conditions

Conventions

$\boldsymbol{\mathcal{F}}$	boundary element flux vector
$\boldsymbol{q}$	primitive variables state vector
$\mathcal{R}$	solution residual vector

Symbols

α	u-w angle of velocity
β	u-v angle of velocity
γ	ratio of specific heats, $\gamma = 1.4$ (air)
λ	eigenvalue
ρ	density

References

1. "Numerical Boundary Condition Procedures," *Symposium on Numerical Boundary Condition Procedures*, ed. P. Kutler, NASA CP–2201, October 1981.

2. Moretti, G.: "A Physical Approach to the Numerical Treatment of Boundaries in Gas Dynamics," NASA CP–2201, October 1981, pp. 73–95.

3. Blottner, F.: "Influence of Boundary Approximations and Conditions on Finite Difference Solutions," NASA CP–2201, October 1981, pp. 227–256.

4. Pulliam, T.: "Characteristic Boundary Conditions for the Euler Equations," NASA CP–2201, October 1981, pp. 165–181.

5. Thompkins, W. and Bush, R.: "Boundary Treatments for Implicit Solutions to Euler and Navier-Stokes Equations," NASA CP–2201, October 1981, pp. 353–363.

6. Anderson, W. and Bonhaus, D. L.: "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructed Grids," *Computers & Fluids*, Vol. 23, 1994, pp. 1–21.

7. Kim, S.; Alonso, J.; Schluter, J.; Wu, X. and Pitsch, H.: "Integrated Simulations for Multi-Component Analysis of Gas Turbines: RANS Boundary Conditions," AIAA Paper 2004-3415, July 2004.

8. Griewank, A.: "On Automatic Differentiation," *Mathematical Programming: Recent Developments and Applications*, Kluwer Academic Publishers, 1989, pp. 83–108.

9. Corliss, G. F. and Griewank, A.: "Operator Overloading as an Enabling Technology for Automatic Differentiation," Technical report, Argonne National Laboratory, 1993.

10. White, J. and Morrison, J.: "A Pseudo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations," AIAA Paper 1999-3360, June 1999.

11. Roe, P. L.: "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comp. Phys.*, Vol. 43, 1981, pp. 357–372.

12. Batten, P.; Clarke, N.; Lambert, C. and Causon, D.: "On the Choice of Wavespeeds for the HCCL Riemann Solver," *SIAM J. Sci. Comput.*, Vol. 18, 1997, pp. 1553–1570.

13. Sun, M. and Takayama, K.: "Artificially Upwind Flux Vector Splitting Scheme for the Euler Equations," *J. Comp. Phys.*, Vol. 189, 2003, pp. 305–329.

14. Edwards, J.: "A Low-Diffusion Flux-Splitting Scheme for Navier Stokes Calculations," AIAA Paper 1996-1704, May 1996.

15. van Leer, B.: "Towards the Ultimate Conservative Difference Schemes V. A second Order Sequel to Godunov's Method," *J. Comp. Phys.*, Vol. 32, 1979, pp. 101–136.

16. Roe, P. L.: "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337–365.

17. Barth, T. and Jespersen, D.: "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 1989-0366, Jan. 1989.

18. Venkatakrishnan, V.: "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *J. Comp. Phys.*, Vol. 118, 1995, pp. 120–130.

19. Anderson, W.; Rausch, R. and Bonhaus, D. L.: "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *J. Comp. Phys*, Vol. 128, 1996, pp. 391–408.

20. Gilbert, G. and Hill, P.: "Analysis and Testing of Two-Dimensional Slot Nozzle Ejectors With Variable Area Mixing Sections," NASA CR–2251, May 1973.

21. *Measurement of Fluid Flow in Pipes Using Orifice, Nozzle and Venturi*, The American Society of Mechanical Engineers, 2005.

22. Pirzadeh, S.: "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.

23. Thomas, J.; Diskin, B. and Rumsey, C.: "Toward Verification of Unstructured-Grid Solvers," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3070–3079.

Appendix A

Node Weighting for a Boundary Tetrahedral

The tetrahedral element that is shown in Figure A1 has four nodes with an associated area-weighted normal, $\vec{n}_1$, $\vec{n}_2$, $\vec{n}_3$, and $\vec{n}_4$, opposite each node. The average value of the

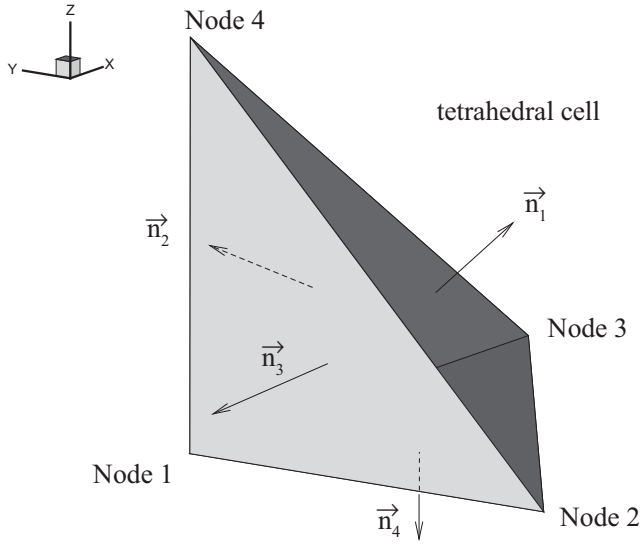


Figure A1. **Tetrahedral element.**

variable $\bar{q}_i$ at the face centers is one-third the sum of the three corners of that face if we assume that a constant gradient exists across the element.

$$\bar{q}_1 = \frac{1}{3}\left(q_{\text{node2}} + q_{\text{node3}} + q_{\text{node4}}\right) \tag{A1}$$

$$\bar{q}_2 = \frac{1}{3}\left(q_{\text{node1}} + q_{\text{node3}} + q_{\text{node4}}\right) \tag{A2}$$

$$\bar{q}_3 = \frac{1}{3}\left(q_{\text{node1}} + q_{\text{node2}} + q_{\text{node4}}\right) \tag{A3}$$

$$\bar{q}_4 = \frac{1}{3}\left(q_{\text{node1}} + q_{\text{node2}} + q_{\text{node3}}\right) \tag{A4}$$

The Green-Gauss (GG) integral over the tetrahedral element is equal to one-third the summation of the product of the average value of $\bar{q}_i$ with each directed face normal, $\vec{n}_i$.

$$\oint_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; \frac{1}{3}\sum_{i=1}^{4} \bar{q}_i \vec{n}_i \tag{A5}$$

The dual of the tetrahedral element shown in Figure A2 has six faces, with the associated normals $\vec{n}_{\text{B}}$, $\vec{n}_{\text{L}}$, and $\vec{n}_{\text{R}}$ on the boundaries and $\vec{n}_{12}$, $\vec{n}_{13}$, and $\vec{n}_{14}$ facing the tetrahedral face opposite node 1. The finite-volume (FV) integral around the dual volume is written as a summation of $\bar{q}$ times the directed normal of the face. At this point, the influence of the nodes on the boundary faces L, R, B, a, b, and c are unknown. The goal is to make the FV

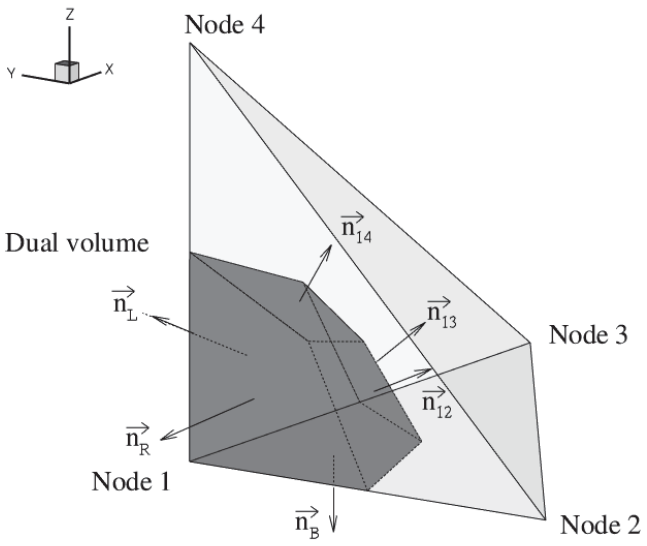


Figure A2. **Tetrahedral dual element.**

integral exact when the q_n are linear.

$$\begin{aligned}\oint_{\text{dual}}^{\text{FV}} \nabla q dV \;=\; & \frac{q_1+q_2}{2}\vec{\text{n}}_{12} + \frac{q_1+q_3}{2}\vec{\text{n}}_{13} + \frac{q_1+q_4}{2}\vec{\text{n}}_{14} \\ & + (aq_1 + bq_3 + cq_4)\,\vec{\text{n}}_L \\ & + (aq_1 + bq_2 + cq_4)\,\vec{\text{n}}_R \\ & + (aq_1 + bq_2 + cq_3)\,\vec{\text{n}}_B \end{aligned} \tag{A6}$$

The FV integral around the dual volume can be equated to one-fourth the GG integral around the tetrahedron.

$$\frac{1}{4}\oint_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; \oint_{\text{dual}}^{\text{FV}} \nabla q dV \tag{A7}$$

The following shows that the left-hand side of Eq. (A7) can be expressed in a form that matches the right-hand side of Eq. (A6). Equation (A5) is substituted into the right-hand side of Eq. (A7) and expanded as

$$\frac{1}{4}\int_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; \frac{1}{4}\frac{1}{3}\sum \bar{q}_i \vec{\text{n}}_i \tag{A8}$$

$$= \frac{1}{12}\left(q_1\vec{\text{n}}_1 + q_2\vec{\text{n}}_2 + q_3\vec{\text{n}}_3 + q_4\vec{\text{n}}_4\right) \tag{A9}$$

Add and subtract $2q_1\vec{\text{n}}_1$ from Eq. (A9), and use the equations that relate $\vec{\text{n}}_2$, $\vec{\text{n}}_3$, and $\vec{\text{n}}_4$ to $\vec{\text{n}}_1$ and the normals of the dual volume, $\vec{\text{n}}_{12}$, $\vec{\text{n}}_{13}$, and $\vec{\text{n}}_{14}$ (Eq. A10).

$$\vec{\text{n}}_i \;=\; 12\vec{\text{n}}_{1i} + \vec{\text{n}}_1 \tag{A10}$$

$$\begin{aligned}\frac{1}{4}\int_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; & \frac{1}{12}\Big[3q_1\vec{\text{n}}_1 - 2q_1\vec{\text{n}}_1 + q_2\left(12\vec{\text{n}}_{12} + \vec{\text{n}}_1\right) \\ & + q_3\left(12\vec{\text{n}}_{13} + \vec{\text{n}}_1\right) + q_4\left(12\vec{\text{n}}_{14} + \vec{\text{n}}_1\right)\Big] \end{aligned} \tag{A11}$$

Split the terms with the $\vec{n}_{1i}$ normal vector as

$$\begin{aligned}\frac{1}{4}\int_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; \frac{1}{12}\Big[&-2q_1\vec{n}_1 + 6q_2\vec{n}_{12} + 6q_3\vec{n}_{13} + 6q_4\vec{n}_{14} \\ &+ 3q_1\vec{n}_1 + q_2\vec{n}_1 + q_3\vec{n}_1 + q_4\vec{n}_1 \\ &+ 6q_2\vec{n}_{12} + 6q_3\vec{n}_{13} + 6q_4\vec{n}_{14}\Big] \end{aligned} \tag{A12}$$

Substituting into Eq. (A12) the additional geometric relations that equate the face normals of $\vec{n}_1$, $\vec{n}_{12}$, $\vec{n}_{13}$, and $\vec{n}_{14}$ with $\vec{n}_L$, $\vec{n}_R$, and $\vec{n}_B$ (Eqs. (A13)-(A16):

$$\vec{n}_1 \;=\; -3\left(\vec{n}_{12} + \vec{n}_{13} + \vec{n}_{14}\right) = 3\left(\vec{n}_L + \vec{n}_R + \vec{n}_B\right) \tag{A13}$$

$$\vec{n}_{12} \;=\; -\frac{1}{4}\left(2\vec{n}_L + \vec{n}_R + \vec{n}_B\right) \tag{A14}$$

$$\vec{n}_{13} \;=\; -\frac{1}{4}\left(\vec{n}_L + 2\vec{n}_R + \vec{n}_B\right) \tag{A15}$$

$$\vec{n}_{14} \;=\; -\frac{1}{4}\left(\vec{n}_L + \vec{n}_R + 2\vec{n}_B\right) \tag{A16}$$

$$\begin{aligned}\frac{1}{4}\int_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; &\frac{q_1+q_2}{2}\vec{n}_{12} + \frac{q_1+q_3}{2}\vec{n}_{13} + \frac{q_1+q_4}{2}\vec{n}_{14} + \frac{1}{12}\left(3q_1\vec{n}_1\right) \\ &+ q_2\left[-\frac{1}{8}\left(2\vec{n}_L + \vec{n}_R + \vec{n}_B\right) + \frac{1}{4}\left(\vec{n}_L + \vec{n}_R + \vec{n}_B\right)\right] \\ &+ q_3\left[-\frac{1}{8}\left(\vec{n}_L + 2\vec{n}_R + \vec{n}_B\right) + \frac{1}{4}\left(\vec{n}_L + \vec{n}_R + \vec{n}_B\right)\right] \\ &+ q_4\left[-\frac{1}{8}\left(\vec{n}_L + \vec{n}_R + 2\vec{n}_B\right) + \frac{1}{4}\left(\vec{n}_L + \vec{n}_R + \vec{n}_B\right)\right] \end{aligned} \tag{A17}$$

Finally, gather terms of similar face vectors:

$$\begin{aligned}\frac{1}{4}\int_{\text{tetrahedron}}^{\text{GG}} \nabla q dV \;=\; &\frac{q_1+q_2}{2}\vec{n}_{12} + \frac{q_1+q_3}{2}\vec{n}_{13} + \frac{q_1+q_4}{2}\vec{n}_{14} \\ &+ \left(\frac{3}{4}q_1 + \frac{1}{8}q_3 + \frac{1}{8}q_4\right)\vec{n}_L \\ &+ \left(\frac{3}{4}q_1 + \frac{1}{8}q_2 + \frac{1}{8}q_4\right)\vec{n}_R \\ &+ \left(\frac{3}{4}q_1 + \frac{1}{8}q_2 + \frac{1}{8}q_3\right)\vec{n}_B \end{aligned} \tag{A18}$$

A direct comparison of the terms in Eq. (A18) with those in Eq. (A6) shows that $a = 3/4$, $b = 1/8$, and $c = 1/8$.

The author would like to thank Nishikawa Hiroaki for this derivation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 10 - 2011	Technical Memorandum	01/2005-12/2008

4. TITLE AND SUBTITLE

Inflow/Outflow Boundary Conditions with Application to FUN3D

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Carlson, Jan-Renee

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20011

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2011-217181

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified Unlimited
Subject Category 64
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Several boundary conditions that allow subsonic and supersonic flow into and out of the computational domain are discussed. These boundary conditions are demonstrated in the FUN3D computational fluid dynamics (CFD) code which solves the three-dimensional Navier-Stokes equations on unstructured computational meshes. The boundary conditions are enforced through determination of the flux contribution at the boundary to the solution residual. The boundary conditions are implemented in an implicit form where the Jacobian contribution of the boundary condition is included and is exact. All of the flows are governed by the calorically perfect gas thermodynamic equations. Three problems are used to assess these boundary conditions. Solution residual convergence to machine zero precision occurred for all cases. The converged solution boundary state is compared with the requested boundary state for several levels of mesh densities. The boundary values converged to the requested boundary condition with approximately second-order accuracy for all of the cases.

15. SUBJECT TERMS

CFD, unstructured mesh, euler, boundary conditions, propulsion simulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	38	(443) 757-5802