*99230*

*P-12*

# Interface of an Uncoupled Boundary Layer Algorithm With an Inviscid Core Flow Algorithm for Unsteady Supersonic Engine Inlets

Douglas Darling and Barbara Sakowski
*Lewis Research Center*
*Cleveland, Ohio*

# NASA

# INTERFACE OF AN UNCOUPLED BOUNDARY LAYER ALGORITHM WITH AN INVISCID CORE FLOW ALGORITHM FOR UNSTEADY SUPERSONIC ENGINE INLETS

Douglas Darling and Barbara Sakowski
NASA - Lewis Research Center, Cleveland, OH

## Nomenclature

| | |
|---|---|
| $A$ | flow area |
| $F_s$ | momentum source term used in core flow equations |
| $M$ | Mach number |
| $M_s$ | mass source term used in core flow equations |
| $P$ | local static pressure |
| $Q_s$ | heat source term used in core flow equations |
| $r_c$ | radius of cowl |
| $r_s$ | radius of centerbody |
| $R_c$ | radius of the cowl lip, used to nondimensionalize the core flow equations |
| $Re$ | local Reynolds Number per unit length |
| $R_0$ | radius of the wall (centerbody or cowl), a function of x but not a function of y |
| $RVB$ | coefficient in front of derivatives of advective terms in y-direction |
| $u$ | local velocity component parallel to the core flow |
| $v$ | local velocity component perpendicular to the core flow |
| $T$ | local static temperature |
| $t$ | time |
| $x$ | distance, direction of core flow |
| $y$ | distance from wall, direction perpendicular to the core flow |
| $y^T$ | distance from wall, transformed by core flow stream area |
| $\rho$ | density |
| $\char94$ | non-dimensional variable in core flow equations |
| $\bar{\ }$ | non-dimensional variable in boundary layer equations |
| $\infty$ | freestream condition, upstream of cowl and centerbody |
| $e$ | core flow, boundary layer edge, condition |
| $max$ | core flow value at point with the highest Mach number, these values are used to non-dimensionalize the boundary layer equations. |
| $0$ | stagnation quantity |
| $ref$ | reference values used to non-dimensionalize core flow equations |

## I. Introduction

Bleeds, bypasses, centerbody translations, and active cooling are some of the complexities of modeling flow within a supersonic jet engine inlet. When developing control laws for inlets a large number of geometries and flow conditions need to be modeled. When a large number of geometries and flow are to be modeled the speed of the modeling technique is very important. It would be beneficial to model the bulk of the flow with a quick inviscid model. In fact, much of the flow inside the inlet of a supersonic jet engine can indeed be treated as inviscid. However, at times, the boundary layer does have a significant effect on the overall flow within an inlet. Thus, we would like a quick and robust method to model the boundary layer effects. It was not the intent of this work to develop a detailed model of shock-wave boundary layer interactions or a detailed flow model, which would be required for a detailed inlet design. The purpose of this work was to develop a fast method to model inlets, including boundary layer effects, when the bulk flow properties and overall flow responses are of the biggest concern, such as in a controls application.

The Large Perturbation Inlet computer code (LAPIN) was developed to model inlets of supersonic jet engines. LAPIN modeled the flow within an inlet as one-dimensional and inviscid. For the most part, this inviscid model did a good job of predicting the flow within a supersonic inlet. However, in regions of strong adverse pressure gradients, near a shock wave, or near the throat of the inlet, the inviscid model did not predict the inlet flow well. These were the regions where boundary layer effects had a significant effect on the overall flow within an inlet.

Roach, et al. [1] showed that an uncoupled, finite difference boundary layer scheme is a fast and accurate algorithm to apply when modeling unsteady, supersonic flows. Using this uncoupled scheme we were able to model attached or separated and turbulent or laminar flows.

In this paper we will look at the problem of interfacing this boundary layer model with an inviscid, core flow model (LAPIN). Then the results of the combined

viscous/inviscid model will be discussed and compared with results of experiments that had been run on the NASA - Lewis 40/60 inlet.

## II. Core Flow and Boundary Layer Models

The core flow and boundary layer regions of the flow field within an inlet were modeled in different manners. This was done due to differences in the governing equations to model the two regions and because they were developed independently. To understand how to interface the two regions (viscous and inviscid) we will first look at each model separately.

The core flow equations were solved by the program LAPIN in conservation form for the flux variables. Then the primitive variables were backed out of the flux terms. The equations were one-dimensional, inviscid, unsteady, and included source terms in each equation.

The core flow governing equations. used in LAPIN. can be written in non-dimensional vector form as--

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}}{\partial x} = \vec{G} \qquad (1)$$

Where the vectors were defined as--

$$\vec{U} = \begin{pmatrix} \hat{\rho} \\ \hat{\rho}\hat{u} \\ \hat{E} \end{pmatrix} \qquad (2)$$

$$\vec{F} = \begin{pmatrix} \hat{\rho}\hat{u} \\ \hat{P} + \hat{\rho}\hat{u}^2 \\ \hat{u}(\hat{E} + \hat{P}) \end{pmatrix} \qquad (3)$$

$$\vec{G} = \begin{pmatrix} \hat{M}_s \\ \frac{\hat{P}}{\hat{A}}\frac{\partial \hat{A}}{\partial x} + \hat{F}_s \\ -\frac{\hat{P}}{\hat{A}}\frac{\partial \hat{A}}{\partial t} + \hat{Q}_s \end{pmatrix} \qquad (4)$$

and the nondimensional variables were defined as--

$$\hat{t} = \frac{u_{ref} t}{R_c} \qquad (5)$$

$$\hat{x} = \frac{x}{R_c} \qquad (6)$$

$$\hat{Q}_s = \frac{Q_s}{\rho_{ref} u_{ref}^3 R_c} \qquad (7)$$

$$\hat{\rho} = \frac{\rho A}{\rho_{ref} R_c^2} \qquad (8)$$

$$\hat{u} = \frac{u}{u_{ref}} \qquad (9)$$

$$\hat{P} = \frac{PA}{\rho_{ref} u_{ref}^2 R_c^2} \qquad (10)$$

$$\hat{E} = \frac{EA}{\rho_{ref} u_{ref}^2 R_c^2} \qquad (11)$$

$$\hat{A} = \frac{A}{R_c^2} \qquad (12)$$

$$\hat{T} = \frac{T}{T_{ref}} \qquad (13)$$

$$\hat{M}_s = \frac{M_s}{\rho_{ref} u_{ref} R_c} \qquad (14)$$

$$\hat{F}_s = \frac{F_s}{\rho_{ref} u_{ref}^2 R_c} \qquad (15)$$

The reference length. $R_c$. was the radius of the cowl lip. All other reference conditions were taken from the upstream sonic conditions. LAPIN solved these equations by marching in time. using any of the following algorithms: Beam-Warming. hybrid Beam Warming. MacCormack's method. or split characteristics MacCormack's method. The split characteristics method tended to yield the best results and allowed the largest time step while maintaining numerical stability. LAPIN was set up to model centerbody translations. bleeds. bypasses. heat sources. and mass injection. A detailed description of the core flow equations and solution algorithms used in LAPIN is given by Varner, et al. [2].

The governing equations used in the boundary layer code were the parabolized Navier-Stokes Equations (PNS). The PNS equations were 2-dimensional. viscous. unsteady. and were solved for the primitive variables. The y-momentum equation simply reduced to

$$\frac{\partial \hat{P}}{\partial y} = 0 \qquad (16)$$

The other governing equations for the boundary layer were as follows:

2

continuity

$$\frac{\partial(\bar{\rho}\ \bar{u})}{\partial x}+\frac{\partial(\bar{\rho}\ \bar{v})}{\partial y}+\frac{\partial\bar{\rho}}{\partial \bar{t}}=0 \qquad (17)$$

x-momentum

$$\bar{\rho}\frac{\partial\bar{u}}{\partial\bar{t}}+\frac{\bar{\rho}\ \bar{u}}{R_o^k}\frac{\partial R_o^k\bar{u}}{\partial x}+\bar{\rho}\ \bar{v}\frac{\partial\bar{u}}{\partial y}=$$

$$\frac{1}{Re}\frac{\partial^2\bar{u}}{\partial y^2}-\frac{\partial\bar{P}}{\partial x} \qquad (18)$$

energy

$$\bar{\rho}\frac{\partial\bar{T}}{\partial\bar{t}}+\frac{\bar{\rho}\ \bar{u}}{R_o^k}\frac{\partial R_o^k\bar{T}}{\partial x}+\bar{\rho}\ \bar{v}\frac{\partial\bar{T}}{\partial y}=$$

$$\frac{1}{RePr}\frac{\partial^2\bar{T}}{\partial y^2}-\frac{1}{Re}\left(\frac{\partial\bar{u}}{\partial y}\right)^2+\frac{\partial\bar{P}}{\partial\bar{t}}+\bar{u}\frac{\partial\bar{P}}{\partial x} \qquad (19)$$

The variables were defined as follows:

$$\bar{\rho}=\frac{\rho}{\rho_{max}} \qquad (20)$$

$$\bar{u}=\frac{u}{u_{max}} \qquad (21)$$

$$\bar{P}=\frac{P}{\rho_{max}u_{max}^2} \qquad (22)$$

$$\bar{T}=\frac{c_pT}{u_{max}^2} \qquad (23)$$

$$\bar{t}=t\ u_{max} \qquad (24)$$

$$Re=\frac{\rho_{max}u_{max}}{\mu_{eff}} \qquad (25)$$

$R_o$ is the radius of the wall for axisymmetric flow ($k=1$). For rectangular coordinates $R_o$ was ignored ($k=0$). It should be noted that the Re and time terms were not actually non-dimensional. Re had units of (length)$^{-1}$, while the time term had units of length. Also, the length variables ($x$, $y$, and $R_o$) remained dimensional. So, each term in each of the above boundary layer equations had units of (length)$^{-1}$.

The boundary layer equations were solved at a given time step knowing the solution at the previous time step. Due to the parabolic nature of the boundary layer equa-

tions, they were solved at each time step by marching from station to station in the direction of the core flow (x-direction). Solutions at a given x-station were obtained by solving the boundary layer equations sequentially. First, the momentum equation was solved for the velocity component in the x-direction (u). Second, continuity was solved for the velocity component in the y direction (v). Then, the energy equation was solved for the temperature. If the solution had not converged, the momentum, continuity, and energy equations were solved again for the velocity components and temperature, using the flow properties from the previous iteration. Convergence at a given x-station was obtained when the x-velocity components (u) at all y-grid locations at that x-station converged. After the values at the station converged, the algorithm marched downstream to the next x-station. This streamwise marching continued throughout the entire solution domain. The differencing of the governing equations followed that of Kwon, et al. [3]. A detailed description of the differencing and solution algorithm used in the boundary layer algorithm is given by Roach, et al. [1].

The boundary layer code used a modified Baldwin-Lomax turbulence model. Modifications were made for regions of adverse pressure gradients and separation. A description of the modified Baldwin-Lomax model is given in Sakowski, et al. [4].

In addition, downstream effects were propagated upstream in the subsonic portions of the boundary layer by using the pressure sweep and a weighted forward/backward differencing of the pressure gradient of Davis and Barnett [5].

### III. Interface of Core Flow and Boundary Layer

As described in the previous section, the core flow and boundary layer models dealt with the solution variables in different manners. The core flow model used conservation forms, while the boundary layer model used non-conservation forms of the governing equations. Also, the two models non-dimensionalize their variables in very different manners. Thus, one of the critical steps in interfacing the two schemes is just translating values from one scheme to the other. The boundary layer model had to be able to interpret the core flow values, and, likewise, the core flow model had be able to interpret the results of the boundary layer model. In addition to this bookkeeping, several other considerations were important. First, the interaction between the inviscid core and the viscous boundary layer had to be modeled quickly and accurately. Second, the coordinate

3

system had to be set up to maintain the parabolic nature of the boundary layer equations in the x-y plane, so that the algorithm could march in the x-direction. Third, the boundary layer properties had to approach the core flow values smoothly as we moved away from the wall of the inlet. Finally, the interface could not compromise the numerical stability of the combined viscous/inviscid model.

Iterating between the core flow and boundary layer codes at a given times step was found to be time consuming and unnecessary for the purpose of this model since such iteration had little effect on the overall flow behavior. A quick method of modeling the core flow/boundary layer interaction was found to be sufficient in this case. The core flow/boundary layer interaction was set up with a time staggering between LAPIN and the boundary layer codes. First, at a given time step, the predicted boundary layer parameters from the previous time step (displacement thickness and friction coefficient) were used to calculate the core flow properties. Then the core flow properties were used to calculate the boundary layer properties at the next time step.

The coordinate system for the boundary layer algorithm had to be set up to maintain the parabolic nature of the boundary layer equations in the x-y plane. One of the boundary layer assumptions was the velocity in the direction perpendicular to the core flow is much smaller than the velocity in the core flow direction. In many boundary layer algorithms, the coordinate system for the boundary layer was set up to follow the wall, with x along the wall and y perpendicular to the wall. There would be no problem with this approach if the duct had a constant area or if the core flow was solved as two dimensional. But, inlets were converging and diverging and LAPIN modeled the core flow as one dimensional. In an actual inlet the inviscid core flow far from the wall would not be parallel to the wall. Thus, as the two dimensional boundary layer code moved away from the wall, the velocity perpendicular to the wall may not necessarily have been small compared with the velocity parallel to the wall. In order to match the one dimensional core flow and maintain the parabolic nature of the equations, a coordinate transformation was set up, based on the core flow stream area. The coordinate system followed the stream lines of the 1-D inviscid core flow. This way the velocity component perpendicular to the core flow stream direction was small everywhere, and the boundary layer assumptions could still be applied.

The transform used is

$$y^T = y * arf(x, t) \qquad (26)$$

the transformed x and t are the same as the original x and t.

$$x^T = x \qquad (27)$$

$$t^T = t \qquad (28)$$

The function arf(x,t) can be taken to be one over the core flow stream area. The boundary layer equations were solved in the transformed $(x^T, y^T, t^T)$ space. The problem with this coordinate system is that $x^T$ was not really perpendicular to $y^T$. This comes from the fact that the core flow solution is 1-D (x-direction only), while the real inlet would have had flow in both the x and y directions. This coordinate system was well suited for the convective terms; however, certain terms should have been expressed in the real (x, y, t) space, such as the viscous terms in the x-momentum equation. For example, the shear stress should have acted act along lines of constant y, not lines of constant $y^T$. For these terms we needed to transform back to the real space to get their values. To do this, the chain rule was used. The partial derivatives in the original boundary layer equations could be replaced with derivatives in the transformed space.

For a generic scalar parameter $\phi$,

$$\frac{\partial \phi}{\partial y} = arf(x^T, t^T) \frac{\partial \phi}{\partial y^T} \qquad (29)$$

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial x^T} +$$

$$\frac{\partial \phi}{\partial y^T} \frac{y^T}{arf(x^T, t^T)} \frac{\partial arf(x^T, t^T)}{\partial x^T} \qquad (30)$$

There was some approximation associated with this scheme, such as mixing terms for the (x, y, y) space and the $(x^T, y^T, t^T)$ space. Again, this is a consequence of matching a 2-D boundary layer model with a 1-D core inviscid model of a flow that was not completely 1-D. This transformation should not be necessary if we were using a 2-D core flow solution. Even with approximation, this transformation worked well as long as there was no y-momentum equation. The v that was calculated in the transformed space fit into the advective terms of the x-momentum and energy equations. However, a $y^T$-momentum equation would be ill-posed since $v^t$ is

4

not always in the same direction nor perpendicular to u or $u^T$.

Another concern when interfacing the boundary layer algorithm with the core flow algorithm was that the boundary layer properties had to smoothly approach the core flow values. That is, when the derivatives perpendicular to the wall were zero, the core flow values had to be a solution to the boundary layer equations. The complicated part of matching the boundary layer and core flows is the two ways the boundary layer and core flow algorithms were differenced. If the differential equations were solved exactly there would not have been a problem, but they were not solved exactly. LAPIN used non-dimensional forms of $\rho$, $\rho u$, and $\rho(e + \frac{1}{2}u^2)$ as the solution variables, while the boundary layer algorithm used non-dimensional forms of the primitive variables $\rho$, $u$, and $T$. Thus, what the core flow algorithm predicted as a solution, may not have been exactly what the boundary layer algorithm predicted as a solution as the y-derivatives go to zero (far from the wall). The difference is usually fairly small (2% or so), but this small difference can have a big effect on the integral performed to calculate the displacement thickness. Adjustment of how the pressure gradient term was calculated in the boundary layer algorithm forced the boundary layer properties to smoothly approach the core values. Without this adjustment, the displacement thickness, predicted by the boundary layer program had large errors. The adjustment of the pressure gradient was performed by solving the finite differenced momentum equations for dP/dx when all y-derivatives were zero. Similarly, the finite-differenced energy equation is solved for dP/dt when all y-derivative were zero. Also, without this adjustment, small errors in the velocity profiles at each station were additive as the algorithm marched downstream and marched in time, resulting in large errors in the friction coefficient near the exit plane of the solution domain and at later time steps.

$$\frac{\partial \overline{P}}{\partial x} = -\overline{\rho}_e \frac{\partial \overline{u}_e}{\partial \overline{t}} - \frac{\overline{\rho}_e \ \overline{u}_e}{R_o^k} \frac{\partial (R_o^k \overline{u}_e)}{\partial x} \qquad (31)$$

$$\frac{\partial \overline{P}}{\partial \overline{t}} = \overline{\rho}_e \frac{\partial \overline{T}_e}{\partial \overline{t}} + \frac{\overline{\rho}_e \ \overline{u}_e}{R_o^k} \frac{\partial (R_o^k \overline{u}_e)}{\partial x} - \overline{u}_e \frac{\partial \overline{P}}{\partial x} \qquad (32)$$

These corrected values of the pressure derivatives were calculated with the edge values from the core flow algorithm using the same differencing scheme used in the boundary layer algorithm. In this way the boundary layer algorithm will approach the core flow values as the y-derivatives approach zero.

The last consideration for interfacing LAPIN and the boundary layer code is a stability consideration. At times the boundary layer algorithm has a stability problem. This problem tended to initiate near the edge of the boundary layer. From one iteration to the next the values near the core flow sometimes fluctuated between less than and greater than the core flow value. Sometimes these fluctuations died out and the program converged. However, other times the oscillations grew, causing the calculations to diverge. To solve this problem flaring was used. RVB was part of the advective terms in the y-direction. RVB was the coefficient in front of the $\partial u/\partial y$ term in the finite differenced x-momentum equation and the $\partial T/\partial y$ term in the finite differenced energy equation. These were the advective terms in the y-direction. Without flaring RVB was simply $\rho v$. With flaring RVB was changed as follows:

$$RVB = K_1 |\ \overline{\rho} \ \overline{v}\ | \qquad (33)$$

$K_1$'s for momentum equation

$$\qquad\qquad if \ u < u_e \qquad\qquad if \ u > u_e$$

$$if \ \frac{\partial u}{\partial y} > 0 \qquad K_1 = -\frac{u}{u_e} \qquad K_1 = \left(1 - \frac{u}{u_e}\right)$$

$$if \ \frac{\partial u}{\partial y} < 0 \qquad K_1 = \left(1 - \frac{u}{u_e}\right) \qquad K_1 = -\frac{u}{u_e}$$

$K_1$'s for energy equation

$$\qquad\qquad if \ T < T_e \qquad\qquad if \ T > T_e$$

$$if \ \frac{\partial T}{\partial y} > 0 \qquad K_1 = -\frac{T}{T_e} \qquad K_1 = \left(1 - \frac{T}{T_e}\right)$$

$$if \ \frac{\partial T}{\partial y} < 0 \qquad K_1 = \left(1 - \frac{T}{T_e}\right) \qquad K_1 = -\frac{T}{T_e}$$

5

The motivation for the above flaring was to make the core flow value a numerically stable solution in the boundary layer algorithm far from the wall. This flaring was found to be very important to help the stability of the algorithm, particularly when there was an adverse pressure gradient, separation, bleeds, or bypasses.

## IV. Results and Discussion

Before describing the results of running the boundary layer/LAPIN codes for an actual inlet, we will look at two simple illustrative examples to show the effects of the corrections described in the last section.

The first illustration is qualitative. Three simple cases were run. All three modeled steady flow over a flat plate. In the first case the correct pressure gradient was used. In the second case the pressure gradient was increased to 0.02/ft over the correct value. Finally, the third case was run by decreasing the pressure gradient by 0.02/ft. Figure 1 shows the predicted velocity profiles 2.5 feet upstream of the leading edge. At this point, the edge Mach number was 1.3. As shown on Figure 1, when the correct pressure gradient was used, the velocity profile smoothly approached the core flow value ($u/u_c=1$). However, if the pressure gradient was too big, the velocity profile approached a value less than the core flow value as the y-derivatives went to zero, far from the wall (then jumped to the core flow value at the last point because of the specified boundary condition). Meanwhile, if the pressure gradient used was too small, the velocity profile approached values larger than the core flow values as the y-derivatives approached zero, far from the wall (then again jumped to the core flow value at the last point because of the forced boundary condition).

This error in the pressure gradient impacted the predicted friction coefficient. The friction coefficient predicted when the pressure gradient was too low was 26% higher than the friction coefficient predicted when the pressure gradient was too high. However, the more significant impact was on the displacement thickness. When the pressure gradient was too high, the displacement thickness was 2.4 times the displacement thickness predicted when the correct pressure gradient was used. When the pressure gradient was too low, an unrealistic negative displacement thickness was predicted. The error in the displacement thickness was a strong function of the height (y-direction) of the solution domain.

As a second illustration, a case was run for unsteady flow over a flat plate. In this illustration the boundary

layer is not interactive with the core flow. Figure 2 shows Mach number distribution at a particular time step. Two cases were run, one using the pressure gradient from the core flow solution and one using the corrected pressure gradient calculated from equation 31. Figure 3 shows the pressure gradients predicted by the core flow calculations and the corrected pressure gradient from equation 32. The correction in the pressure gradient was small and had only a slight impact on the predicted friction coefficient (figure 4). However, as shown on figure 5 the uncorrected pressure gradient allowed the boundary layer code to predict unrealistic negative displacement thicknesses when the pressure gradient was favorable. Then the displacement thickness predicted using the core flow pressure gradient diverged from the displacement thickness predicted using the corrected pressure gradient when the pressure gradient was adverse.

In this second illustration, without the flaring described in the previous section the algorithm was unstable.

Now we will look at a more complicated example. LAPIN was combined interactively with the boundary layer code as described in the previous section. The combined model was used to predict flow in the NASA - Lewis 40-60 Inlet. The geometry of the inlet is given on Table 1 and pictured on figure 6, where $r_c$ is the radius of the cowl and $r_s$ is the radius of the centerbody.

Table 1. NASA - LeRC 40-60 Inlet Geometry

| x/R_c | r_s/R_c | r_c/R_c |
|-------|---------|---------|
| 0.000 | 0.000   |         |
| 2.009 | 0.446   | 1.000   |
| 2.152 | 0.477   | 1.000   |
| 2.294 | 0.509   | 1.000   |
| 2.437 | 0.540   | 1.000   |
| 2.580 | 0.572   | 0.997   |
| 2.722 | 0.604   | 0.990   |
| 2.865 | 0.636   | 0.983   |
| 3.008 | 0.665   | 0.976   |
| 3.150 | 0.688   | 0.970   |
| 3.293 | 0.705   | 0.963   |
| 3.436 | 0.708   | 0.954   |
| 3.578 | 0.698   | 0.943   |
| 3.721 | 0.684   | 0.933   |

| | | |
|---|---|---|
| 3.864 | 0.670 | 0.923 |
| 4.007 | 0.656 | 0.915 |
| 4.150 | 0.640 | 0.908 |
| 4.292 | 0.623 | 0.904 |
| 4.434 | 0.605 | 0.902 |
| 4.577 | 0.586 | 0.900 |
| 4.720 | 0.567 | 0.900 |
| 4.862 | 0.546 | 0.900 |
| 5.005 | 0.524 | 0.902 |
| 5.148 | 0.501 | 0.904 |
| 5.291 | 0.475 | 0.906 |
| 5.433 | 0.449 | 0.908 |
| 5.576 | 0.426 | 0.912 |
| 5.719 | 0.403 | 0.916 |
| 5.861 | 0.378 | 0.918 |
| 6.004 | 0.351 | 0.918 |
| 6.147 | 0.330 | 0.918 |
| 6.289 | 0.309 | 0.915 |
| 6.432 | 0.289 | 0.908 |
| 6.575 | 0.271 | 0.901 |
| 6.717 | 0.256 | 0.893 |
| 6.860 | 0.245 | 0.887 |
| 7.003 | 0.240 | 0.882 |
| 7.145 | 0.239 | 0.878 |
| 7.288 | 0.239 | 0.873 |
| 7.431 | 0.239 | 0.869 |
| 7.573 | 0.239 | 0.865 |
| 7.716 | 0.239 | 0.862 |

$$R_c = 0.78 \text{ ft}$$

With bleeds at the following locations:

| | | |
|---|---|---|
| $3.223 < x/R_c < 3.254$ | | on the cowl |
| $3.683 < x/R_c < 3.697$ | | on the cowl |
| $3.783 < x/R_c < 3.797$ | | on the cowl |
| $3.363 < x/R_c < 3.429$ | | on the centerbody |
| $3.359 < x/R_c < 3.373$ | | on the centerbody |
| $3.693 < x/R_c < 3.707$ | | on the centerbody |

And, a bypass at the following location:

$$6.235 < x/R_c < 6.845 \quad \text{on the cowl}$$

The inlet was modeled with an upstream Mach number $(M_\infty)$ of 2.5 and with the exit Mach number changing from 0.361 to 0.397 over 0.005 seconds. A time step of 0.0002 seconds was used. Figure 7 shows the predicted displacement thickness after the tenth time step (t = 0.002 sec). The displacement thickness predicted without using the pressure correction is erratic and unrealistic. Because the boundary layer and core codes were interactive, the pressure correction now effects the combined algorithm as well. As shown on figure 8, the displacement thickness predicted without the pressure correction moved farther from the corrected model as the algorithm marched in time. In fact the algorithm actually became unstable on the 16th time step, so it did not predict any more solutions.

As in the second illustrative example above, the algorithm was unstable without the flaring of the advective terms described in the previous section.

Finally, the combined core flow/boundary layer model was compared with experimental results from Cubbison, et al. [6], run on the NASA - Lewis 40-60 inlet. The Mach number upstream of the inlet $(M_\infty)$ was again 2.5. Figure 9 shows the ratio of local static pressure to the free stream static pressure versus position in the inlet when a normal shock was near the throat of the inlet. Without the boundary layer code LAPIN predicted the shock wave to be downstream of experimental position. The combination of LAPIN and the boundary layer code predicted the shock wave farther upstream, closer to the experimental value. Also, the prediction of the exit pressure is improved when the boundary layer code was included. (Note: an exit Mach number boundary condition was used). Figure 10 shows similar results when the shock wave was well downstream of the throat. Again, with the boundary layer code, the prediction of the shock position is closer to the experimental value than without the boundary layer code. Also, the prediction of the pressures near the throat was also improved by including the boundary layer code.

## V. Conclusion

The uncoupled boundary-layer scheme was shown to be an effective method for correcting for viscous effects in a model of a supersonic inlet. This was effective when the overall flow variables are of concern such as in a controls type application. However, a more detailed model of shock-wave boundary layer interactions and other specific flow details should be used for detailed designs of inlets.

Corrections to the model were made to maintain the parabolic nature of the boundary layer equations and to ensure the boundary layer properties smoothly approached the core flow values. Also flaring was used to improve the stability of the algorithm. With these corrections the boundary layer algorithm corrected for boundary layer effects near the throat of an inlet and near a shock wave.

## References

1.      R.L. Roach. C. Nelson. B.A. Sakowski. D.D. Darling. A.G. van de Wall. "A Fast. Uncoupled. Compressible, Two-Dimensional, Unsteady, Boundary Layer Algorithm with Sep-

aration for Engine Inlets," AIAA No. 92-3082, 1992.

2.    M.O. Varner. W.R. Martindale. W.J. Phares. J.C. Adams, Jr., "Large Perturbation Flow Field Analysis and Simulation for Supersonic Inlets: Final Report," NASA CR 174676, 1984.

3.    O.K. Kwon. R.H. Pletcher. R.A. Delaney. "Solution Procedure for Unsteady Two-Dimensional Boundary Layer." ASME Journal of Fluids Engineering, Vol. 110, March 1988. pp. 69-75.

4.    B.A. Sakowski, D.D. Darling. R.L. Roach, A.G. van de Wall, "Evaluation and Application of the Baldwin-Lomax Turbulence Model in Two-Dimensional. Unsteady. Compressible Boundary Layers with and without Separation in Engine Inlets," AIAA 92-3676. 1992.

5.    R.T. Davis, M. Barnett. "The calculation of Supersonic Viscous Flows using the Parabolized Navier Stokes Equations." Computers and Fluids. Vol. 14, No.3, pp 197-224. 1986.

6.    R.W. Cubbison. E.T. Meleason. D.F. Johnson. "Performance Characteristics from Mach 2.58 to 1.98 of an Axisymmetric Mixed-Compression Inlet System with 60-Percent Internal Contraction," NASA TM X-1739, 1969.
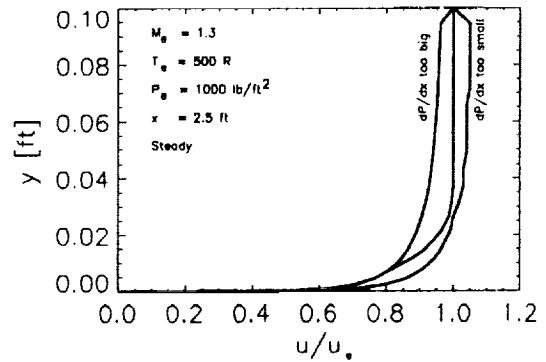
**Figure 1**    Predicted velocity profiles using prescribed pressure gradients.
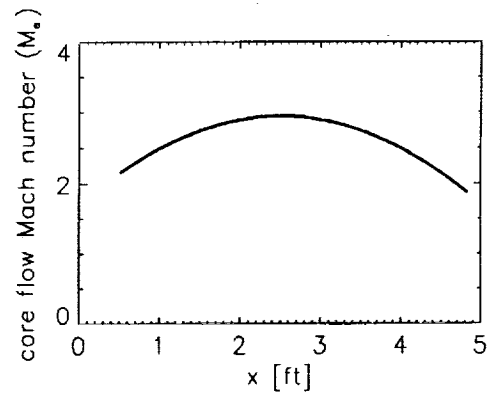


**Figure 2**    Core flow Mach number versus x for flow over a flat plate with a variable pressure gradient.
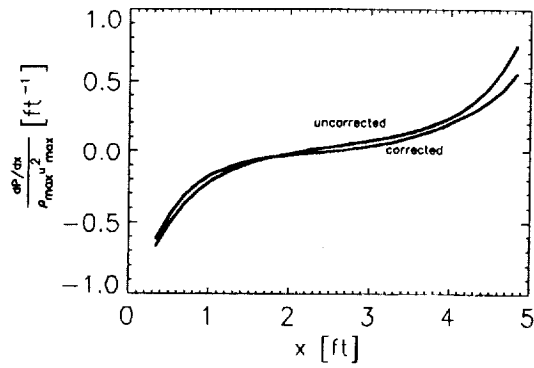
**Figure 3** Corrected pressure gradient (equation 31) and uncorrected pressure gradient (using core flow values) for flow over a flat plate.
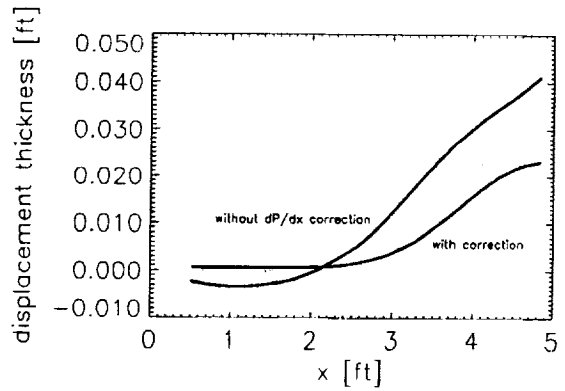
**Figure 5** Predicted displacement thickness with and without the pressure gradient correction of equation 31.
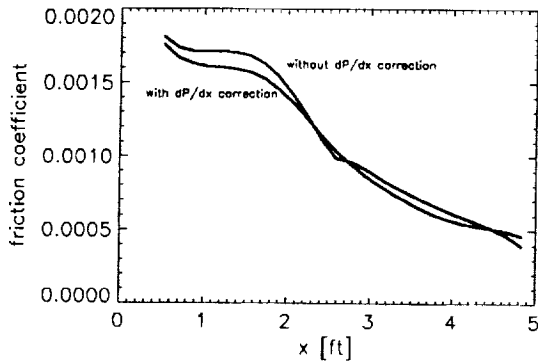


**Figure 4** Predicted friction coefficient with and without the pressure gradient correction of equation 31.
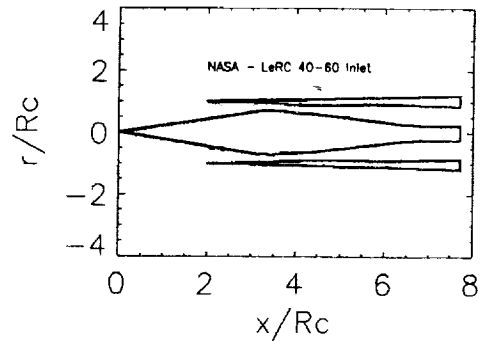
**Figure 6** Geometry of the NASA - LeRC 40-60 supersonic inlet.
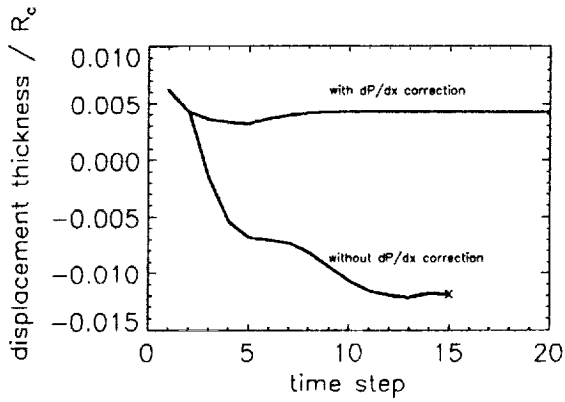
9

**Figure 7**     Predicted displacement thickness on the cowl inside the 40-60 inlet with an without the pressure gradient correction of equation 31, at time = 0.002 seconds.
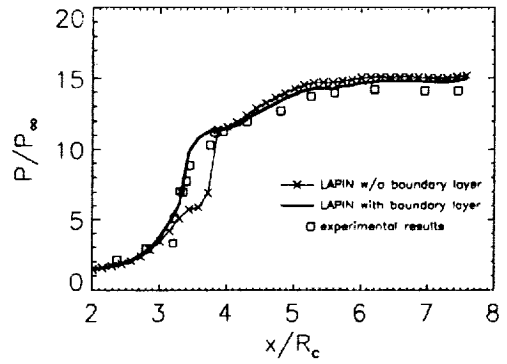
**Figure 9**     Predictions of pressure distribution within the 40-60 supersonic inlet, with and without the boundary layer algorithm, compared with experimental results of Cubbison, et al. [6].
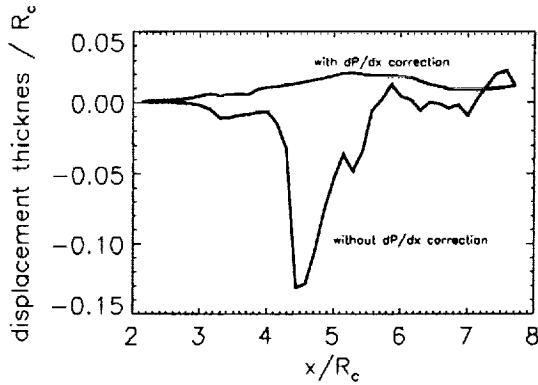


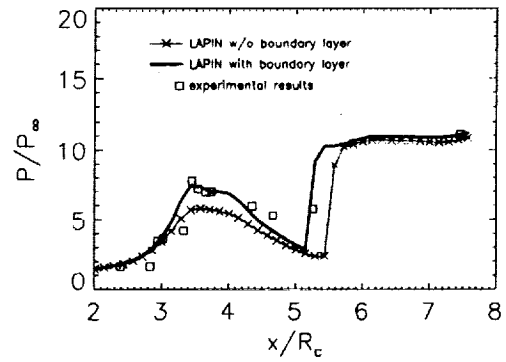**Figure 8**     Predicted displacement thickness on the cowl at location $x/R_c=3.29$.

**Figure 10**     Predictions of pressure distribution within the 40-60 supersonic inlet, with and without the boundary layer algorithm, compared with experimental results of Cubbison, et al. [6].

10

# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | July 1992 | Technical Memorandum |

**4. TITLE AND SUBTITLE**

Interface of an Uncoupled Boundary Layer Algorithm With an Inviscid Core Flow Algorithm for Unsteady Supersonic Engine Inlets

**6. AUTHOR(S)**

Douglas Darling and Barbara Sakowski

**5. FUNDING NUMBERS**

WU–505–62–20

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

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

**8. PERFORMING ORGANIZATION REPORT NUMBER**

E–7063

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

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

NASA TM–105684
AIAA–92–3083

**11. SUPPLEMENTARY NOTES**

Prepared for the 28th Joint Propulsion Conference and Exhibit cosponsored by the AIAA, SAE, ASME, and ASEE, Nashville, Tennessee, July 6–8, 1992. Responsible person, Douglas Darling, (216) 433–2370.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Unclassified - Unlimited
Subject Categories 07, 34, and 61

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (Maximum 200 words)**

An uncoupled boundary layer algorithm was combined with an inviscid core-flow algorithm to model flows within supersonic engine inlets. The inviscid flow algorithm that was used was the Large Perturbation Inlet Code (LAPIN). The boundary layer and inviscid core-flow algorithms were formulated in different manners. The boundary layer algorithm was two-dimensional and solved in non-conservation form, while the core-flow algorithm was one-dimensional and solved in conservation form. In order to interface the two codes the following modifications were important. The coordinate system was set up to maintain the parabolic nature of the boundary layer algorithm while approaching the one-dimensional core-flow solution far from a wall. The pressure gradient used in the boundary layer equation was calculated using the core-flow values and the boundary layer equations so the boundary layer solution smoothly approached the core-flow values far from the wall. Flaring was used for the advective terms perpendicular to the core-flow to maintain the stability of the algorithm. With these modifications, the combined viscous/inviscid algorithm matched well with experimental observations of pressure distributions within a supersonic inlet.

**14. SUBJECT TERMS**

Boundary layer flow; Compressible flow; Numerical analysis; Intake systems

**15. NUMBER OF PAGES**
12

**16. PRICE CODE**
A03

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |