

Combined LAURA-UPS Solution Procedure
for Chemically-Reacting Flows

by

William A. Wood

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science

University of Washington

1994

Approved by _____

(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree _____

Date _____

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature_____

Date_____

University of Washington

Abstract

Combined LAURA-UPS Solution Procedure for Chemically-Reacting Flows

by William A. Wood

Chairperson of Supervisory Committee: Professor Scott Eberhardt
Department of Aeronautics
and Astronautics

A new procedure seeks to combine the thin-layer Navier-Stokes solver LAURA with the parabolized Navier-Stokes solver UPS for the aerothermodynamic solution of chemically-reacting air flowfields. The interface protocol is presented and the method is applied to two slender, blunted shapes. Both axisymmetric and three-dimensional solutions are included with surface pressure and heat transfer comparisons between the present method and previously published results. The case of Mach 25 flow over an axisymmetric six degree sphere-cone with a non-catalytic wall is considered to 100 nose radii. A stability bound on the marching step size was observed with this case and is attributed to chemistry effects resulting from the non-catalytic wall boundary condition. A second case with Mach 28 flow over a sphere-cone-cylinder-flare configuration is computed at both two and five degree angles of attack with a fully-catalytic wall. Surface pressures are seen to be within five percent with the present method compared to the baseline LAURA solution and heat transfers are within 10 percent. The effect of grid resolution is investigated and the nonequilibrium results are compared with a perfect gas solution, showing that while the surface pressure is relatively unchanged by the inclusion of reacting chemistry the nonequilibrium heating is 25 percent higher. The procedure demonstrates significant, order of magnitude reductions in solution time and required memory for the three-dimensional case over an all thin-layer Navier-Stokes solution.

TABLE OF CONTENTS

List of Figures	iii
List of Tables	v
Nomenclature	vi
Chapter 1: Introduction	1
Chapter 2: Prelude to the Present Method	3
Chapter 3: Present Method	5
3.1 The Codes and Their Governing Equations	5
3.1.1 LAURA	5
3.1.2 UPS	7
3.2 Modifications for Compatibility	9
3.3 Remaining Differences Between the Codes	10
3.4 Interface Protocol	12
Chapter 4: Results	16
4.1 Case 1	17
4.1.1 Attempts at Larger Marching Steps	25
4.2 Case 2	29
4.2.1 Two Degrees Angle of Attack	29
4.2.2 Timing	33
4.2.3 Grid Convergence	35
4.2.4 Comparison with Perfect Gas	38
4.2.5 Five Degrees Angle of Attack	41
4.3 Shuttle: A Case for Future Work	44

Chapter 5: Concluding Remarks	47
Bibliography	49
Appendix A: UPS Starting Plane Extraction Program	53
Appendix B: UPS External Grid Extraction Program	64
Appendix C: External Grid Adaptor Routine for UPS Solutions	66

LIST OF FIGURES

3.1	Comparison of molecular oxygen species viscosities.	11
3.2	Comparison of nitric oxide species viscosities.	11
3.3	Comparison of molecular nitrogen species viscosities.	12
4.1	Case 1 LAURA computational grid, showing every fourth body-normal point.	18
4.2	Mach contours: UPS solution overlaid upon LAURA solution.	19
4.3	Case 1 surface pressures—the interface region.	20
4.4	Case 1 surface pressures to 100 nose radii.	20
4.5	Case 1 surface heating—the interface region.	21
4.6	Case 1 surface heating to 100 nose radii.	22
4.7	Case 1 atomic oxygen mass fraction profiles at $X/R_n = 5$	23
4.8	Damköhler stability restriction.	25
4.9	Case 1 LAURA convergence history.	26
4.10	Case 1 heat transfers for a fully-catalytic wall.	28
4.11	LAURA Case 2 symmetry plane grid, showing every eighth body-normal point.	30
4.12	Case 2 surface pressures, windside and leeside.	31
4.13	Case 2 pressure contours in the crossflow plane at $X/R_n = 29$	31
4.14	Case 2 windside heat-transfer rates.	32
4.15	Case 2 leeside heat-transfer rates.	33
4.16	Case 2 LAURA convergence history.	34
4.17	Case 2 body-normal grid resolution: LAURA windside heating.	36
4.18	Case 2 body-normal grid resolution: LAURA leeside heating.	36
4.19	Case 2 body-normal grid resolution: UPS windside heating.	37
4.20	Case 2 body-normal grid resolution: UPS leeside heating.	37
4.21	UPS Case 2 marching step size convergence check—heating.	38

4.22	UPS Case 2 marching step size convergence check—surface pressure. .	39
4.23	Perfect gas heating at Case 2 conditions.	40
4.24	Perfect gas surface pressures at Case 2 conditions.	40
4.25	Windside heat transfers at five degrees angle of attack.	41
4.26	Leeside heat transfers at five degrees angle of attack.	42
4.27	Surface pressures for Case 2, five degrees angle of attack.	43
4.28	LAURA convergence history for Case 2, five degrees angle of attack. .	43
4.29	Surface geometry of shuttle orbiter.	44
4.30	Shuttle windside and leeside surface pressures.	45

LIST OF TABLES

4.1	Nominal conditions for Cases 1 and 2.	17
4.2	Reaction rates for ionized nitric oxide.	23

NOMENCLATURE

c	mass fraction
C_p	frozen specific heat at constant pressure
D	binary diffusion coefficient
D_a	Damköhler number
e	internal energy
E_t	total energy
$\vec{\mathbf{F}}$	flux vector
h	enthalpy
h_0	reference enthalpy at the reference temperature
H	total enthalpy
$\vec{i}, \vec{j}, \vec{k}$	unit vectors
k	conductivity
k_b	backward reaction rate constant
k_f	forward reaction rate constant
k_{eq}	equilibrium constant
L_e	Lewis number
M	Mach number
M	molecular weight
\vec{n}	normal vector
\mathcal{N}	number of moles
P	pressure
\mathbf{Q}	conserved variables
q	heat transfer rate
\bar{R}	universal gas constant, 8314.3 J/kg-mole-K
R_n	nose radius
Re	Reynolds number
s	distance measured along body surface

t	time
T	temperature
u, v, w	cartesian velocity components
\vec{V}	velocity vector
U	diffusion velocity
\mathbf{W}	source terms
X, Y, Z	cartesian coordinates
y_s	mole fraction of species s
Y_b	body location
Y_{shock}	shock location
α	angle of attack
γ	ratio of frozen specific heats
θ_b	body angle
μ	viscosity
ξ, η, ζ	curvilinear coordinates
ρ	density
σ	area element
τ	characteristic time
$\tau_{(x, y, z)}$	stress tensor elements
$\dot{\omega}$	production rate source term
ω_v	Vigneron factor
Ω	volume element
∇	gradient operator

Subscripts:

i, j, k	vector components
l	LAURA
s	species
u	UPS
$wall$	value at the wall
∞	freestream value

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to Dr. S. Eberhardt, Associate Professor of Aeronautics and Astronautics, for his assistance and guidance in the preparation of this manuscript. Thanks are also extended to Dr. P. A. Gnoffo for his help in achieving the LAURA solutions and Drs. R. N. Gupta and K.-P. Lee for providing the Gupta-VSL solution data. Assistance in formatting the text was provided by W. L. Kleb.

Chapter 1

INTRODUCTION

International interest in a space station, the possibilities for human exploration to other planets, and the advancing age of the space shuttle fleet have all brought the issue of advanced launch and reentry vehicles to the forefront. A critical design point for these vehicles is during hypersonic reentry, when peak heating rates occur and aerodynamic control effectiveness may be altered due to flowfield phenomenon unique to the high-altitude, high-velocity conditions. The high temperatures and high convective velocities relative to reaction times create an environment where chemical nonequilibrium effects can be significant. Accurate aerothermodynamic predictions during this part of the reentry trajectory are essential for sizing both the thermal protection system and aerodynamic control surfaces. Ground based tests simulating these flight conditions, including considerable nonequilibrium effects, are difficult to perform. Flight tests can be prohibitively expensive.

Two popular computational approaches for obtaining aerothermodynamic predictions on these classes of vehicles are to solve the thin-layer Navier-Stokes (TLNS) equations or the parabolized Navier-Stokes (PNS) equations. TLNS is derived from the full Navier-Stokes equations by neglecting viscous terms in the streamwise and crossflow directions. The assumptions inherent in the TLNS equations are often acceptable for a wide class of conditions and configurations, including cases of hypersonic, chemically reacting flow. Excessive computational requirements can become a drawback to using TLNS as the entire solution domain is relaxed simultaneously in time. Complex configurations[17] can tax computer memory with millions of grid points, and solution times may be measured in CPU days. In addition, solving for chemical nonequilibrium can, for some algorithms performing exact matrix inversions, increase the computer memory and time requirements by the cube of the number of species considered[7].

The PNS equations are obtained from the full Navier-Stokes equations by neglect-

ing the time derivatives and the streamwise viscous derivatives. Limited to flowfields with streamwise supersonic flow outside the boundary layer, no streamwise separation, and weak streamwise pressure gradients in the subsonic region, PNS algorithms are well suited for solving sharp-nosed, slender-body supersonic/hypersonic configurations. Being space marching and steady state, PNS formulations can realize appreciable decreases in both computational time and memory requirements relative to TLNS algorithms. The principle difficulty in applying the PNS equations to the class of vehicles considered here is that commonly the algorithms cannot solve blunt-body flowfields, and most reentry vehicle designs incorporate blunted nose and leading edge regions in order to reduce peak heating rates.

The present study looks to combine two well-established computational codes, one a TLNS algorithm and the other PNS, for the solution of chemically-reacting, hypersonic flowfields. The technique is successfully applied to axisymmetric geometries at both zero and non-zero angles of attack. Different sets of freestream conditions are considered, and the effects of wall catalycity are investigated. Challenges and obstacles to the consistent integration of the two codes are observed and comments regarding the applicability and limitations of the procedure are documented.

Chapter 2

PRELUDE TO THE PRESENT METHOD

Recently, efforts have been made to combine the TLNS and PNS approaches in order to get timely, accurate hypersonic viscous solutions while circumventing some of the above mentioned limitations. Weilmuenster and Gnoffo[31] proposed a multi-block solution procedure, in which the domain is divided into blocks ordered in the streamwise direction. The general idea is to march these blocks downstream, analogous to the PNS approach of marching two-dimensional planes, and to solve the interior of each block with TLNS. This procedure principally attacks the memory requirements inherent in obtaining a full-body TLNS solution by splitting the domain, but does not decrease the time required to obtain the solution since TLNS remains the governing equations.

The TLNS code used by Weilmuenster is Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)[6, 9, 10, 11]. LAURA is a finite-volume, shock-capturing, hyperbolic equation solver with second-order spatial accuracy for the steady-state solution of viscous or inviscid hypersonic flows. The scheme employs a point implicit relaxation strategy with the upwind flux-difference splitting of Roe[24]. The right-hand-side (RHS) of the equations are formulated according to Yee[34] with the entropy condition of Harten[15]. Perfect gas, equilibrium air, and nonequilibrium air calculations can all be performed.

Greene[12] has extended the LAURA code into a PNS version. With this method, LAURA-TLNS is used on the blunt-nose portion of a hypersonic vehicle. At a point in the flowfield consistent with the PNS equations, the transfer is made to LAURA-PNS, which is then marched down the remainder of a slender vehicle afterbody. This particular formulation, being a TLNS extension, is locally iterative in pseudo-time steps, and its performance suffers from arriving as a PNS solver via a TLNS algorithm, rather than being a code that was optimized as a PNS solver from inception. Thus, while this method significantly reduced the memory required to obtain a solution, it was not able to reduce solution time to the level desired.

Upwind Parabolized Navier-Stokes Solver (UPS)[5, 18, 19, 20, 21, 26, 27, 28, 29] is an upwind, finite-volume, state-of-the-art PNS code with chemical nonequilibrium capability. It is second-order accurate in the crossflow plane and first order accurate in the marching direction. The equations are approximately factored and solved implicitly, with the approach of Vigneron *et al.*[30] employed to suppress departure solutions.

UPS was identified as a code that, when combined with LAURA, might provide the tremendous reduction in vehicle solution time originally sought with the LAURA-TLNS/LAURA-PNS method. The present method seeks to combine LAURA and UPS for a consistent solution procedure for air flows in chemical nonequilibrium.

Previously, UPS has been joined with the TLNS code CNS by Lawrence *et al.*[20] for perfect gas computations. Nonequilibrium solutions are presented by Buelow *et al.*[4] and Muramoto[22] using UPS with the TLNS code TUFF. LAURA has the advantage over TUFF in that it can handle generic, three-dimensional geometric shapes, as are encountered with real vehicle configurations, and is an upwind, finite volume method, like UPS.

Chapter 3

PRESENT METHOD

A combined LAURA-UPS solution procedure has been implemented by Wood and Thompson[33] for perfect gas and equilibrium air flows. That study included detailed solutions for an axisymmetric perfect gas case and a three-dimensional equilibrium air solution for the Reentry F vehicle[25], including turbulence. Generally good results were seen with the combined method, and a very significant reduction in solution time was achieved. The extension of this procedure to nonequilibrium air calculations, however, is not straightforward, because while both UPS and LAURA use the same equilibrium air curve fits, they do not use the same chemistry models for nonequilibrium air.

3.1 The Codes and Their Governing Equations

3.1.1 LAURA

LAURA versions 3.1 and 4.0.2 were used in the present study. TLNS solutions with chemical nonequilibrium and thermal equilibrium were sought for a seven-species air model. The governing equations can be written in integral form, for simplicity presented here in a three-dimensional coordinate system with the body surface in the x-y plane. The LAURA code itself incorporates these equations with a full curvilinear coordinate transformation. The conservation equations are,

$$\iiint \frac{\partial \mathbf{Q}}{\partial t} d\Omega + \iint \vec{\mathbf{F}} \cdot \vec{n} d\sigma = \iiint \mathbf{W} d\Omega \quad (3.1)$$

where the conserved variables are,

$$\mathbf{Q} = \begin{bmatrix} \rho_s \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{bmatrix} \quad (3.2)$$

and the flux vector is,

$$\begin{aligned}
\vec{\mathbf{F}} &= \begin{bmatrix} \rho_s u - \rho D_s \frac{\partial y_s}{\partial x} \\ \rho u^2 + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ \rho u H - k \frac{\partial T}{\partial x} - u \tau_{xx} - v \tau_{xy} - w \tau_{xz} - \rho \sum_s h_s D_s \frac{\partial y_s}{\partial x} \end{bmatrix} \vec{i} \\
&+ \begin{bmatrix} \rho_s v - \rho D_s \frac{\partial y_s}{\partial y} \\ \rho v^2 + P - \tau_{yy} \\ \rho vu - \tau_{yx} \\ \rho vw - \tau_{yz} \\ \rho v H - k \frac{\partial T}{\partial y} - u \tau_{yx} - v \tau_{yy} - w \tau_{yz} - \rho \sum_s h_s D_s \frac{\partial y_s}{\partial y} \end{bmatrix} \vec{j} \\
&+ \begin{bmatrix} \rho_s w - \rho D_s \frac{\partial y_s}{\partial z} \\ \rho w^2 + P - \tau_{zz} \\ \rho wu - \tau_{zx} \\ \rho wv - \tau_{zy} \\ \rho w H - k \frac{\partial T}{\partial z} - u \tau_{zx} - v \tau_{zy} - w \tau_{zz} - \rho \sum_s h_s D_s \frac{\partial y_s}{\partial z} \end{bmatrix} \vec{k} \quad (3.3)
\end{aligned}$$

The source term is,

$$\mathbf{W} = \begin{bmatrix} \dot{\omega}_s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.4)$$

where $\dot{\omega}_s$ accounts for the species production due to finite-rate chemistry.

The thin-layer assumption as applied in LAURA retains viscous derivatives only in the body-normal direction. The shear stresses then become,

$$\tau_{xx} = \tau_{yy} = -\frac{2}{3} \mu \frac{\partial w}{\partial z} \quad (3.5)$$

$$\tau_{zz} = \frac{4}{3} \mu \frac{\partial w}{\partial z} \quad (3.6)$$

$$\tau_{xy} = \tau_{yx} = 0 \quad (3.7)$$

$$\tau_{xz} = \tau_{zx} = \mu \frac{\partial u}{\partial z} \quad (3.8)$$

$$\tau_{yz} = \tau_{zy} = \mu \frac{\partial v}{\partial z} \quad (3.9)$$

The system is closed with the equation of state by assuming the fluid to be a mixture of calorically imperfect gases, such that,

$$P = \sum_s \rho_s \bar{R}T / M_s \quad (3.10)$$

For more details on the governing equations, see [1, 8, 9, 11].

3.1.2 UPS

The PNS solutions were obtained with UPS using the seven species, seven reaction nonequilibrium air chemistry model with the reaction rates of Blottner *et al.*[3]. Like LAURA, UPS uses a general curvilinear coordinate transformation when solving the governing equations, but for clarity the equations are presented here in cartesian form, following the approach of section 3.1.1.

Starting from Eqn. 3.1, the first term, the time derivatives, are dropped as part of the PNS assumptions. UPS solves the fluid dynamics and reacting chemistry in a decoupled manner, so $\dot{\omega}$ in Eqn. 3.4 is set to zero while solving the fluid mechanics, which drops out the RHS of Eqn. 3.1. This leaves only the second term, which in strong conservation form can be written,

$$\frac{\partial \mathbf{F}_i}{\partial x} + \frac{\partial \mathbf{F}_j}{\partial y} + \frac{\partial \mathbf{F}_k}{\partial z} = 0 \quad (3.11)$$

where the three vector components of $\vec{\mathbf{F}}$ have been expressed as,

$$\mathbf{F}_i = \vec{\mathbf{F}} \cdot \vec{i}, \quad \mathbf{F}_j = \vec{\mathbf{F}} \cdot \vec{j}, \quad \mathbf{F}_k = \vec{\mathbf{F}} \cdot \vec{k}$$

The PNS assumption on the streamwise viscous derivatives as applied in UPS drops the streamwise derivatives in the shear stress terms, as well as the entire viscous flux in the streamwise direction, which allows space marching in the streamwise direction outside the subsonic boundary layer region. The Vigneron factor[30], ω_v , is applied to the streamwise pressure gradient to allow marching in the subsonic region.

Incorporating these assumptions, the fluxes can be written as,

$$\mathbf{F}_i = \begin{bmatrix} \rho u \\ \rho u^2 + \omega_v P \\ \rho uv \\ \rho uw \\ u(E_t + P) \end{bmatrix} \quad (3.12)$$

$$\mathbf{F}_j = \begin{bmatrix} \rho v \\ \rho v u - \tau_{yx} \\ \rho v^2 + P - \tau_{yy} \\ \rho v w - \tau_{yz} \\ v(E_t + P) - u\tau_{yx} - v\tau_{yy} - w\tau_{yz} - k\frac{\partial T}{\partial y} - \rho \sum_s c_s U_s h_s \end{bmatrix} \quad (3.13)$$

$$\mathbf{F}_k = \begin{bmatrix} \rho w \\ \rho w u - \tau_{zx} \\ \rho w v - \tau_{zy} \\ \rho w^2 + P - \tau_{zz} \\ w(E_t + P) - u\tau_{zx} - v\tau_{zy} - w\tau_{zz} - k\frac{\partial T}{\partial z} - \rho \sum_s c_s U_s h_s \end{bmatrix} \quad (3.14)$$

where,

$$\tau_{xx} = -\frac{2}{3}\mu \left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (3.15)$$

$$\tau_{yy} = \frac{4}{3}\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \frac{\partial w}{\partial z} \quad (3.16)$$

$$\tau_{zz} = \frac{4}{3}\mu \frac{\partial w}{\partial z} - \frac{2}{3}\mu \frac{\partial v}{\partial y} \quad (3.17)$$

$$\tau_{xy} = \tau_{yx} = \mu \frac{\partial u}{\partial y} \quad (3.18)$$

$$\tau_{xz} = \tau_{zx} = \mu \frac{\partial u}{\partial z} \quad (3.19)$$

$$\tau_{yz} = \tau_{zy} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \quad (3.20)$$

and

$$E_t = \rho \left[e + \frac{1}{2}(u^2 + v^2 + w^2) \right] \quad (3.21)$$

The Vigneron factor is,

$$\omega_v = \min \left[1, \frac{\gamma M^2}{1 + (\gamma - 1)M^2} \right] \quad (3.22)$$

In practice, Eqn. 3.22 is applied with a safety factor.

The chemistry is loosely coupled to the fluid dynamics through the species continuity equation,

$$\rho \left[\frac{\partial c_s}{\partial t} + \vec{V} \cdot \nabla c_s \right] = \nabla \cdot (\rho D_s \nabla c_s) + \dot{\omega}_s \quad (3.23)$$

The equation of state completes the equation set in the same manner as Eqn. 3.10. For further details, see [21, 28, 29].

3.2 Modifications for Compatibility

Changes made to the LAURA pre-processor for compatibility with UPS focus mostly on grid generation. The grid on the cone portion of a sphere-cone was changed from being body normal to being axis normal so as to facilitate space marching on slender bodies. The spacing normal to the body in the initial grid was modified so as to better capture the bow shock for vehicles with very slender afterbodies. The number of cells solved on spherical nose caps was reduced to 12.

The wall boundary conditions in LAURA were changed to correspond with the UPS wall boundary conditions by switching from the standard LAURA boundary conditions to the primary alternate boundary conditions. The standard LAURA viscous wall boundary conditions apply the wall values, *i.e.*, zero velocity, fixed wall temperature, *etc.*, at the center of an image cell below the vehicle surface. The UPS approach is to use reflected boundary conditions for the image cell, so as to apply the boundary conditions to be at the cell face defining the wall. The UPS approach is considered to be a higher-order method than the default LAURA boundary imposition. However, the LAURA default boundary conditions were found to be more robust than the reflected boundary conditions, so the LAURA solutions were first partially converged with the standard boundary conditions, and then switched to the reflective boundary conditions during the later stages of convergence after the flow-field had stabilized. This switch is usually made at the same time spatial second-order accuracy is enforced.

Of the five kinetic models available in LAURA, the 15 reaction model of Kang *et al.*[16] was chosen as being the closest match with the rates of Blottner[3] in UPS. Two further parameters were toggled from the default in LAURA to better deal with slender-body configurations. The eigenvalue limiter was set to be scaled by the cell aspect ratio and the upwinding of the surface properties was turned off.

The principle change made to UPS involved the restart file. A jump in properties was observed during nonequilibrium restarts. This was tracked to the use of a freestream value of the mixture molecular weight when initially decoding the temperature from the energy and species concentrations, prior to marching. The remedy was to read the local mixture molecular weight into the standard restart file.

3.3 Remaining Differences Between the Codes

Some differences in the chemistry models remain between LAURA and UPS. Algorithmically, LAURA solves the chemistry equations with a fully-coupled procedure while UPS uses a loosely-coupled approach, but with the option for local subiterations to get a close approximation to a fully-coupled scheme. The two codes compute the species enthalpies with fundamentally different approaches, as LAURA uses polynomial curve fits while UPS uses interpolated table look-ups. This prevents the exact matching of species concentrations, internal energy, and temperature between the codes, though the magnitude of the difference is considered to be small enough to not prohibit the interfacing of the codes.

Further differences exist in the way each code computes the bulk thermodynamic and transport properties. This leads to small mismatches between the codes for parameters such as viscosity and speed of sound. One question this raises is whether to match the non-dimensional freestream quantities Mach number and Reynolds number between the codes, or to match the dimensional freestream velocity and density. For high Reynolds number, hypersonic applications where Mach number independence has been reached, the decision made here is to match the dimensional freestream conditions. As examples of the differences in the transport property computations, plots of viscosity versus temperature are presented for molecular oxygen, Fig. 3.1, and nitric oxide, Fig. 3.2. The computations of both LAURA and UPS are presented along with the recommendations of Gupta *et al.*[14], who conducted one of the most recent studies into transport property computations. Generally, the other species

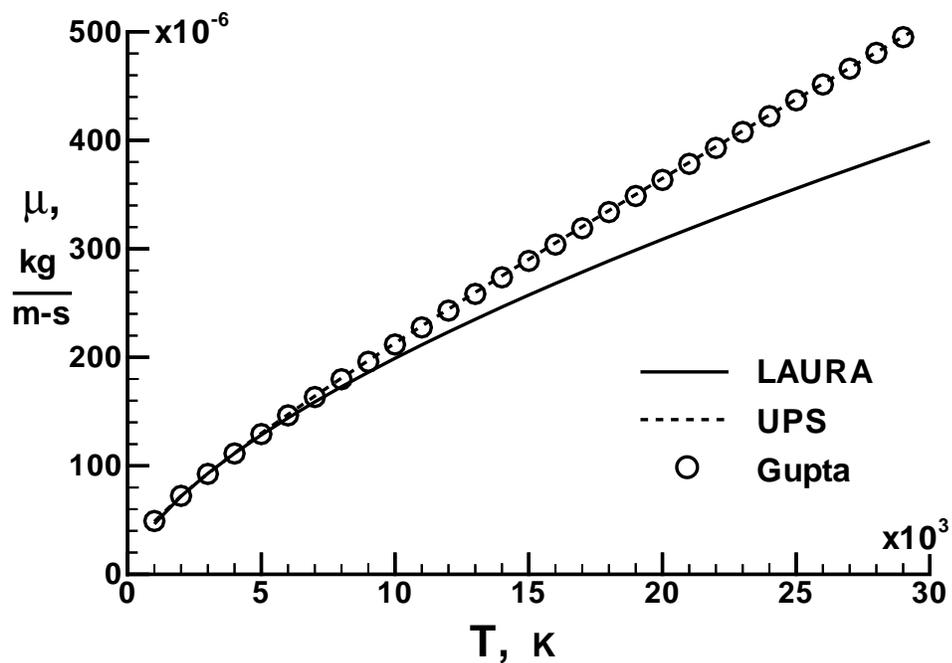


Fig. 3.1 Comparison of molecular oxygen species viscosities.

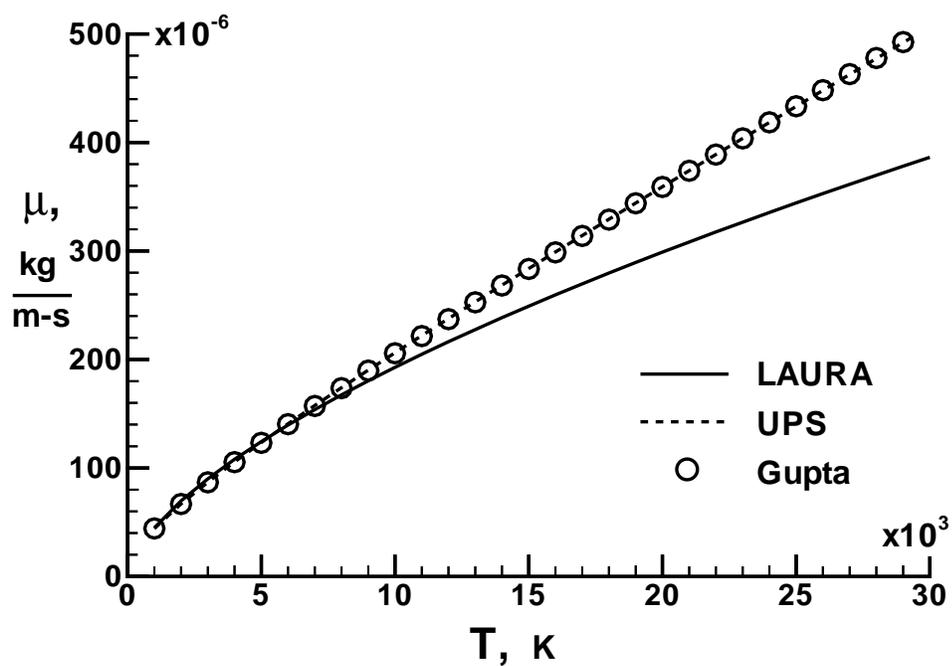


Fig. 3.2 Comparison of nitric oxide species viscosities.

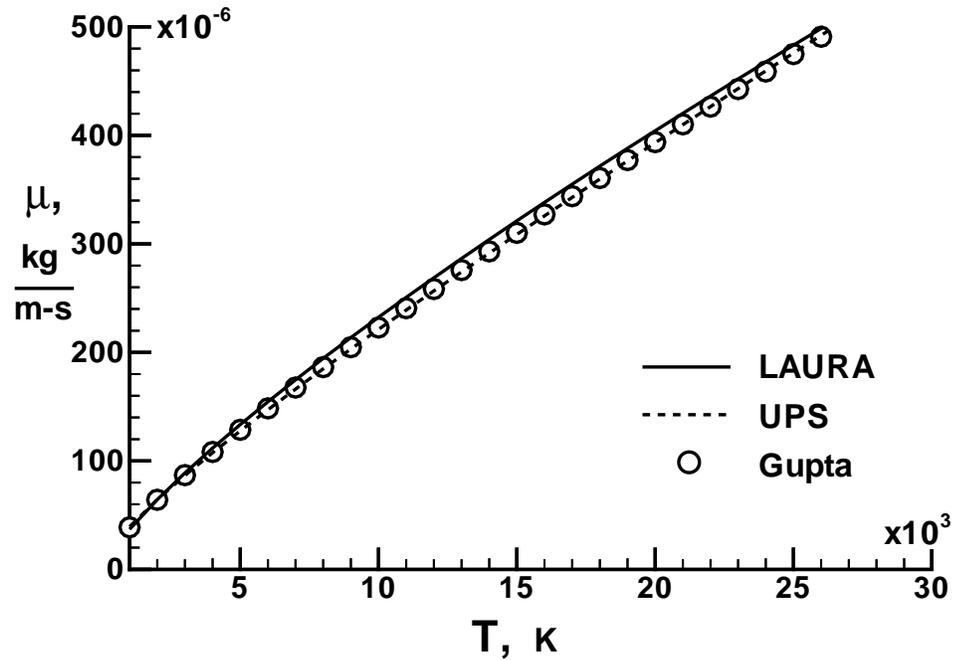


Fig. 3.3 Comparison of molecular nitrogen species viscosities.

viscosities match fairly well over the temperature range 1000–30,000 K, with the molecular nitrogen viscosity computations presented in Fig. 3.3 as a typical example. Sample computations of thermal conductivities performed for typical near wall conditions resulted in a 4–5 percent higher value from UPS than LAURA. It is difficult to predict *a priori* what effect these differences would have relative to solutions from the two codes.

3.4 Interface Protocol

The interface procedure between LAURA and UPS begins with the standard LAURA restart file for a converged chemical nonequilibrium solution. From the LAURA restart file, a crossflow data plane is extracted to become the UPS starting plane. Currently, this plane is chosen at least three cells upstream of the final LAURA solution plane in order to avoid possible contamination from the outflow boundary conditions. The variables available in the LAURA restart file are: the three velocity

components, temperature, the seven species densities, and the finite volume grid,

$$[u, v, w, T, \rho_s, x, y, z]_l$$

The variables needed by UPS to start are: mixture density, the three momentum components, total energy, mixture molecular weight, species mass fractions, and the starting plane in finite volume form,

$$[\rho, \rho u, \rho v, \rho w, E_t, M, c_s, x, y, z]_u$$

In the equations which follow, a subscripted “l” is used to indicate a LAURA variable or quantity, while the subscript “u” refers to the corresponding UPS parameter.

The variables required by UPS are obtained from the LAURA variables in the following manner. The grid is transformed according to the transformation of the physical coordinates as,

$$x_u = -z_l, \quad y_u = x_l, \quad z_u = y_l \quad (3.24)$$

The total density is found from summing the species densities,

$$\rho_u = \sum_s \rho_{s,l} \quad (3.25)$$

The three components of momentum are obtained from the velocity components and the total density,

$$\rho u_u = \rho_u \cdot (-w_l), \quad \rho v_u = \rho_u \cdot u_l, \quad \rho w_u = \rho_u \cdot v_l \quad (3.26)$$

Species mass fractions are found by dividing the species densities by the total density,

$$c_{s,u} = \frac{\rho_{s,l}}{\rho_u} \quad (3.27)$$

The mixture molecular weight is found by applying the perfect gas equation of state to the mixture temperature and pressure,

$$M_u = \frac{\rho_u \bar{R} T_l}{P_u} \quad (3.28)$$

where the mixture pressure was determined from summing the species partial pressures,

$$P_u = \rho_u \sum_s \frac{c_{s,u} \bar{R} T_l}{M_s} \quad (3.29)$$

a step consistent with the assumption, common to both codes, that the working fluid is a mixture of ideal gases.

The UPS total energy now remains to be computed. Initially, the effort was made to take the temperature and species densities from the LAURA solution, pass them through the LAURA enthalpy curve fits, add in the velocity and species property information, and obtain a total energy that would be passed directly to UPS. A problem was encountered when UPS took this energy and decoded temperature and pressure. The differences between the UPS and LAURA enthalpy computations lead to differences between the decoded UPS temperatures and pressures and the original LAURA temperatures and pressures. These variations, in combination with a fixed wall-temperature boundary condition and the Vigneron condition's limitation on the pressure gradient near the wall set up oscillations that restricted the stability of the marching UPS solution. The fix to this problem was to pass the LAURA temperatures directly through to the UPS species enthalpy interpolated table look-ups, then to complete the computation of the total energy as described above,

$$E_{t,u} = \frac{1}{2}(u_l^2 + v_l^2 + w_l^2) + H_u - \frac{P_u}{\rho_u} \quad (3.30)$$

where,

$$H_u = \sum_s h_{s,u} c_{s,u} \quad (3.31)$$

and,

$$h_{s,u} = C_{p,u} T_l + h_{0,s,u} \quad (3.32)$$

Since both LAURA and UPS are finite volume formulations, the UPS starting-plane grid is taken at a streamwise location corresponding to the location of the cell-centered LAURA data. Converting the nondimensionalizations so that the UPS velocities are normalized by the freestream speed of sound, rather than the freestream velocity as is done in LAURA, and performing the curvilinear transformation between the two codes,

$$\xi_u = -\eta_l, \quad \eta_u = \zeta_l, \quad \zeta_u = \xi_l \quad (3.33)$$

completes the interface process. The actual programs used to extract the UPS starting plane and external grid from the LAURA solution can be found in Appendices A and B, respectively.

Chapter 4

RESULTS

The present method is successfully applied to two primary configurations and flow conditions. Case 1 is an axisymmetric sphere-cone, chosen to correspond with the results of Gupta *et al.*[13] The nose radius is 0.0381 m and the body angle is six degrees. The freestream conditions are for Mach 25 at an altitude of 53.34 km (175 kft.). The wall temperature is held fixed at 1260 K, with a non-catalytic chemistry boundary condition. Case 2 is for Mach 28 flow over the sphere-cone-cylinder-flare configuration studied by Bhutta *et al.*[2] at both two and five degree angles of attack. This configuration has a 0.1524 m spherical nosecone followed by a nine degree cone. After 10 nose radii the cone is followed by a cylinder and then a five degree flare, each of 10 nose radii length. The freestream conditions correspond to an altitude of 83.8 km, (275 kft.), at a Reynolds number per meter of 6148. The wall temperature for this case is 833 K and a fully-catalytic boundary condition is employed. Table 4.1 presents a summary of the nominal conditions for the two cases. For all calculations the freestream species concentrations were set at,

$$\begin{bmatrix} c_{N_2} \\ c_{O_2} \\ c_N \\ c_O \\ c_{NO} \\ c_{NO^+} \end{bmatrix} = \begin{bmatrix} 0.767 \\ 0.233 \\ 6.217 \times 10^{-20} \\ 7.758 \times 10^{-9} \\ 4.981 \times 10^{-5} \\ 4.567 \times 10^{-24} \end{bmatrix} \quad (4.1)$$

The seventh specie, electrons, are found from a charge balance with the ionized nitric oxide,

$$\mathcal{N}_{e^-} = \mathcal{N}_{NO^+} \quad (4.2)$$

Table 4.1 Nominal conditions for Cases 1 and 2.

	Case 1	Case 2
Configuration	sphere-cone	blunted multi-conic
M_∞	25	28
Re (m^{-1})	3.95×10^5	6148
Altitude (km)	53.34	83.8
R_n (m)	0.0381	0.1524
θ_b (deg)	6	9-0-5 (10 R_n each)
T_{wall} (K)	1260	833
Wall catalycity	none	fully
α (deg)	0	2, 5

4.1 Case 1

A viscous, second-order accurate TLNS LAURA solution was obtained for Case 1 with chemical nonequilibrium, thermal equilibrium, and a non-catalytic wall condition, implemented in both codes as,

$$\left. \frac{\partial c_s}{\partial n} \right|_{wall} = 0 \quad (4.3)$$

i.e., the mass fractions of the image cells are set equal to the mass fractions of the first cell outside the wall. The axisymmetric LAURA computational grid contains 64 cells normal to the body and 28 cells in the streamwise direction, extending five nose radii to 0.19 m. This grid was adapted using the standard LAURA grid adaption routine. Figure 4.1 displays the final LAURA grid, for clarity showing only every fourth point in the body-normal direction. For consistency, Fig. 4.1 and all subsequent figures use the UPS coordinate system. The location where the UPS starting plane was extracted from the LAURA solution is indicated in Fig. 4.1. That portion of the LAURA grid downstream of the UPS starting plane was supplied as an external grid to UPS. Since the UPS marching step size was smaller than the LAURA cell sizes shown in Fig. 4.1, the LAURA grid was linearly interpolated in the streamwise direction to obtain the actual UPS grid. This is the standard UPS approach for handling external grids. A significant overlap of the solution domains was deliberately chosen for this case to

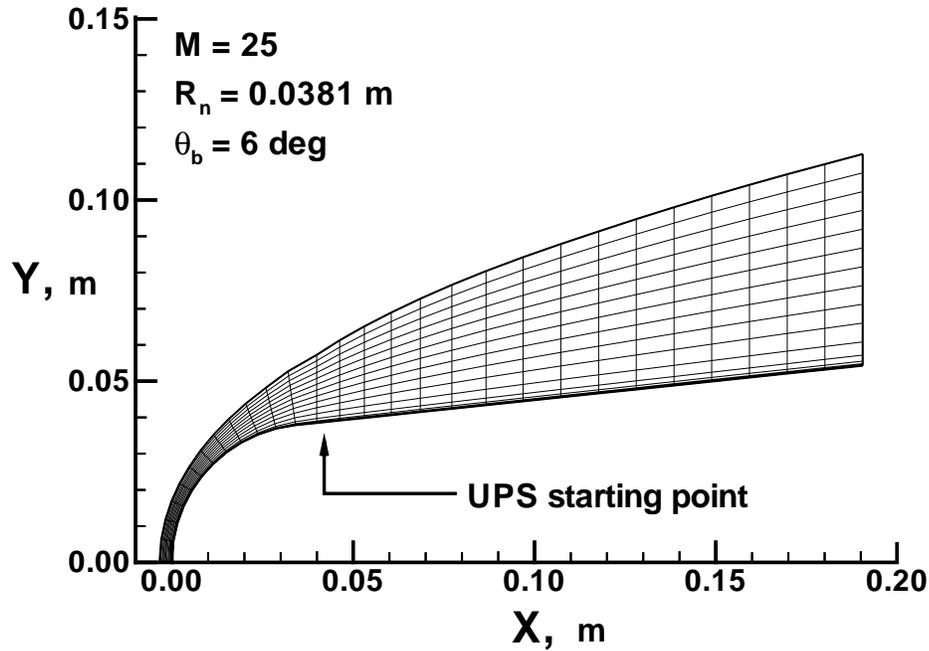


Fig. 4.1 Case 1 LAURA computational grid, showing every fourth body-normal point.

allow for a direct code-to-code comparison between LAURA and UPS. In general, an overlap of this size is not required by the combined procedure.

The UPS solution was carried out 100 nose radii to 3.8 m by extending the external grid downstream in a conical extrapolation. The grid was moderately adapted to the solution in the body-normal direction as the solution proceeded, in such a way as to maintain the original grid spacing at the wall while linearly stretching the outer 60 percent of the grid. This adaption routine is currently not fully integrated into the version of UPS used here, and relies upon the user to provide the necessary stretching parameters. A copy of the grid adaption routine can be found in Appendix C.

Figure 4.2 displays Mach contours from the LAURA and UPS solutions, covering the overlap region to five nose radii. The location of the UPS starting plane is indicated, and the UPS Mach contours are overlaid upon the LAURA Mach contours downstream of that point. Excellent agreement is seen between the present method and the LAURA-only solution.

Figures 4.3 and 4.4 plot surface pressures, normalized by twice the freestream

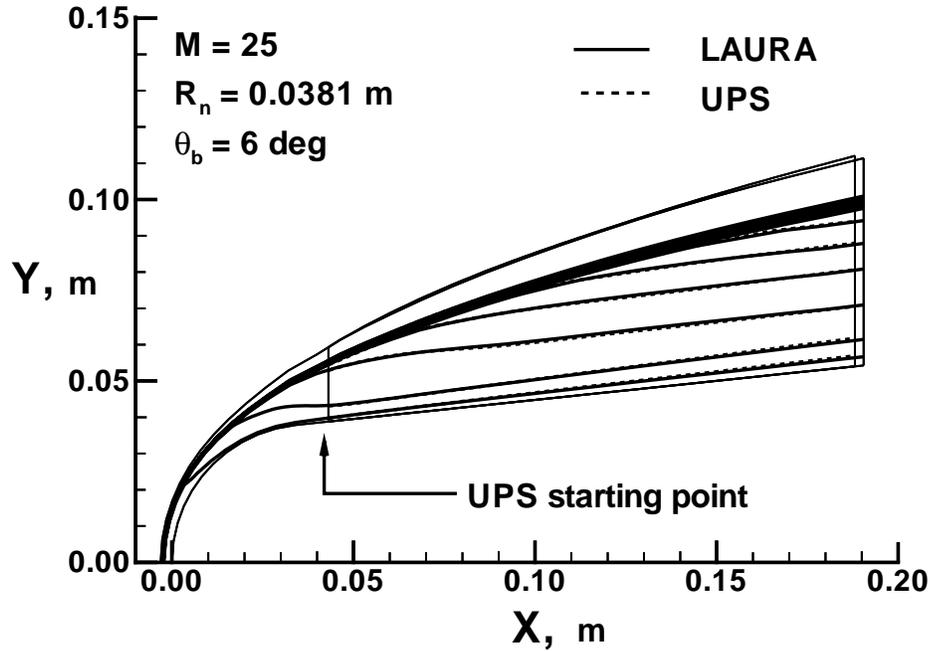


Fig. 4.2 Mach contours: UPS solution overlaid upon LAURA solution.

dynamic pressure, versus the streamwise distance measured along the surface, normalized by the nose radius. The viscous shock-layer (VSL) solutions of Gupta[13] are included for comparison. The VSL equations employ a further approximation to the governing equations beyond the PNS equations to allow solution marching in both the streamwise and circumferential directions. Figure 4.3 is a close-up on the interface region, extending to 10 nose radii. The Gupta-VSL solution extends the full 10 nose radii, while the LAURA solution was terminated at six nose radii in this plot. The UPS solution was initiated at two nose radii and extends to 10 nose radii. Excellent agreement is seen between the UPS and LAURA solutions. The Gupta-VSL solution is seen to agree very well with the LAURA and UPS solutions outside of the region of sphere-cone tangency, where Gupta-VSL predicts higher pressures. An elevated pressure bump appears in the UPS solution at five nose radii. The cause for this is not known for certain at this time, but it could be a residual of the LAURA-UPS interface.

Figure 4.4 extends the surface pressure plot out to 100 nose radii, capturing the overexpansion and recompression regions. There is a maximum difference between the UPS and Gupta-VSL solutions of 3–4 percent in the recompression region. As in

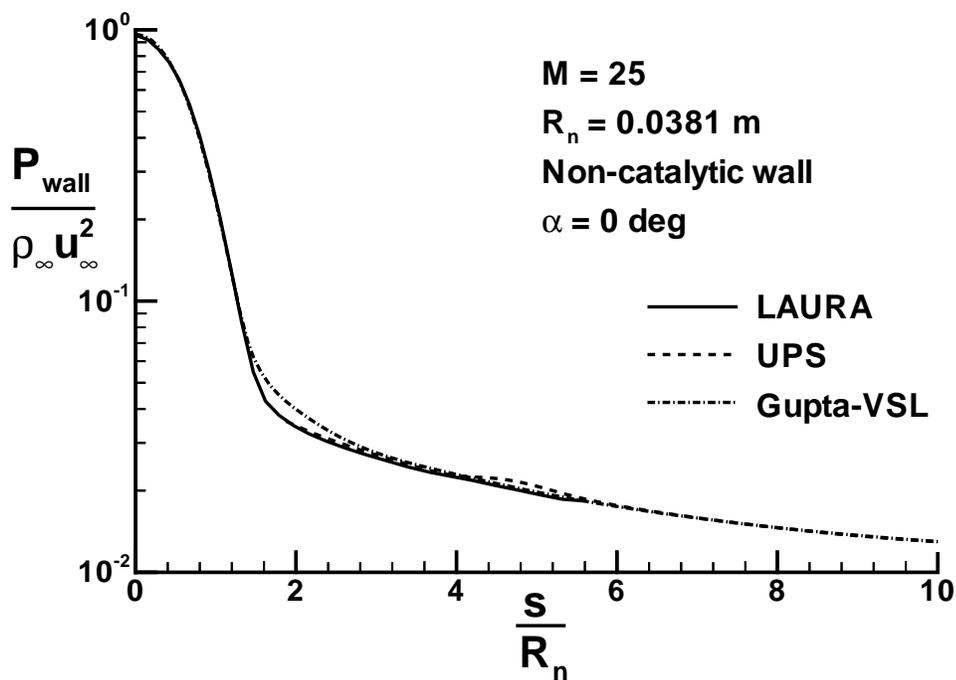


Fig. 4.3 Case 1 surface pressures—the interface region.

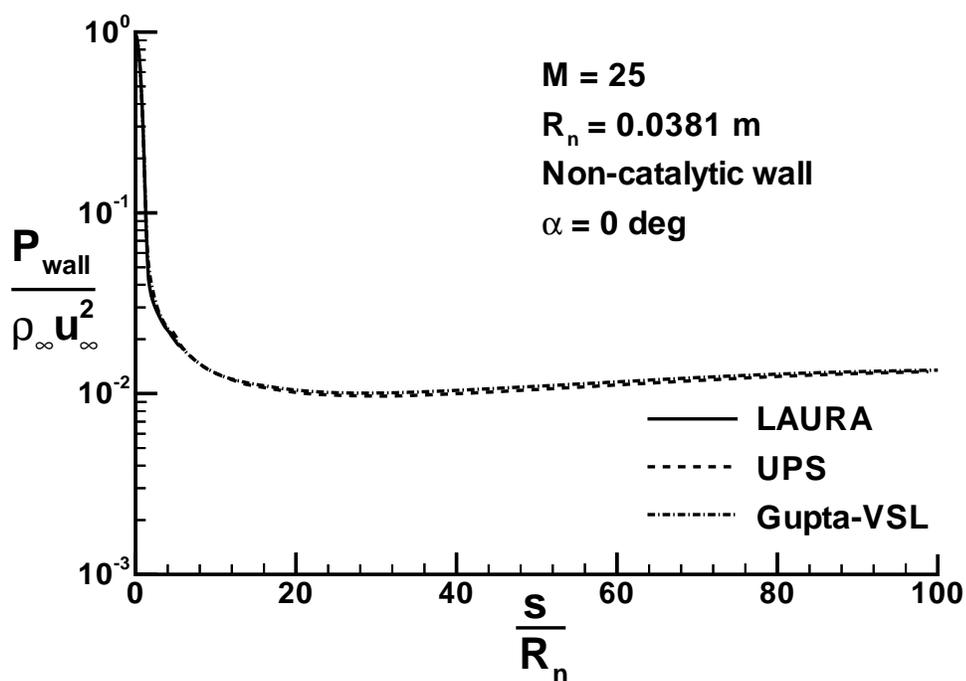


Fig. 4.4 Case 1 surface pressures to 100 nose radii.

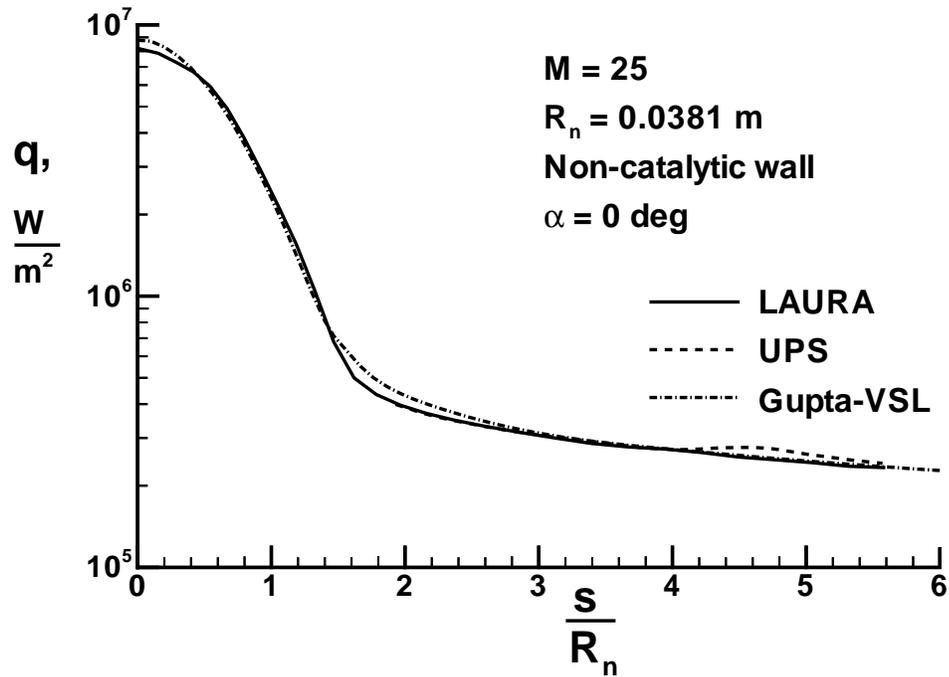


Fig. 4.5 Case 1 surface heating—the interface region.

Fig. 4.3, the LAURA solution was terminated at six nose radii.

Surface heat transfer results for LAURA, UPS, and Gupta-VSL are presented in Figs. 4.5 and 4.6. Figure 4.5 plots the interface region out to a distance of six nose radii. Similar trends are seen in the heating as were seen for the pressure in this region. The heating at the interface between the LAURA and UPS codes picks up smoothly, but there is a bump in the UPS heating between four and five nose radii, corresponding to the pressure bump discussed earlier. The Gupta-VSL heating is elevated above the LAURA-UPS heating in the region of the sphere-cone juncture.

Figure 4.6 carries the present method and Gupta-VSL heating out to 100 nose radii. Note that the LAURA heating terminates at six nose radii. A noticeable difference exists between the UPS and Gupta-VSL solutions that persists from the overexpansion region on downstream. The Gupta-VSL results are consistently 18–22 percent lower than the UPS heating. The VSL equations employ further assumptions on the governing equations than PNS, which may account for some of the heating difference, and while Figs. 3.1 and 3.2 show good agreement between the UPS and Gupta viscosities, there are differences in other aspects of the kinetic models which may also be contributing to a heating disparity.

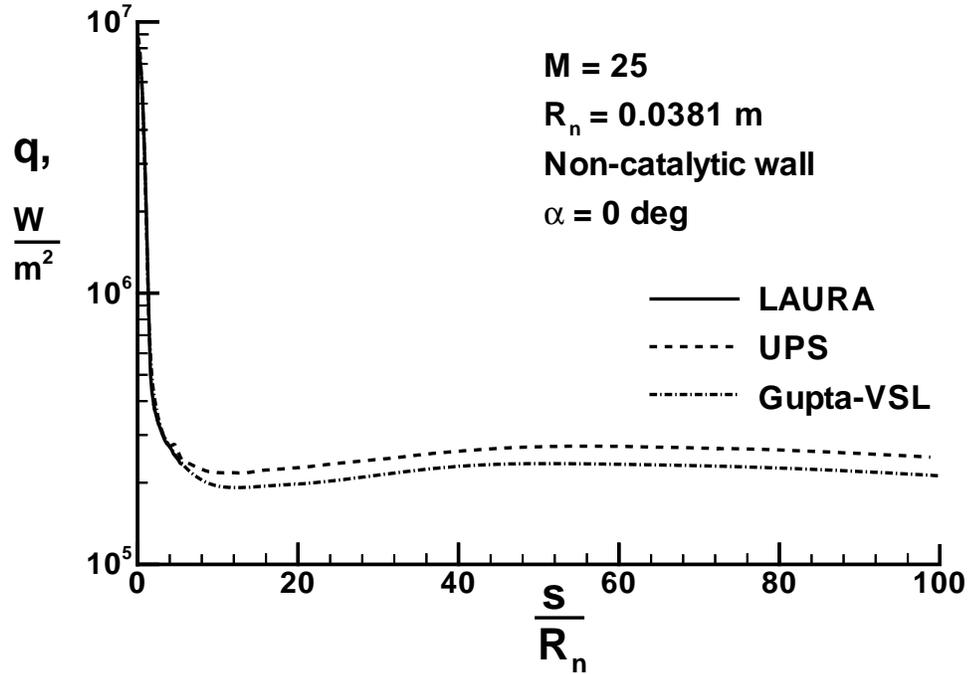


Fig. 4.6 Case 1 surface heating to 100 nose radii.

Looking specifically at reacting chemistry effects, Fig. 4.7 profiles the atomic oxygen mass fraction versus normal distance from the surface, as a fraction of the shock layer, at an axial position five nose radii from the nosetip. The profiles from the LAURA and UPS solutions are seen to be similar, with a difference in mass fraction at the surface of two percent. The mass fraction gradients at the surface are seen to be zero, as they should for the non-catalytic wall assumption.

This difference in oxygen mass fraction at the surface becomes critical in realizing the difficulties encountered in obtaining the combined LAURA-UPS solution for this particular configuration. While the Blottner and Kang reaction sets are similar or identical for most reactions, a significant difference in the equilibrium constant can occur in the equation controlling production of ionized nitric oxide,



Table 4.2 lists the forward and backward rates for Eqn. 4.4 from the two kinetic models. At a temperature of 1280 K, an average temperature for a Case 1 surface

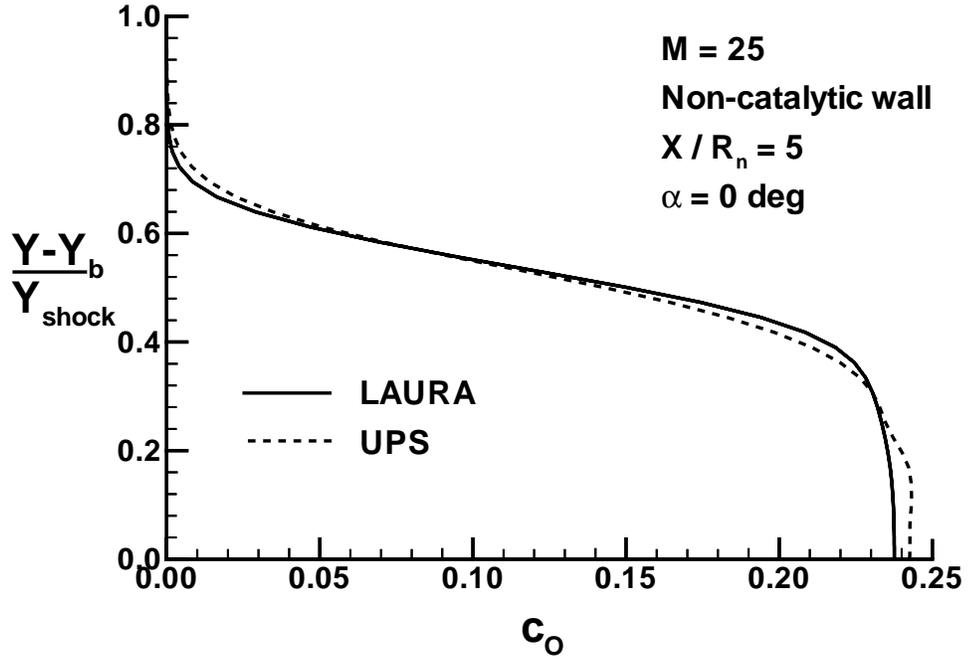


Fig. 4.7 Case 1 atomic oxygen mass fraction profiles at $X/R_n = 5$.

Table 4.2 Reaction rates for ionized nitric oxide.

	k_f	k_b
Kang (LAURA)	$1.4 \times 10^6 T^{1.5} \exp^{-\frac{31900}{T}}$	$6.7 \times 10^{21} T^{-1.5}$
Blottner (UPS)	$9.03 \times 10^9 T^{0.5} \exp^{-\frac{32400}{T}}$	$1.8 \times 10^{19} T^{-1.0}$

cell, the Blottner equilibrium constant,

$$k_{eq} = \frac{k_f}{k_b} \quad (4.5)$$

for this reaction is 2.33×10^{-16} , while the Kang equilibrium constant is two orders of magnitude lower at 6.58×10^{-18} . Under the flow conditions for this case, both atomic oxygen and atomic nitrogen concentrations at the surface are large, with the flow consisting of roughly equal parts atomic oxygen, atomic nitrogen, and molecular nitrogen near the wall. The net result is that the UPS solution produces significantly more ionized nitric oxide relative to the starting solution provided by LAURA, and at a fast rate. This creates a marching step-size stability restriction characterized by the Damköhler number,

$$D_a = \frac{\tau_{flow}}{\tau_{reactions}} = \frac{\text{available flow residence time}}{\text{time required for reaction equilibration}} \quad (4.6)$$

see [23], which is exacerbated by a tight grid spacing near the wall. Because of the no-slip viscous wall boundary condition the fluid velocity in the first cell outside the wall is very low. Coupling this with the extremely high aspect ratio of the cells in the wall region results in the flow having a relatively long residence time. However, due to the difference in equilibrium constants between UPS and LAURA, the present method produces fast reactions downstream of the interface, driving up the Damköhler number. The net result on the solution is to put a maximum value on the aspect ratio of the near-wall cells, which translates into a restriction on the marching step size of the present method. A schematic of the stability restriction is presented in Fig. 4.8

A compromise was sought whereby the LAURA grid was modified to double the cell size of the first grid cell, which sets a nominal cell Reynolds number of two at the wall. This was found to still allow accurate resolution of gradients at the wall while somewhat relaxing the Damköhler imposed marching stability restriction. In this case the non-linearity inherent in the chemical reactions allows for marching steps more than twice as large as were possible with a wall cell Reynolds number of one. Larger grid spacings at the wall were found to be too coarse to provide suitable LAURA solutions.

The LAURA solution for this case was converged through an L_2 norm of the residual of seven orders of magnitude in 2200 iterations. The total CPU time on a

$$D_a = \tau_{\text{flow}} / \tau_{\text{reaction}}$$

**Fluid velocity is low,
but reactions need only equilibrate
over a short vertical distance.**

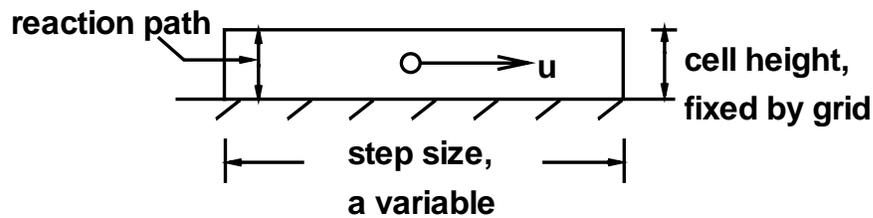


Fig. 4.8 Damköhler stability restriction.

Cray 2 was 1411 seconds. Figure 4.9 contains the convergence history of the LAURA solution. One caveat to this performance is that the solution was begun from a converged solution for a similar, but not identical, case. The sharp spikes occurring early in Fig. 4.9 are the result of grid adaptations, while the later spikes are due to shock ringing.

The UPS solution was obtained on a Cray YMP with a final marching step size of 0.25 mm. This is a small step size in relation to other cases which have been run with the present method, but is a result of the previously mentioned marching stability restrictions. Muramoto[22] reports using the same marching step size for a Mach 20, seven-degree sphere-cone nonequilibrium case, with a modified version of UPS, and Tannehill *et al.*[27] report using a step size of 0.2 mm with an axisymmetric cone. The full UPS solution to 100 nose radii required 4198 seconds.

4.1.1 Attempts at Larger Marching Steps

While the axisymmetric geometry of Case 1 was able to be solved by the present method in a reasonable amount of computer time with the small marching step size,

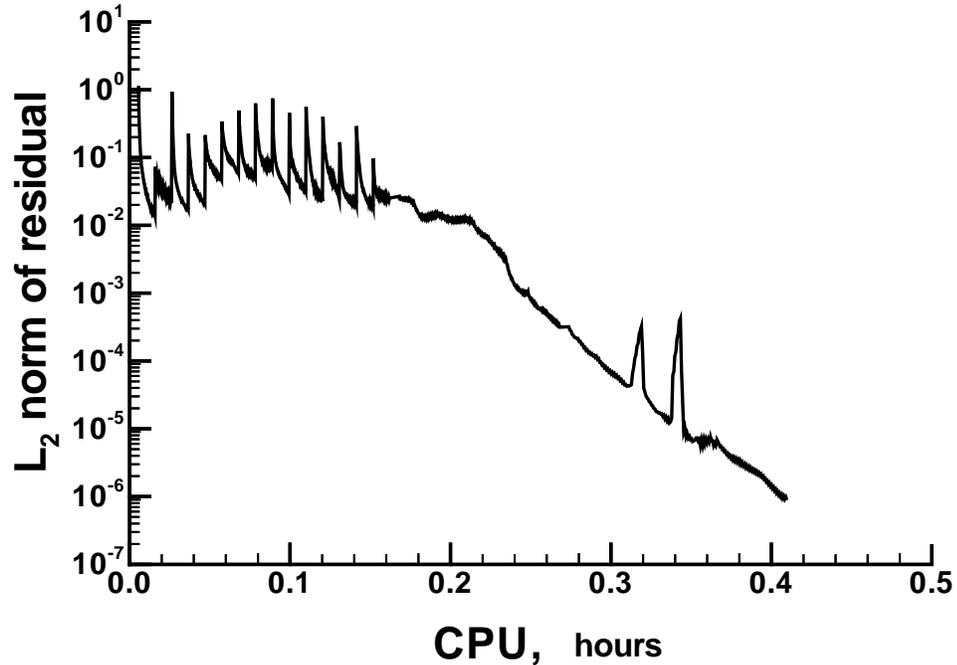


Fig. 4.9 Case 1 LAURA convergence history.

there is concern that a full-sized, three-dimensional vehicle might require excessive computational resources if conditions were such that the stability restriction observed here applied. Several attempts were made to enhance the stability of the UPS marching solution for the Case 1 conditions. Local chemistry iterations were added, second- and fourth-order subsonic smoothing terms were turned on, the safety factor applied to the Vigneron condition in Eqn. 3.22 was adjusted, and the eigenvalue stability parameters *EPSA* and *EPSS* in the UPS input file were changed. Some small stability improvement was found by increasing the values of the second-order implicit smoothing term, the Vigneron safety factor, and the stability parameter *EPSA*, but not enough to allow order-of-magnitude larger step sizes.

The UPS solution instability was typically manifested by a divergence of the cell temperatures at the wall. It was thought that the reflected boundary conditions, where the wall temperature is enforced only as the geometric average of the image cell temperature and the temperature of the first cell above the wall, might be contributing to the instability because the wall temperature is not explicitly enforced. The UPS boundary conditions were altered to apply the wall boundary conditions of no slip, no penetration, and fixed wall temperature at the image cell center, and a new solution

was obtained with both LAURA and UPS using this boundary condition, but no appreciable improvement in the stability of the present method was observed. An effort to enforce a limiter on the Newton iteration used to decode the temperature and pressure given the total energy, mixture density, and species concentrations also failed to produce a useful relaxation of the stability restriction on the marching step size.

Some attempts at solution smoothing and solution modulation were tried with the present method. Several approaches were attempted, beginning by trying to march the UPS solution one step, modifying the original interface plane with an under relaxation scheme by adding some fraction of the difference between the initial starting plane data and the first step solution, and repeating in a locally iterative procedure. The idea was to allow the solution to relax without creating excessive transients. The next attempt tried to march the UPS solution while modulating it with the LAURA solution, so that the first step was 10 percent UPS and 90 percent LAURA, the second step 20 percent UPS and 80 percent LAURA, and so on. While these attempts had some small success in delaying or postponing the instability with large step sizes, they were unable to suppress the instability enough to solve a significant portion of the geometry with large marching steps. More exotic solution modulation methods were tried whereby the UPS domain was split to allow the inviscid, viscous, and near-wall regions to relax from the LAURA solution at different rates, but the result was still the same—the marching step-size was limited to the millimeter range or less.

An attempt at a solution was made using the Park[23] kinetic model in LAURA, with no more justification than that it is a readily available option. Perhaps predictably, this did not produce any improvement in stability. The location of the interface point was varied as well, without producing a change in the behavior of the solution with the present method.

Changes to the grid included trying 40, 64, and 128 points in the body normal direction with nominal cell Reynolds numbers at the wall of 0.5, 1, 2, 5, and 10. The number of points did not seem to alter the solution appreciably for this configuration, but as discussed earlier the cell size at the wall proved to be very important. The tradeoff had to be made between a tight clustering at the wall for good gradient resolution and a more reasonable cell aspect ratio to allow feasible marching step

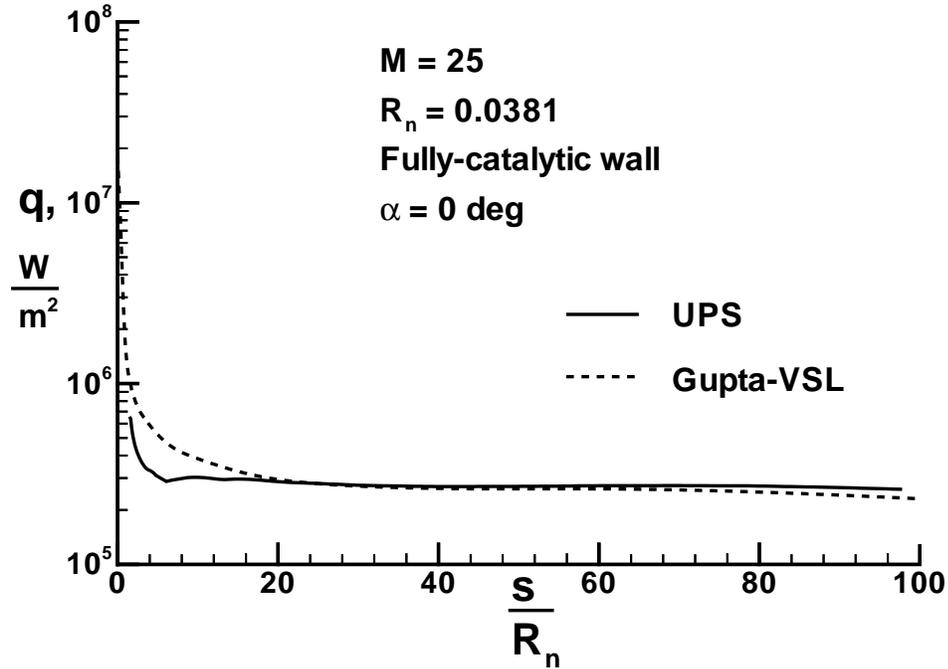


Fig. 4.10 Case 1 heat transfers for a fully-catalytic wall.

sizes.

A final parametric on the basic Case 1 solution was performed by employing a fully-catalytic wall instead of the non-catalytic boundary condition. The surface heat-transfer results for this case are presented in Fig. 4.10. For this solution a much larger step size was possible with UPS, because the fully-catalytic wall condition creates a different gas composition in the near-wall region which does not involve the ionized nitric oxide reaction, Eqn. 4.4, to the same degree as the non-catalytic solution. However, as can be seen in Fig. 4.10 the heating from the present method immediately downstream of the interface region does not look good, although the heat transfers agree well from 15–100 nose radii with the results of Gupta for the same configuration. Interestingly, the same behavior in the UPS heating near the interface point is reported by Muramoto in Fig. 11 of Ref. [22] for an axisymmetric, seven degree sphere-cone with a fully-catalytic boundary condition. In discussing his result, Muramoto further cites Buelow[4] as another investigator who has seen a similar heating behavior with UPS.

4.2 Case 2

Three-dimensional finite-rate chemistry solutions were sought for the Case 2 configuration for Mach 28 at two and five degree angles of attack. The wall boundary condition was set to be fully catalytic, which in both LAURA and UPS sets the wall species concentrations equal to their freestream values,

$$c_{s,wall} = c_{s,\infty} \quad (4.7)$$

Since the species concentration gradients are no longer zero at the wall for this case, as they were for the non-catalytic solutions, a computation of the diffusive heating rate was added to the UPS surface property output routine as,

$$q_{diffusive} = \frac{k L_e}{C_p} \sum_s h_s \left. \frac{\partial c_s}{\partial n} \right|_{wall} \quad (4.8)$$

Computations of the diffusive heating for the cases considered in the present study showed its contribution to the total heat transfer to be a very small percentage. In the calculations of Ref. [2] a variable wall temperature was employed, but for the present calculations a fixed wall temperature of 833 K was used. This was chosen as a rough average to use for comparison with the results of Bhutta.

4.2.1 Two Degrees Angle of Attack

The LAURA symmetry plane grid for this case is displayed in Fig. 4.11, for clarity showing only every eighth point in the body-normal direction. The full LAURA grid contains 51 streamwise cells, 18 circumferential cells, and 128 cells normal to the body. The UPS starting plane was extracted from the fifteenth streamwise cell in the LAURA solution.

Figure 4.12 contains both windside and leeside surface pressures, normalized by twice the freestream dynamic pressure, versus axial distance, normalized by the nose radius, for both the full-body LAURA solution and the coupled LAURA-UPS solution of the present method. The agreement is very good, with the most noticeable difference occurring at the cylinder-flare junction. The pressure jump at the flare is much more sharply defined with the UPS solution, whereas LAURA predicts a less abrupt pressure change. Part of this difference is attributed to the prevention

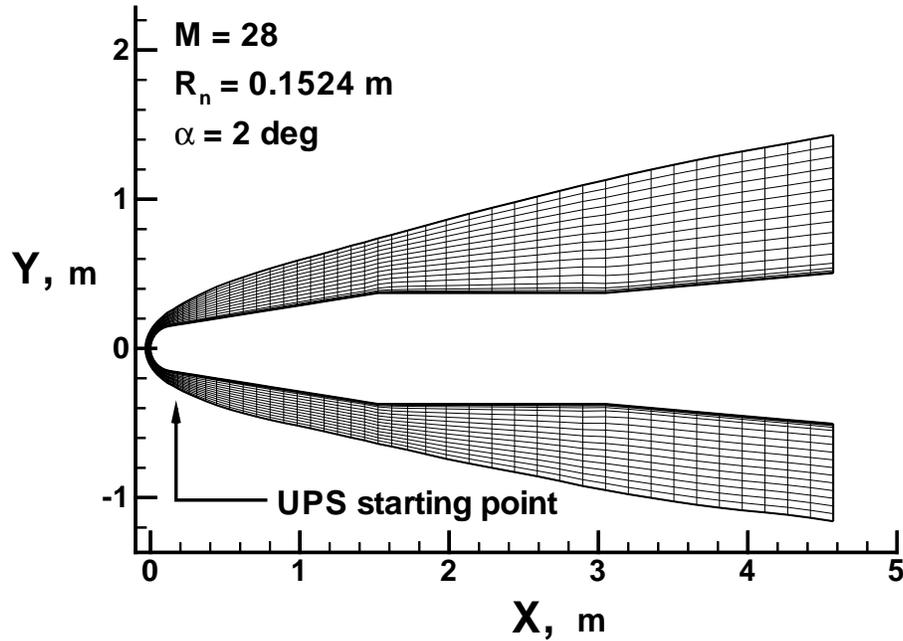


Fig. 4.11 LAURA Case 2 symmetry plane grid, showing every eighth body-normal point.

of upstream propagation of pressure waves in the subsonic portion of the boundary layer by the PNS code. Also contributing is a somewhat coarse LAURA grid in the streamwise direction at this point. The UPS solution was marched at a step size of 0.01 m, which is about one-tenth the streamwise length of the corresponding LAURA cells at the cylinder-flare junction. Remember though that LAURA is second-order accurate in the streamwise direction, while UPS is only a first-order algorithm in the marching direction. It can be seen that with the present method UPS picks up the pressure accurately from the LAURA solution at the interface region, located at two nose radii.

A mirrored pressure contour plot is presented in Fig. 4.13. The left half of the solution is from LAURA while the right side is the UPS solution. Both solutions are taken from a cross section at 29 nose radii. It can be seen that the UPS bow shock is crisper than the LAURA bow shock. This feature holds true in general, and is a result of the different types and levels of numerical dissipation used in each scheme. Reasonable agreement is seen between the two solutions, 27 nose radii downstream

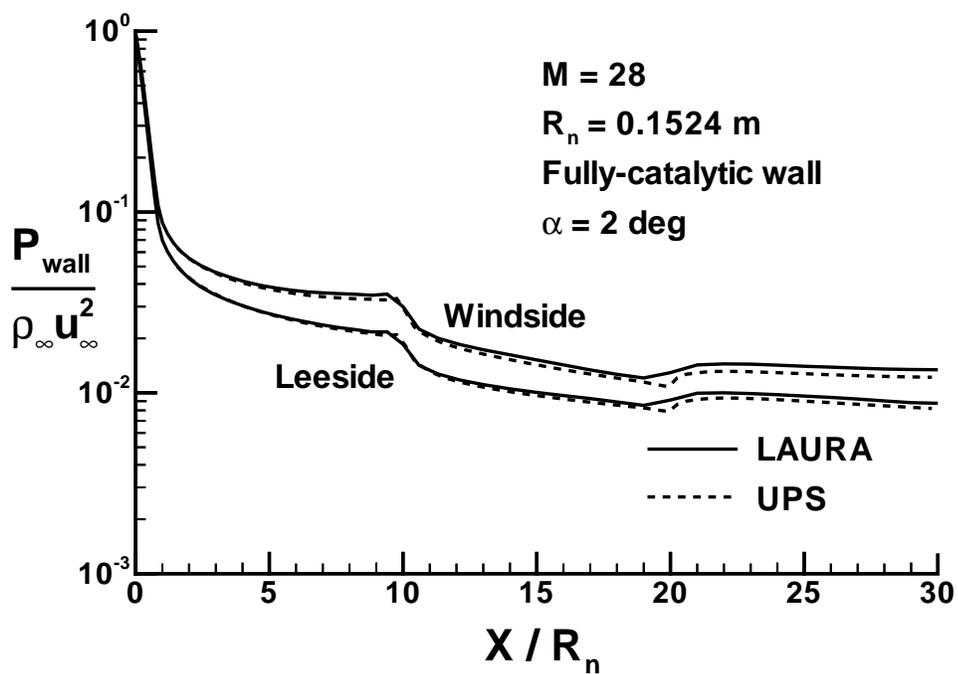


Fig. 4.12 Case 2 surface pressures, windside and leeward.

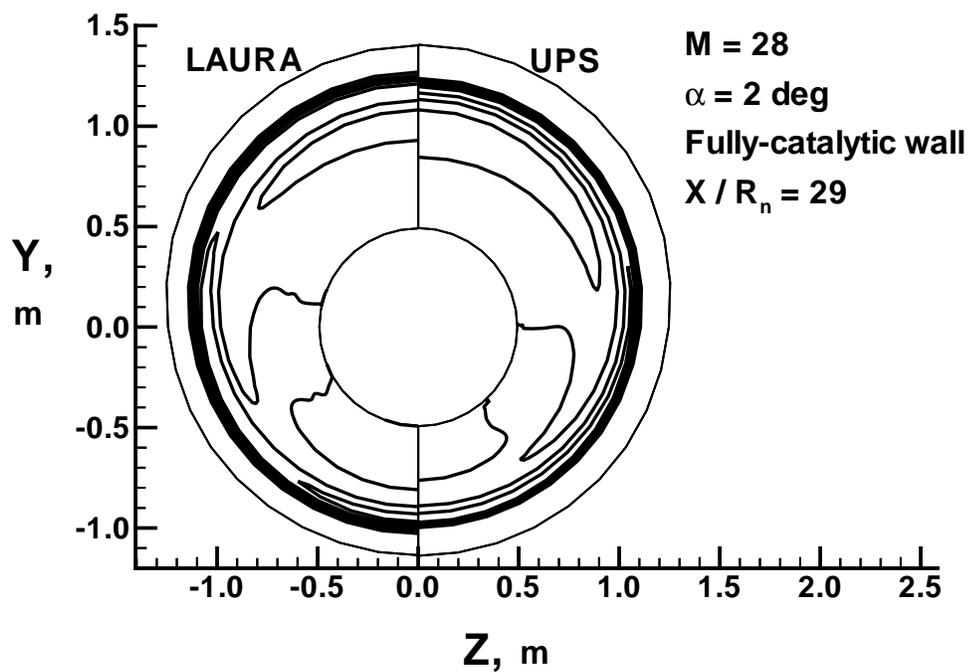


Fig. 4.13 Case 2 pressure contours in the crossflow plane at $X/R_n = 29$.

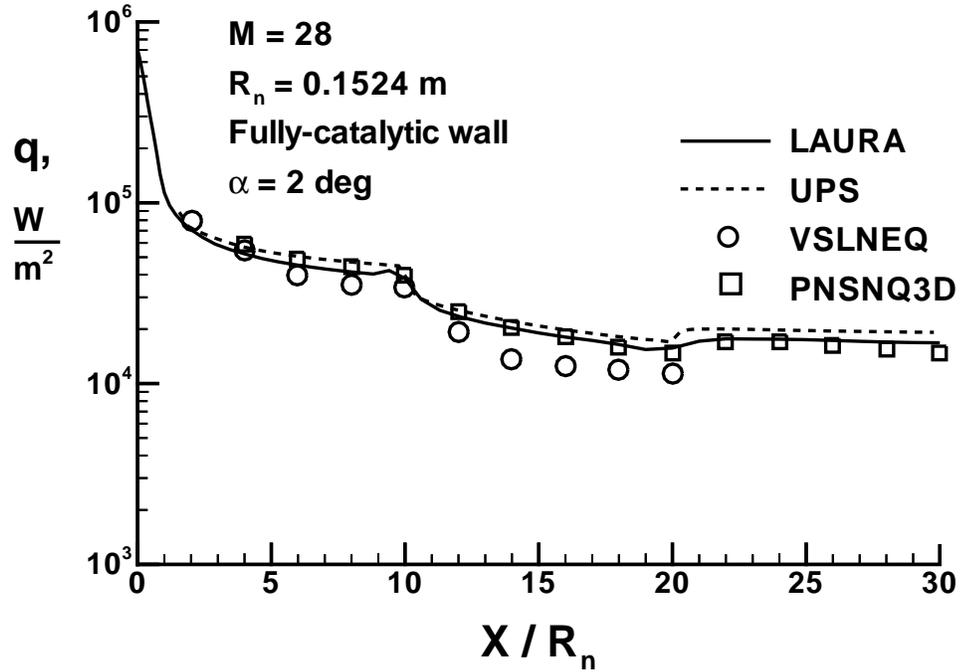


Fig. 4.14 Case 2 windside heat-transfer rates.

of the interface point.

Axial surface heat transfers are plotted in Figs. 4.14 and 4.15. Along with the results from the two codes in the present method, heat transfer results from Ref. [2] for a nonequilibrium PNS code, PNSNQ3D, and a nonequilibrium VSL solver, VSLNEQ, are presented for comparison. In Fig. 4.14 very good agreement is seen with the present method, as the distribution in windside heating spans 5–10 percent between LAURA, UPS, and PNSNQ3D over the vehicle body. The VSLNEQ results are as much as 35 percent lower than the other solutions on the cylinder. Looking at the interface region, the UPS heating is seen to pick up very well from the LAURA starting solution. At the juncture between the cylinder and the flare, the UPS solution is seen to capture a more abrupt change in heating than the LAURA solution. As was the case with the surface pressure, the cause of this difference is attributed to the suppression of upstream information propagation by the space marching scheme and axial smearing by the LAURA grid.

The corresponding leeside heat transfers are plotted in Fig. 4.15. Leeside heating for this case, with a two degree angle of attack, is 40 percent lower than the windside heating. The same trends between the four solutions are seen on the leeside as on the

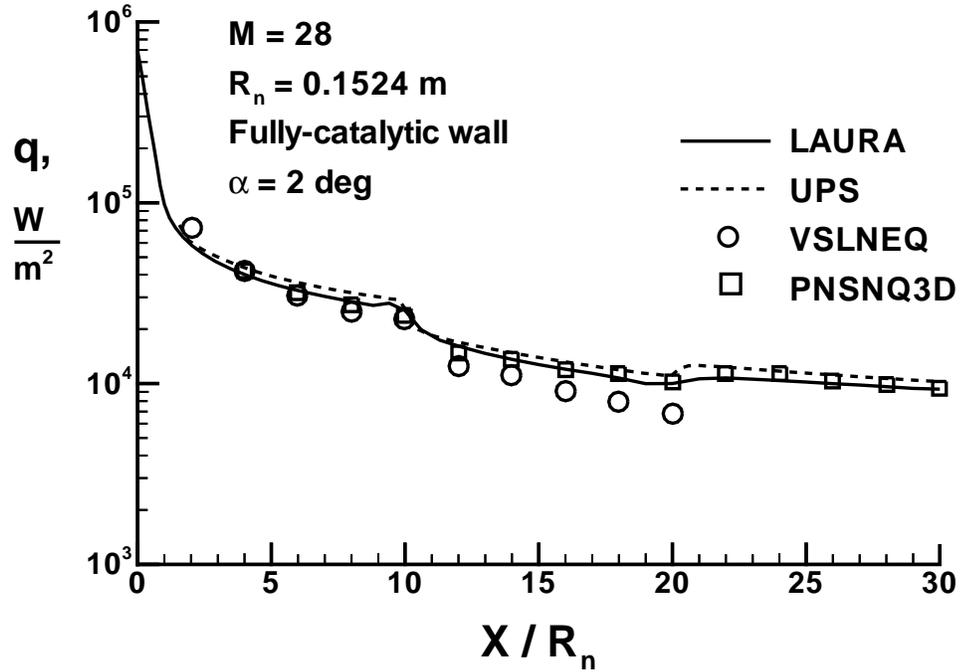


Fig. 4.15 Case 2 leeside heat-transfer rates.

windside, with even slightly better agreement. The LAURA and UPS solutions on the leeside agree to within nine percent. Again, the VSLNEQ results are lower than the other heating rates, which is similar to the results seen with Case 1 in Fig. 4.6 between the Gupta-VSL and UPS heat transfers.

4.2.2 Timing

It was seen earlier that for the non-catalytic wall conditions in Case 1 the present method was limited in its computational advantage over the full TLNS solution by a marching stability step-size restriction. This is not the case for the fully-catalytic surface of Case 2. Since the species concentrations are forced to return to freestream values at the wall, there is very little atomic oxygen and atomic nitrogen at the wall, and hence the reaction controlling production of ionized nitric oxide, Eqn. 4.4, is not the factor it was in Case 1. Much larger marching step sizes were able to be taken for the sphere-cone-cylinder-flare configuration than for Case 1, and a substantial reduction in solution time was achieved with the present method over a full TLNS solution. Both the LAURA and UPS solutions for this case were obtained on a Cray

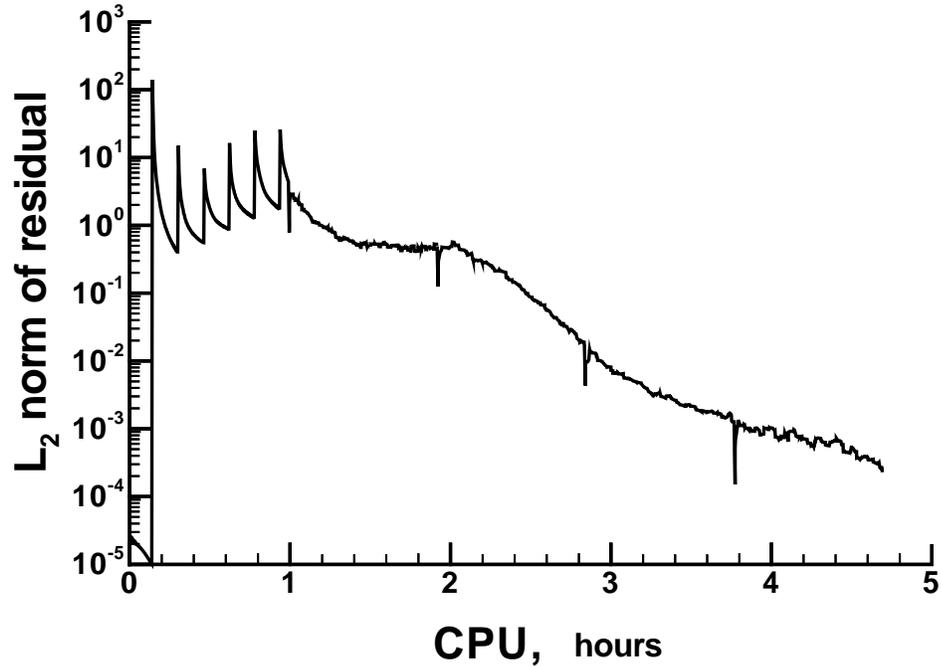


Fig. 4.16 Case 2 LAURA convergence history.

YMP. Figure 4.16 tracks the convergence history of the LAURA solution for this case. In this plot the residual starts out small and then jumps up abruptly. This is part of the initialization and restart procedure, and does not represent a converged solution. As with Fig. 4.9, the spikes in the convergence history during the first hour are the result of grid adaptations. The later spikes are associated with the multi-tasked restart procedure in LAURA. The total LAURA solution CPU time was 4.73 hours, requiring 25 megawords of memory.

With a marching step size of 0.01 m, two orders of magnitude larger than were possible for Case 1, the UPS portion of the solution was obtained in only 776 CPU seconds, and required only 2.15 megawords of memory. This represents an order of magnitude reduction in both time and memory over the full-body LAURA solution. Results presented in the next section show that, with a slight reduction in solution resolution, the present method can achieve results with even five times less CPU time.

4.2.3 Grid Convergence

Body-normal distribution

The effect of grid resolution in the body-normal direction was investigated for both the LAURA and UPS Case 2, two degree angle of attack solutions. The two grids used the same number of cells in the streamwise and circumferential directions, but had 64 and 128 cells in the body-normal direction, respectively. The wall cell size for the 128 cell solutions was set to be half that of the 64 cell solutions, so as to maintain the same grid stretching.

Figures 4.17 and 4.18 plot the windside and leeside centerline heat transfer rates from the two LAURA solutions. Heat transfer rates, being a gradient of the numerical solution, are particularly sensitive to variations in the solution, and are thus considered a good indication of how well a calculation has converged. In both of these plots the LAURA solution can be seen to vary by 20–25 percent in the heating between the two grids. Obviously, for these particular conditions LAURA is not grid converged with 64 points in the body-normal direction. As seen before in Figs. 4.14 and 4.15, the 128 point LAURA solution agrees well with the UPS and PNSNQ3D solutions, so the baseline Case 2 results use the 128 point LAURA solution.

The corresponding UPS heat transfer rates for the two grids are presented in Fig. 4.19, for the windside centerline, and Fig. 4.20, for the leeside centerline. It is immediately apparent from these heating plots that the UPS solution was grid converged with the 64 point grid. On both the windside and leeside there is a difference in heating between the grids in the immediate vicinity of the interface, but this is because the two solutions were started from the corresponding LAURA solutions, which displayed a significant difference in heating on the two grids. It is interesting to note that while the starting planes for the two UPS solutions were different, within only five nose radii downstream the UPS solutions have converged, indicating that that UPS is relatively robust with regards to the blunt-nose starting solution and grid distribution. It had been expected that LAURA would have been grid converged with fewer points than UPS, since it solves a higher-order set of equations, TLNS *vs.* PNS, and employs a second-order accurate numerical scheme in all three dimensions. Figures 4.17–4.20 show clearly that, in fact, the opposite is true for this particular case.

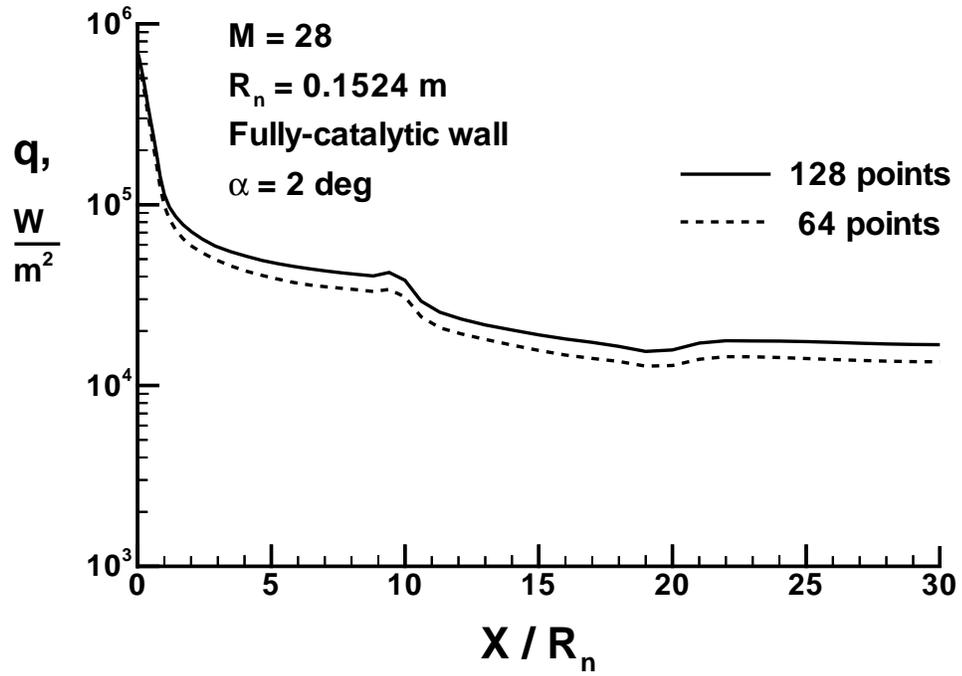


Fig. 4.17 Case 2 body-normal grid resolution: LAURA windside heating.

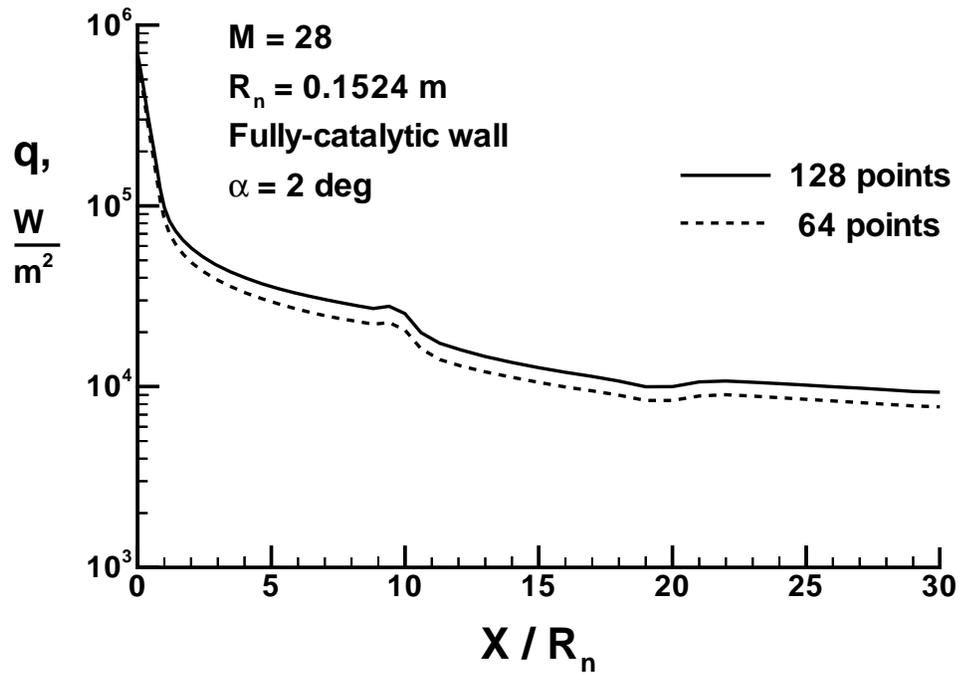


Fig. 4.18 Case 2 body-normal grid resolution: LAURA leeside heating.

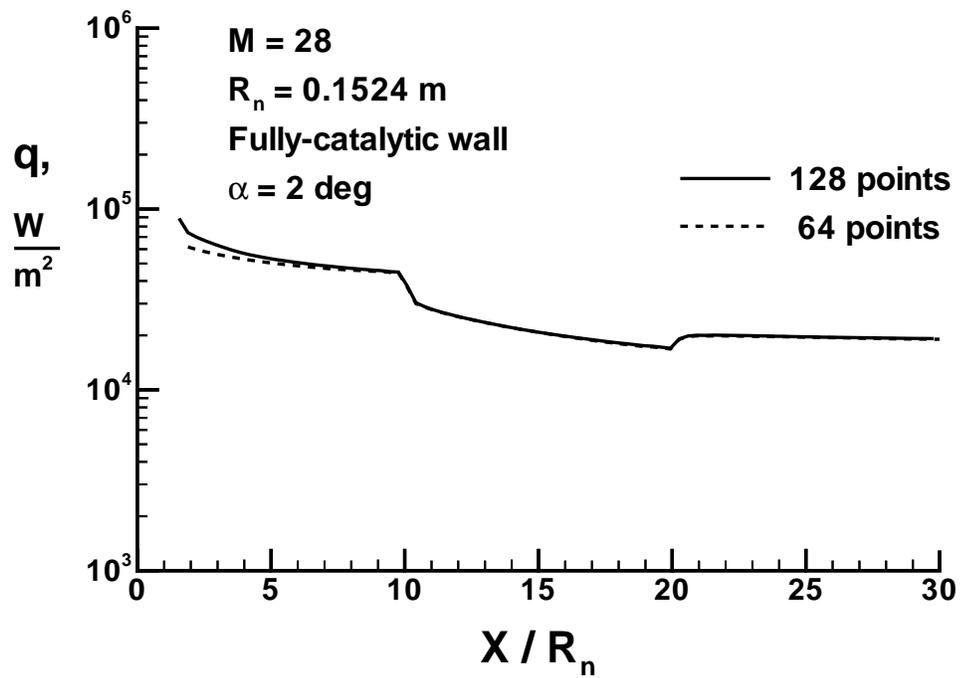


Fig. 4.19 Case 2 body-normal grid resolution: UPS windside heating.

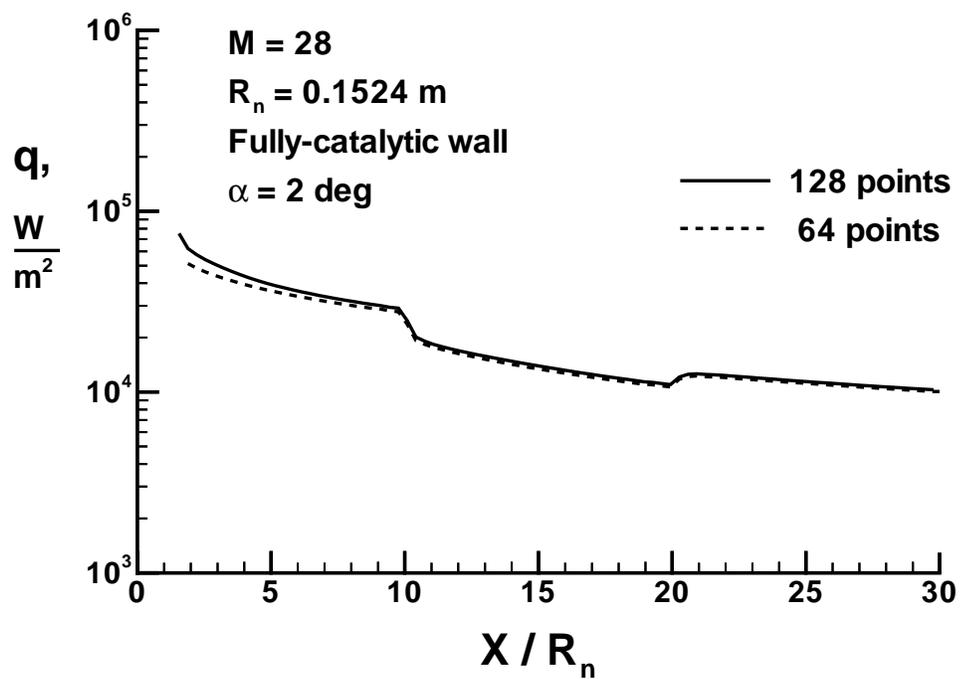


Fig. 4.20 Case 2 body-normal grid resolution: UPS leeside heating.

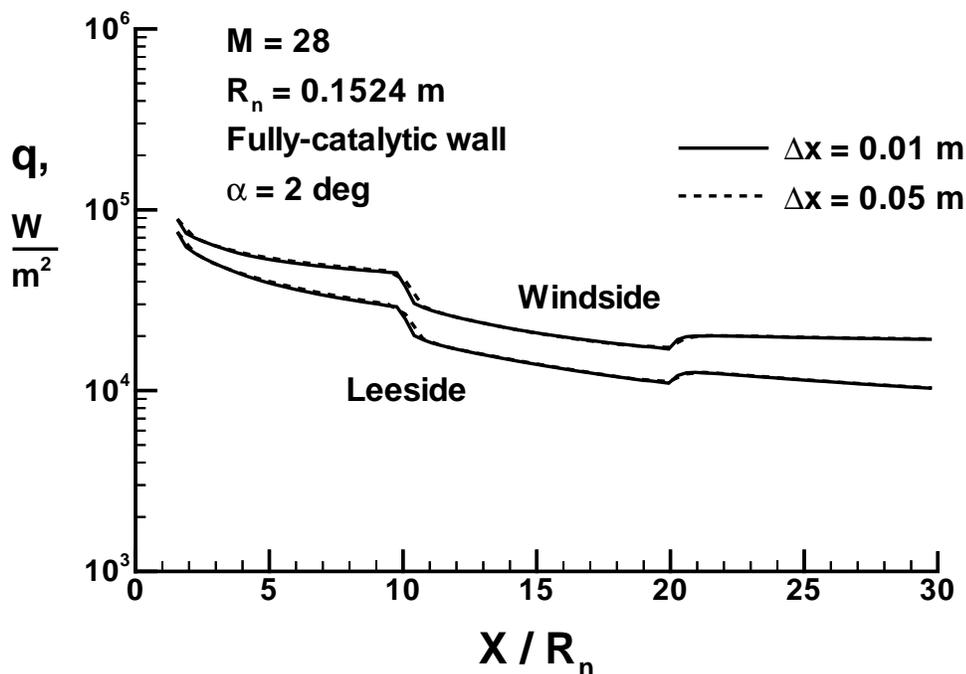


Fig. 4.21 UPS Case 2 marching step size convergence check—heating.

Marching step size

The grid convergence of the UPS marching step size in the present method for Case 2 was investigated by repeating the solution with a step size of 0.05 m, five times larger than was used for the baseline solution. Windside and leeside centerline comparisons of heat transfer, Fig. 4.21, and surface pressure, Fig. 4.22, are presented for both step sizes. Clearly, the baseline UPS portion of the present method's solution is grid converged with respect to marching step size at 0.01 m. The cone-cylinder and cylinder-flare junctions are slightly better resolved for both the heating and surface pressure for the smaller, 0.01 m, step size solution, which would be expected. The 0.05 m step size UPS solution was obtained with 86 steps in 169 CPU seconds on a Cray YMP.

4.2.4 Comparison with Perfect Gas

The Case 2 calculations were repeated using a perfect gas air model, rather than the nonequilibrium model. Figure 4.23 contains the windside and leeside heat transfer rates and Fig. 4.24 displays the corresponding surface pressures. Note that for this

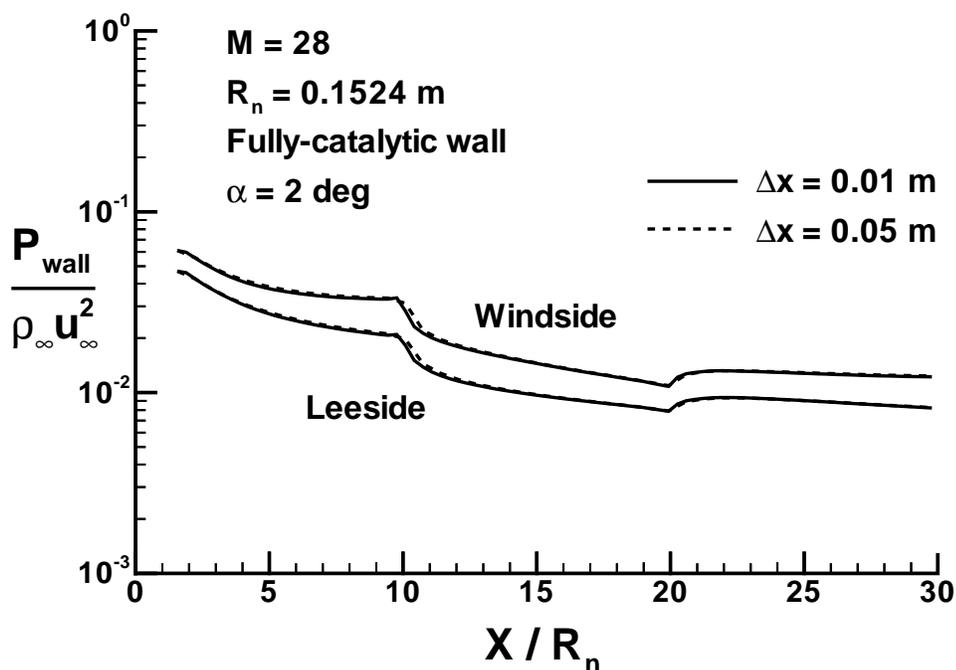


Fig. 4.22 UPS Case 2 marching step size convergence check—surface pressure.

calculation the LAURA solution was performed only on the sphere-cone portion of the geometry, and not the entire vehicle. The UPS solution was started from the sphere-cone junction point and extended the whole length of the body.

Comparing Figs. 4.14, 4.15, and 4.23, the effect of chemical nonequilibrium on the surface heat transfer can be seen for this case. The perfect gas heating is seen to be 20-25 percent lower than the reacting flow heat-transfer results. This is a consequence of the fully-catalytic wall boundary condition in the chemical nonequilibrium solution. The dissociated and ionized reactants from the shock transport chemical potential energy through the boundary layer, and then release the energy at the wall during exothermic recombination, leading to the elevated heat transfer predictions from the chemical nonequilibrium calculation, relative to the perfect gas solution.

Looking at Figs. 4.12 and 4.24, it is seen that chemical nonequilibrium does not have much of an effect on the surface pressure for this case, as the perfect gas pressures are only 2-3 percent higher than the corresponding nonequilibrium pressures. For this particular case then it appears that a perfect gas calculation might suffice for determining the aerodynamic characteristics, while a nonequilibrium solution is required to accurately predict the thermal environment. This is not a general result,

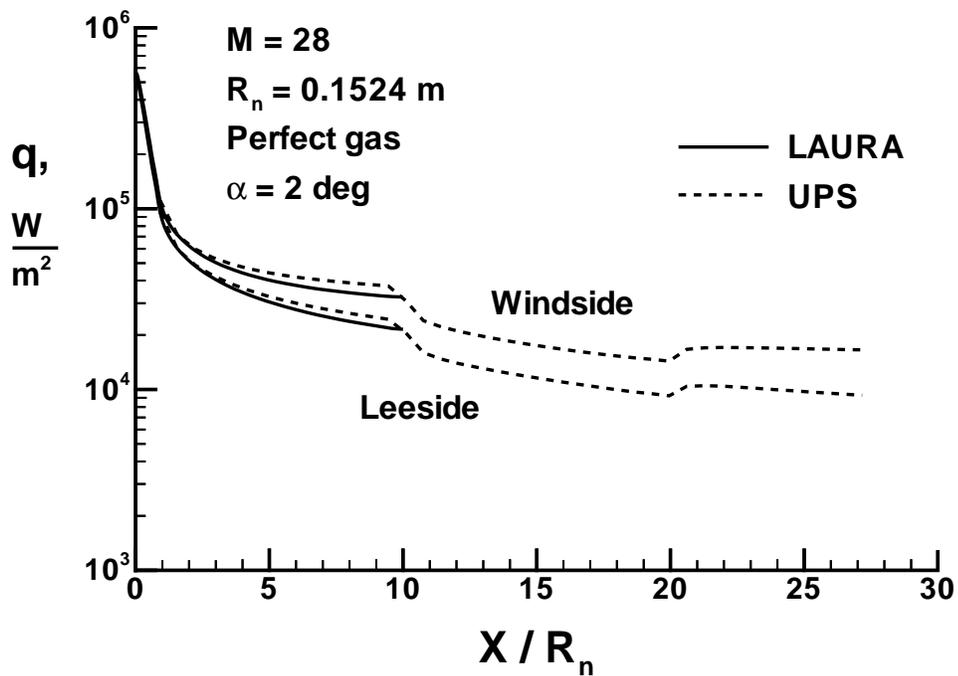


Fig. 4.23 Perfect gas heating at Case 2 conditions.

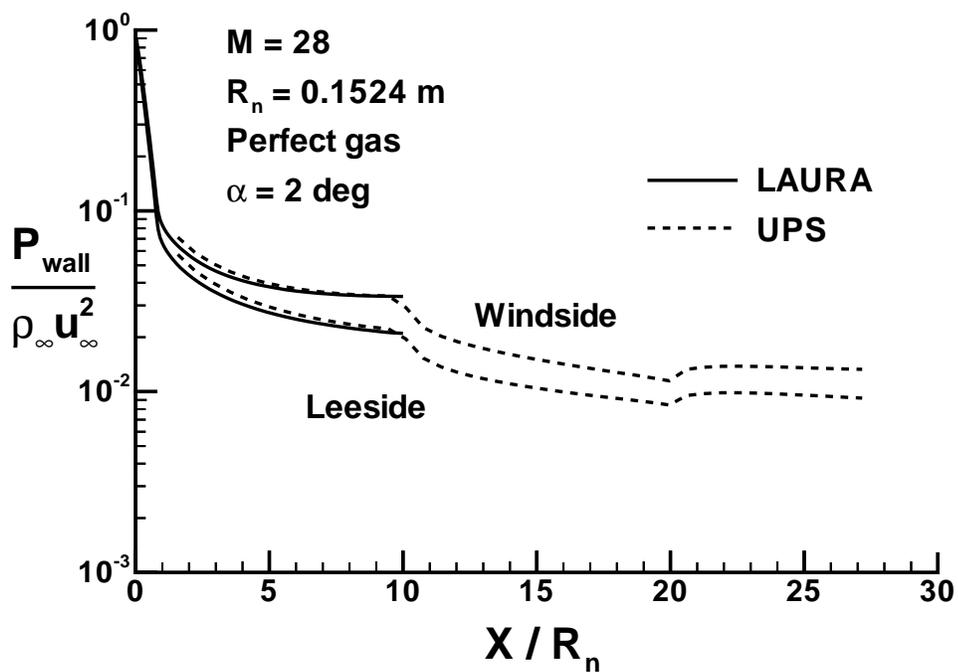


Fig. 4.24 Perfect gas surface pressures at Case 2 conditions.

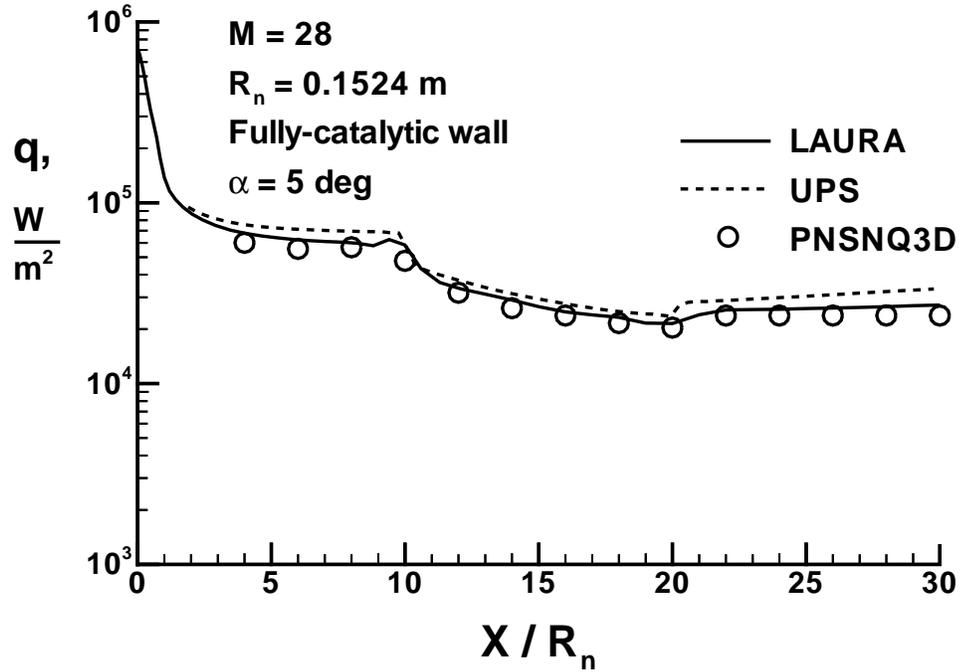


Fig. 4.25 Windside heat transfers at five degrees angle of attack.

with a noticeable exception being the space shuttle orbiter for which a nonequilibrium solution was required at a similar Mach number in order to predict the aerodynamics correctly[32].

4.2.5 Five Degrees Angle of Attack

A further nonequilibrium air solution for the PNSNQ3D code is presented in Ref. [2] for the Case 2 configuration, but at five degrees angle of attack. Figures 4.25 and 4.26 present the corresponding windside and leeside heat transfers, respectively, for the present method along with the results of Bhutta, who only reported a PNS solution and not a VSL solution for this configuration.

Leeside agreement is excellent between all three solutions, with the present method matching the full-body LAURA solution to within three percent. The windside agreement is fair, though not as good as the leeside. On the windside centerline the UPS heating is seen to be 10 percent higher on the cone, seven percent higher on the cylinder, and 18 percent higher on the flare. Also, the UPS and LAURA heating trends appear to be separating at the tail end of the body.

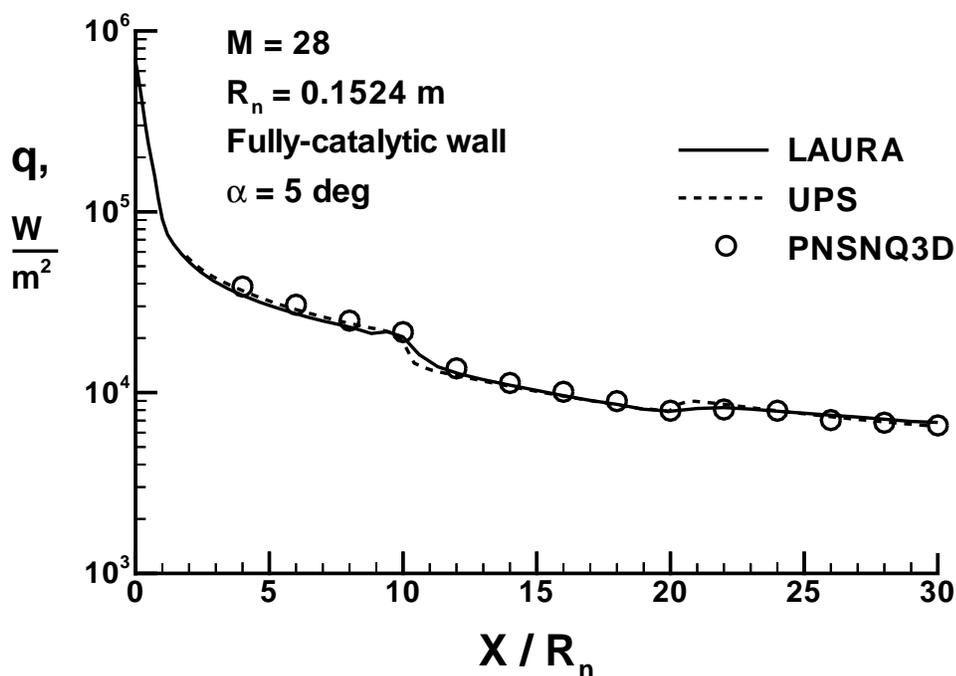


Fig. 4.26 Leeward heat transfers at five degrees angle of attack.

The windside and leeward surface pressures from LAURA and UPS are shown in Fig. 4.27. For the surface pressure excellent agreement is seen on the windside, while good agreement is seen on the leeward, a slightly different trend than for the heating. Also, the two solutions are in very good agreement on the surface pressure at the end of the body, as contrasted with the windside heating trend in Fig. 4.25.

The LAURA convergence history is plotted in Fig. 4.28. Again, the early spikes in Fig. 4.28 result from grid alignments while the later spikes are caused by the multi-tasked restart procedure in LAURA. The LAURA solution was achieved with some difficulty for this case. Because of the strong expansion on the leeward at the cone-cylinder junction the solution had to be relaxed very conservatively to maintain stability. The full-body solution required 20 CPU hours on a Cray YMP. By contrast, the UPS solution required only 800 CPU seconds on the same machine, nearly two orders of magnitude less time.

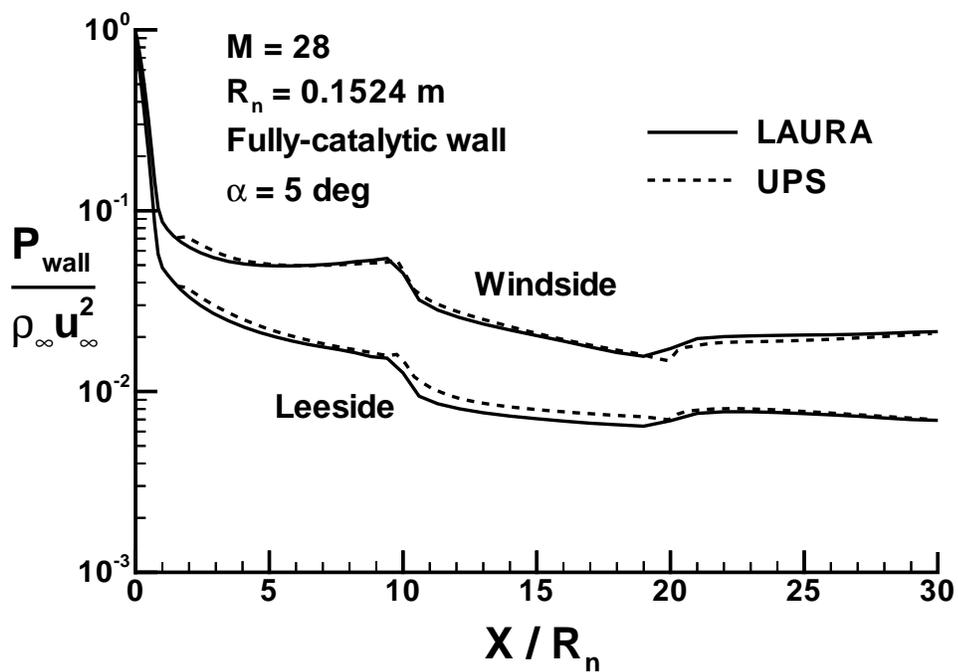


Fig. 4.27 Surface pressures for Case 2, five degrees angle of attack.

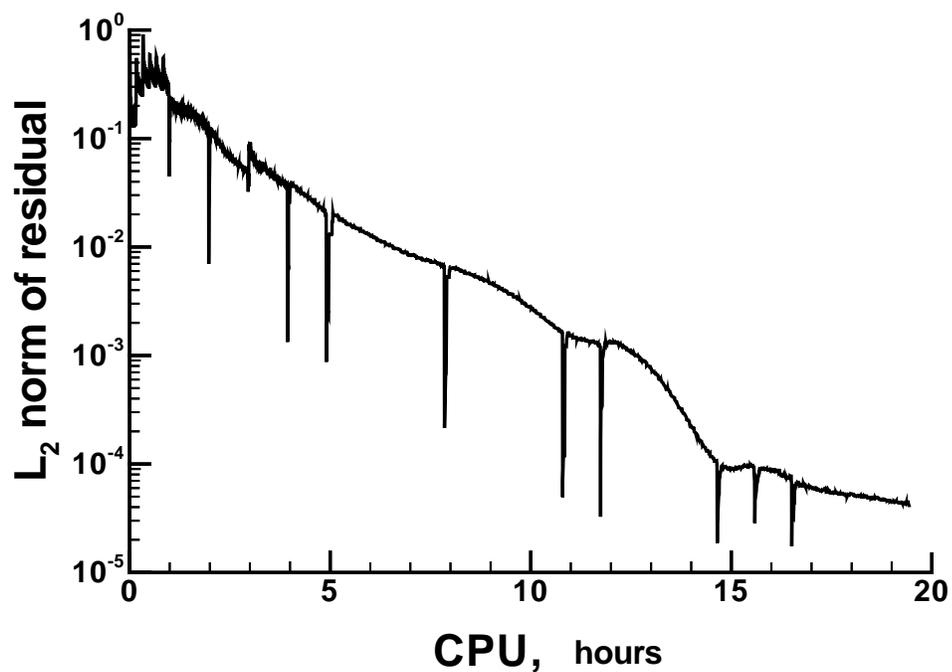


Fig. 4.28 LAURA convergence history for Case 2, five degrees angle of attack.

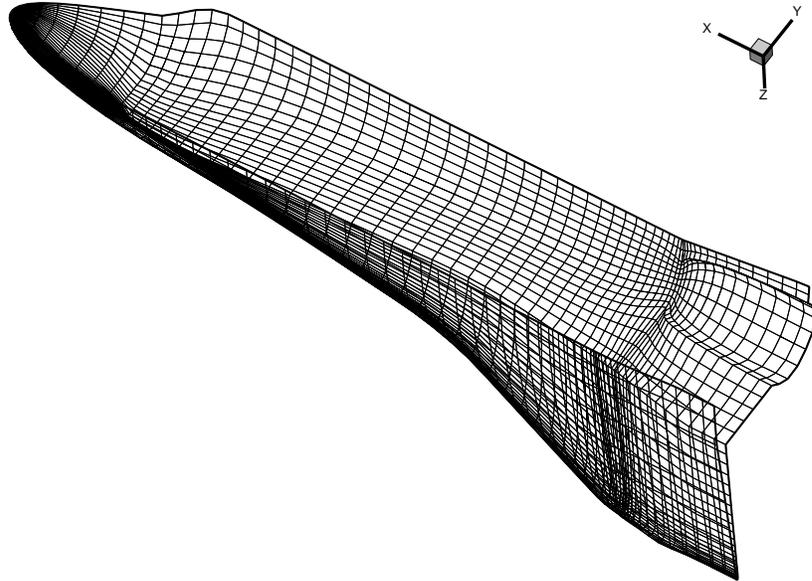


Fig. 4.29 Surface geometry of shuttle orbiter.

4.3 Shuttle: A Case for Future Work

A brief effort was made to apply the present method to a flight case of the Space Shuttle. A LAURA solution was available corresponding to the point in flight STS-28 studied by Kleb and Weilmuenster[17]. The nonequilibrium LAURA solution at Mach 24.3 and 39.4 degrees angle of attack had been performed on a 128 x 100 x 60 grid. The altitude was 73.2 km and the wall temperature was fixed at 1100 K. A plot of the orbiter geometry is presented in Fig. 4.29.

Forty degrees is a very large angle of attack for a PNS code to handle. Also, the LAURA solution was performed with a finite-rate wall catalycity. This capability is not available in the present version of UPS. The non-catalytic wall boundary condition was chosen for UPS as being the closest of the two options to the LAURA results. With the mismatched chemistry at the wall, the UPS solution was severely limited by step-size stability constraints, similar to what was seen in Case 1. Due to these stability problems, a UPS solution was only able to be achieved for a half meter section of the geometry, in the region behind the canopy and ahead of the wings. For Case 1 it was seen that the stability difficulties could be circumvented by taking

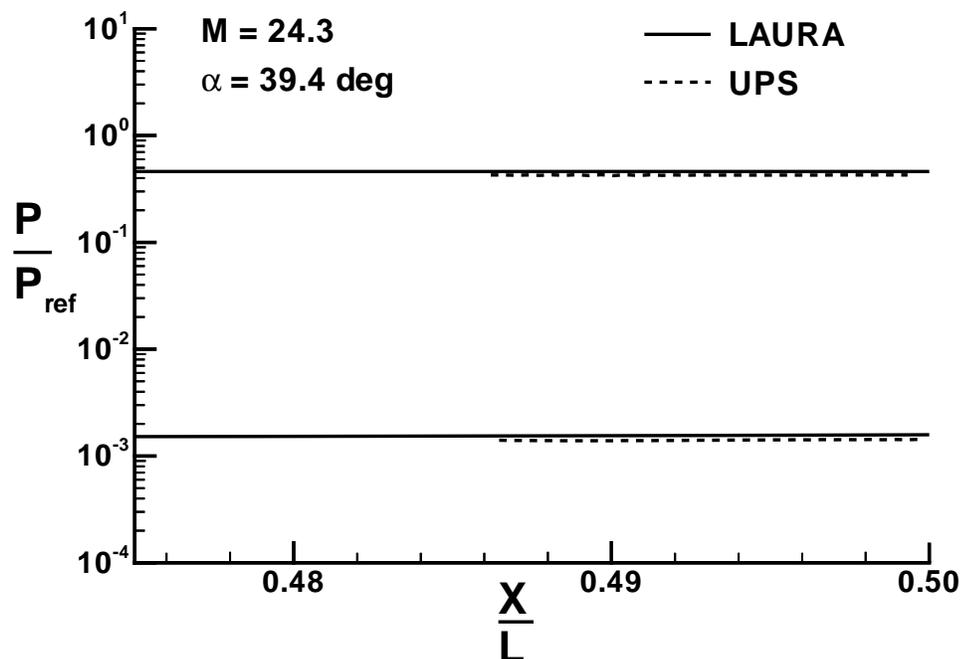


Fig. 4.30 Shuttle windside and leeside surface pressures.

marching step sizes in the sub-millimeter range. In the case of the orbiter, the sheer size of the three-dimensional grid made these small step sizes prohibitively expensive.

Figure 4.30 plots the windside and leeside centerline surface pressures for the LAURA and UPS solutions. At this location on the orbiter the streamwise pressures are relatively constant. The UPS pressures are seen to be 5–10 percent lower than the LAURA pressures on both the wind and lee sides. This could be due to the differences in wall catalycity, as exothermic recombinations will release energy which drives up the temperature and thus the pressure. However, a competing effect occurring at the same time is that the recombination of atoms into molecules reduces the moles of gas particles and thus the pressure. The reference quantities “L” and “ P_{ref} ” in Fig. 4.30 have been intentionally left unspecified.

While a full, stable solution was not able to be obtained for this case with the present method, there are some indications of the potential benefits and future directions to take. The LAURA solution requires 170 megawords of computer memory and took 190 seconds to perform a single iteration on a Cray YMP, with thousands of iterations being required to converge a solution. Clearly, this type of solution is a one-time calculation even on the largest and fastest computers available today. The

UPS solution, by contrast, requires only 2.3 megawords of memory, and runs at about 20 seconds per step.

For the present calculation a step size of 0.0125 m was used for the UPS solution. While this size of a step allows for a timely solution, it is seen that the solution is not stable, with a possible cause being the differences between the finite-rate wall catalycity in the starting plane and the non-catalytic boundary condition. This points to a future area of work in adding a finite-rate catalycity option to UPS. Another area of work could be to alter either the UPS or LAURA kinetic models to match more exactly, but this would involve extensive reprogramming, and other possibilities or codes should be considered before making that step.

Chapter 5

CONCLUDING REMARKS

A new procedure has been implemented for the aerothermodynamic solution of hypersonic, chemically-reacting air flowfields that combines two proven, existing solvers. The robustness of the thin-layer Navier-Stokes solver LAURA has been joined with the speed of the parabolized Navier-Stokes solver UPS. The class of vehicles to which the method is applicable are blunt-nosed configurations with slender afterbodies. The method offers the potential benefits of obtaining efficient solutions with second-order accuracy in the crossflow planes, while requiring only a fraction of the computer time and memory that a full-body LAURA solution would require.

Surface pressure and heat transfer results from the present method compare well with the baseline LAURA solution for the first case considered, an axisymmetric six degree sphere-cone at Mach 25. The downstream solution to 100 nose radii with the present method compares well with the surface pressure of a viscous shock-layer solution, but the viscous shock-layer heating is as much as 20 percent lower than the present method. For the non-catalytic wall boundary condition it was found that the differences in chemistry models between LAURA and UPS created a stability restriction on the marching step size of the UPS solution, which tended to offset the decrease in solution time expected with a marching scheme.

The second case considered, a blunted multi-conic at Mach 28, showed good agreement between the present method and an all-body LAURA solution for surface pressures and heat transfers. Results are obtained at both two and five degree angles of attack. The effect of grid resolution was investigated in the body-normal direction for both UPS and LAURA, and in the streamwise direction for the UPS solution. A comparison is also made between the chemical nonequilibrium results and a perfect gas solution. This case employed a fully-catalytic wall boundary condition, and did not encounter any stability restriction on marching step sizes. A significant reduction in both computer memory and solution time is demonstrated with the present method over an all-body thin-layer Navier-Stokes solution.

The present method is shown to be a fast, efficient procedure for obtaining aerothermodynamic predictions on blunted, slender vehicles at hypersonic speeds with reacting air flowfields.

BIBLIOGRAPHY

- [1] Dale A. Anderson, John C. Tannehill, and Richard H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis, 1984.
- [2] B. A. Bhutta, C. H. Lewis, and F. A. Kautz. A Fast Fully-Iterative Parabolized Navier-Stokes Scheme for Chemically-Reacting Reentry Flows. AIAA Paper 85-0926, June 1985.
- [3] F. G. Blottner, M. Johnson, and M. Ellis. Chemically Reacting Viscous Flow Program for Multi-component Gas Mixtures. SC-RR 70-754, December 1971.
- [4] P. Buelow, J. Ievalts, and J. Tannehill. Comparison of Three-Dimensional Nonequilibrium PNS Codes. AIAA Paper 90-1572, June 1990.
- [5] P. Buelow, J. Tannehill, J. Ievalts, and S. Lawrence. A Three-Dimensional Upwind Parabolized Navier-Stokes Code for Chemically Reacting Flows. AIAA Paper 90-0394, January 1990.
- [6] F. McNeil Cheatwood and R. A. Thompson. The Addition of Algebraic Turbulence Modeling to Program LAURA. NASA TM 107758, April 1993.
- [7] T. A. Edwards. CFD Analysis of Hypersonic, Chemically Reacting Flow Fields. In *Theoretical and Experimental Methods in Hypersonic Flows*, pages 1.1-1.12. AGARD, April 1993.
- [8] P. A. Gnoffo. An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows. NASA TP 2953, February 1990.
- [9] P. A. Gnoffo. Code Calibration Program in Support of the Aeroassist Flight Experiment. *Journal of Spacecraft and Rockets*, 27(2):131-142, March-April 1990.

- [10] Peter A. Gnoffo. Point-Implicit Relaxation Strategies for Viscous, Hypersonic Flows. In T. K. S. Murthy, editor, *Computational Methods in Hypersonic Aerodynamics*, pages 115–151. Kluwer Academic Publishers, 1991.
- [11] Peter A. Gnoffo, Roop N. Gupta, and Judy L. Shinn. Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium. NASA TP 2867, February 1989.
- [12] Francis A. Greene. An Upwind-Biased Space Marching Algorithm for Supersonic Viscous Flow. NASA TP 3068, March 1991.
- [13] R. N. Gupta, K. P. Lee, and E. V. Zoby. A Viscous Shock-Layer Analysis of 2-D and Axisymmetric Flows. AIAA Paper 93–2751, July 1993.
- [14] Roop N. Gupta, Jerrold M. Yos, Richard A. Thompson, and Kam-Pui Lee. A Review of Reaction Rates and Thermodynamic and Transport Properties for an 11-Species Air Model for Chemical and Thermal Nonequilibrium Calculations to 30 000 K. NASA RP 1232, August 1990.
- [15] Ami Harten. High Resolution Schemes for Hyperbolic Conservation Laws. *Journal of Computational Physics*, 49:357–393, 1983.
- [16] Sang-Wook Kang, Michael G. Dunn, and W. Linwood Jones. Theoretical and Measured Electron-Density Distributions for the RAM Vehicle at High Altitudes. AIAA Paper 72–689, June 1972.
- [17] W. L. Kleb and K. James Weilmuenster. Characteristics of the Shuttle Orbiter Leaside Flow During a Reentry Condition. AIAA Paper 92–2951, July 1992.
- [18] S. L. Lawrence, D. S. Chaussee, and J. C. Tannehill. Application of an Upwind Algorithm to the Three-Dimensional Parabolized Navier-Stokes Equations. AIAA Paper 87–1112, June 1987.
- [19] Scott L. Lawrence and A. Balakrishnan. UPS Code Development. Fifth National Aerospace Plane Technology Symposium Paper 13, October 1988.

- [20] Scott L. Lawrence, Upender Kaul, and John C. Tannehill. UPS Code Enhancements. Sixth National Aerospace Plane Technology Symposium Paper 16, April 1989.
- [21] Scott L. Lawrence, John C. Tannehill, and Denny S. Chausee. Upwind Algorithm for the Parabolized Navier-Stokes Equations. *AIAA Journal*, 27(9):1175–1183, September 1989.
- [22] K. K. Muramoto. The Prediction of Viscous Nonequilibrium Hypersonic Flows about Ablating Configurations using an Upwind Parabolized Navier-Stokes Code. AIAA Paper 93–2998, July 1993.
- [23] Chul Park. *Nonequilibrium Hypersonic Aerothermodynamics*. John Wiley & Sons, Inc., 1990.
- [24] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *Journal of Computational Physics*, 43:357–372, October 1981.
- [25] P. C. Stainback, C. B. Johnson, L. B. Boney, and K. C. Wicker. Comparison of Theoretical Predictions and Heat-Transfer Measurements for a Flight Experiment at Mach 20 (Reentry F). NASA TM X 2560, 1972.
- [26] J. Tannehill, P. Buelow, J. Ievalts, and S. Lawrence. A Three-Dimensional Upwind Parabolized Navier-Stokes Code for Real Gas Flows. AIAA Paper 89–1651, June 1989.
- [27] J. C. Tannehill, J. O. Ievalts, and S. L. Lawrence. An Upwind Parabolized Navier-Stokes Code for Real Gas Flows. AIAA Paper 88–0713, January 1988.
- [28] J. C. Tannehill and G. Wadawadigi. Development of a 3-D Upwind PNS Code for Chemically Reacting Hypersonic Flowfields. NASA CR 190182, 1992.
- [29] John C. Tannehill, John O. Ievalts, Philip E. Buelow, Dinesh K. Prabhu, and Scott L. Lawrence. Upwind Parabolized Navier-Stokes Code for Chemically Reacting Flows. *Journal of Thermophysics*, 4(2):149–156, April 1990.

- [30] Y. C. Vigneron, J. V. Rakich, and J. C. Tannehill. Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges. AIAA Paper 78-1137, 1978.
- [31] K. James Weilmuenster and Peter A. Gnoffo. Solution Strategies and Heat Transfer Calculations for Three-Dimensional Configurations at Hypersonic Speeds. AIAA Paper 92-2921, July 1992.
- [32] K. James Weilmuenster, Peter A. Gnoffo, and Francis A. Greene. Navier-Stokes Simulations of the Shuttle Orbiter Aerodynamic Characteristics with Emphasis on Pitch Trim and Body Flap. AIAA Paper 93-2814, July 1993.
- [33] William A. Wood and Richard A. Thompson. Combined LAURA-UPS Hypersonic Solution Procedure. NASA TM 107682, March 1993.
- [34] H. C. Yee. On Symmetric and Upwind TVD Schemes. NASA TM 88325, 1986.

Appendix A

UPS STARTING PLANE EXTRACTION PROGRAM

```
program laura01_to_fort2

c   uses UPS enthalpy curves

c   w.a.wood  9 - 16 - 93
c   modified  12 - 6 - 93

c   extracts UPS starting plane fort.2 from LAURA restart
c   file RESTART.in
c   both files are fortran binary

c   aux. file 'l01tof2.dat' contains:
c           description line
c           conversion factor to meters
c           cell number for starting plane
c           mach number
c   freestream velocity

c   grid averaged to the cell center plane

c   xl ( i, j, k, XYZ)   laura grid coordinates

c   xu ( k, l, XYZ)     UPS grid coordinates

c   vl ( i, j, k, variables)   laura cell centered variables
c           1-u, 2-v, 3-w, 4-T, 5-Tv,
c           densities: 6-n, 7-o, 8-n2, 9-o2, 10-no, 11-no+, 12-e-

c   vu ( k, l, variables)     UPS cell centered variables
c           1-rho, 2-rho*u, 3-rho*v, 4-rho*w, 5-Etotal,
c           concentrations: 6-o2, 7-o, 8-n, 9-no, 10-no+, 11-n2

parameter ( iplanes = 121, jplanes = 101, kplanes = 61)
```

```

common / d01 / hctb (50, 6), ttb (50), amws (6), hs0 (6)
common / d02 / spf ( 50, 6 )
common / d04 / hcint ( 6 ), tint

dimension xl( iplanes, jplanes, kplanes, 3 ),
-          xu( jplanes, kplanes, 3 ),
-          vl( iplanes, jplanes, kplanes, 12 ),
-          vu( jplanes, kplanes, 11 ),
-          avmw( jplanes, kplanes )

open ( 10, file = 'RESTART.in', form = 'unformatted',
-      status = 'old')
open ( 11, file = 'l01tof2.dat', form = 'formatted',
-      status = 'old')
read ( 11, * )

c      confac = .3048 ! convert from feet to meters
read ( 11, * ) confac

c      laura dimensions (cell centered) and number of species
read ( 10 ) ilm, jlm, klm, llms

      llm = llms + 5 ! # of laura variables

      kum = jlm + 1 ! ups grid dimensions
      lum = klm + 1

write ( 2 ) kum, lum

c      laura variables
read ( 10 ) ((( ( vl(il, jl, kl, ll), il=1, ilm), jl=1, jlm),
-             kl=1, klm), ll=1, llm),
c      laura grid
-             ((( ( xl(il, jl, kl, ll), il=1, ilm+1), jl=1, jlm+1),
-             kl=1, klm+1), ll=1, 3)

c      location of ups starting plane
read ( 11, * ) idata
igrid = idata + 1

c      grid averaging
do 201 ll = 1, 3

```

```

do 201 jl = 1, jlm + 1
do 201 kl = 1, klm + 1
201 xl( igrd, jl, kl, ll ) = .5 * ( xl( igrd, jl, kl, ll ) +
-                               xl( idata, jl, kl, ll ) )

c    obtain ups grid from laura grid
do 20 ku = 1, kum
jl = kum - ku + 1
do 20 lu = 1, lum
kl = lu
xu( ku, lu, 1) = -xl( igrd, jl, kl, 3) * confac
xu( ku, lu, 2) = xl( igrd, jl, kl, 1) * confac
20 xu( ku, lu, 3) = xl( igrd, jl, kl, 2) * confac

c    ups grid
write (2) ((( xu( ku, lu, mu), ku=1, kum), lu=1, lum), mu=1, 3)

c    total density from species densities
do 21 jl = 1, jlm
ku = jlm + 1 - jl
do 21 kl = 1, klm
lu = kl
vu( ku, lu, 1 ) = 0.
do 21 ll = 6, llm
21 vu( ku, lu, 1 ) = vu( ku, lu, 1 ) + vl( idata, jl, kl, ll )

c    momentums
read ( 11, * ) amach, velinf
vel2 = velinf * velinf
snd2 = vel2 / amach**2

do 22 jl = 1, jlm
ku = jlm + 1 - jl
do 22 kl = 1, klm
lu = kl
vu(ku,lu,2) = -amach * vl(idata,jl,kl,3) * vu(ku,lu,1)
vu(ku,lu,3) = amach * vl(idata,jl,kl,1) * vu(ku,lu,1)
22 vu(ku,lu,4) = amach * vl(idata,jl,kl,2) * vu(ku,lu,1)

c    species mass fractions
do 23 jl = 1, jlm
ku = jlm + 1 - jl
do 23 kl = 1, klm

```

```

    lu = kl
    vu( ku, lu, 6 ) = vl( idata, jl, kl, 9 ) / vu( ku, lu, 1 )
    vu( ku, lu, 7 ) = vl( idata, jl, kl, 7 ) / vu( ku, lu, 1 )
    vu( ku, lu, 8 ) = vl( idata, jl, kl, 6 ) / vu( ku, lu, 1 )
    vu( ku, lu, 9 ) = vl( idata, jl, kl, 10) / vu( ku, lu, 1 )
    vu( ku, lu, 10) = vl( idata, jl, kl, 11) / vu( ku, lu, 1 )
23 vu( ku, lu, 11) = vl( idata, jl, kl, 8 ) / vu( ku, lu, 1 )

c    energy
c    using UPS enthalpy table data

    rbar = 8314.34 ! universal gas constant

    call csplin ( 50, 6 ) ! determines enthalpy spline coeff.

    do 30 j1 = 1, jlm ! main energy loop
    ku = jlm + 1 - j1
    do 30 kl = 1, klm
    lu = kl

    tint = vl( idata, jl, kl, 4 ) ! cell temperature

c    compute dimensional species enthalpies
    call speval ( 50, 6 )

    hs1 = hcint( 1 ) * tint + hs0( 1 )
    hs2 = hcint( 2 ) * tint + hs0( 2 )
    hs3 = hcint( 3 ) * tint + hs0( 3 )
    hs4 = hcint( 4 ) * tint + hs0( 4 )
    hs5 = hcint( 5 ) * tint + hs0( 5 )
    hs6 = hcint( 6 ) * tint + hs0( 6 )

c    mixture enthalpy
    hhi = hs1 * vu(ku,lu,6) +hs2 * vu(ku,lu,7) +hs3 * vu(ku,lu,8) +
_      hs4 * vu(ku,lu,9) +hs5 * vu(ku,lu,10) +hs6 * vu(ku,lu,11)

c    kinetic energy
    u2 = .5 * vel2 * ( vl(idata,jl,kl,1)**2 + vl(idata,jl,kl,2)**2
_      + vl(idata,jl,kl,3)**2 )

    htot = u2 + hhi          ! total enthalpy

    por = 0.          ! pressure over density

```

```

do 34 is = 1, 6
34 por = por + vu(ku, lu, is+5) * rbar / amws(is) * tint

    etot = htot - por                ! total energy

    avmw( ku, lu ) = rbar * tint / por  ! average molecular weight

30 vu( ku, lu, 5 ) = etot * vu(ku, lu, 1) / snd2

c    ups variables
write (2) ((( vu( ku, lu, mu), ku=1, jlm), lu=1, klm), mu=1, 5)
write (2) ((( vu( ku, lu, mu), ku=1, jlm), lu=1, klm), mu=6, 10)
write (2) (( avmw( ku, lu ), ku = 1, jlm ), lu = 1, klm )

stop
end

subroutine csplin ( n, ne )           ! csplin from UPS

common / d01 / hctb (50, 6), ttb (50), amws (6), hs0 (6)
common /d02 / spf(50, 6)
common /d03 / asp(50, 6),bsp(50, 6),csp(50, 6),fsp(50, 6)

c=====
c=====>                               <=====
c=====>          evaluation of spline coefficients      <=====
c=====>                               <=====
c=====

do 10 m = 1,ne
do 10 i = 2,n-1
asp(i,m) = ttb(i)-ttb(i-1)
bsp(i,m) = 2.d0*(ttb(i+1)-ttb(i-1))
csp(i,m) = ttb(i+1)-ttb(i)
10 fsp(i,m) = 6.d0*((hctb(i+1,m)-hctb(i,m))/csp(i,m)-(hctb(i,m)-
+          hctb(i-1,m))/asp(i,m))

do 20 m = 1,ne
asp(1,m) = 0.d0
bsp(1,m) = 1.d0
csp(1,m) = -.5d0
fsp(1,m) = 0.d0
asp(n,m) = - .5d0

```

```

    bsp(n,m) = 1.d0
    csp(n,m) = 0.d0
20   fsp(n,m) = 0.d0

    call strid(1,n,ne)

    do 30 m = 1,ne
    do 30 k = 1,n
30   spf(k,m) = fsp(k,m)

    return
    end

    subroutine strid(nl,nu,ne) ! strid from UPS

    common /d03 / asp(50, 6),bsp(50, 6),csp(50, 6),fsp(50, 6)

c=====
c=====>                                     <=====
c=====>      non-periodic scalar tridiagonal solver   <=====
c=====>                                     <=====
c=====

    nlp = nl+nu
    do 10 m = 1,ne
    csp(nl,m) = csp(nl,m)/bsp(nl,m)
10   fsp(nl,m) = fsp(nl,m)/bsp(nl,m)
    do 20 j = nl+1,nu
    do 20 m = 1,ne
    z      = 1.d0/(bsp(j,m)-asp(j,m)*csp(j-1,m))
    csp(j,m) = csp(j,m)*z
20   fsp(j,m) = (fsp(j,m)-asp(j,m)*fsp(j-1,m))*z
    do 30 k = nl+1,nu
    do 30 m = 1,ne
30   fsp(nlp-k,m) = fsp(nlp-k,m)-csp(nlp-k,m)*fsp(nlp-k+1,m)

    return
    end

    subroutine speval(n,ne) ! speval from UPS

    common / d01 / hctb (50, 6), ttb (50), amws (6), hs0 (6)
    common /d02 / spf(50, 6)

```


1 / 32.d0, 16.d0, 14.008d0, 30.008d0, 30.008d0, 28.016d0 /

c..... species formation enthalpy (j/kg)

data (hs0(n),n=1,6)

1 / 0.d0 , 1.543119d+07, 3.362161d+07,
2 2.996123d+06, 3.283480d+07, 0.d0 /

c..... temperature table

data (ttb(n),n=1,50)

1 / 50.d0, 400.d0, 600.d0, 800.d0, 1000.d0, 1200.d0,
2 1400.d0, 1600.d0, 1800.d0, 2000.d0, 2200.d0, 2400.d0,
3 2600.d0, 2800.d0, 3000.d0, 3200.d0, 3400.d0, 3600.d0,
4 3800.d0, 4000.d0, 4200.d0, 4400.d0, 4600.d0, 4800.d0,
5 5000.d0, 5200.d0, 5400.d0, 5600.d0, 5800.d0, 6000.d0,
6 6200.d0, 6400.d0, 6600.d0, 6800.d0, 7000.d0, 7200.d0,
7 7400.d0, 8000.d0, 9000.d0, 10000.d0, 11000.d0, 12000.d0,
8 13000.d0, 14000.d0, 15000.d0, 16000.d0, 17000.d0, 18000.d0,
9 19000.d0, 20000.d0 /

c..... species enthalpy tables (calorically imperfect)

data (hctb(n, 1),n=1,50)

1 / 908.813d0, 914.579d0, 933.384d0,
+ 957.352d0, 980.354d0, 1000.647d0,
2 1018.239d0, 1033.615d0, 1047.298d0,
+ 1059.715d0, 1071.173d0, 1081.888d0,
3 1091.997d0, 1101.590d0, 1110.724d0,
+ 1119.433d0, 1127.740d0, 1135.662d0,
4 1143.21d0 , 1150.397d0, 1157.237d0,
+ 1163.742d0, 1169.928d0, 1175.812d0,
5 1181.414d0, 1186.751d0, 1191.844d0,
+ 1196.714d0, 1201.379d0, 1205.863d0,
6 1210.186d0, 1214.363d0, 1218.416d0,
+ 1222.361d0, 1226.214d0, 1229.992d0,
7 1233.708d0, 1269.796d0, 1291.02d0,
+ 1309.428d0, 1325.003d0, 1337.846d0,
8 1348.051d0, 1355.812d0, 1361.299d0,
+ 1364.696d0, 1366.212d0, 1366.029d0,
9 1364.370d0, 1361.391d0 /

data (hctb(n, 2),n=1,50)

```

1 /1313.994d0, 1394.834d0, 1372.763d0,
+ 1358.171d0, 1348.181d0, 1340.985d0,
2 1335.576d0, 1331.374d0, 1328.021d0,
+ 1325.303d0, 1323.08d0, 1321.263d0,
3 1319.805d0, 1318.674d0, 1317.848d0,
+ 1317.315d0, 1317.061d0, 1317.08d0,
4 1317.351d0, 1317.869d0, 1318.608d0,
+ 1319.554d0, 1320.691d0, 1321.995d0,
5 1323.448d0, 1325.031d0, 1326.73d0,
+ 1328.52d0, 1330.391d0, 1332.322d0,
6 1334.303d0, 1336.32d0, 1338.361d0,
+ 1340.415d0, 1342.472d0, 1344.526d0,
7 1346.566d0, 1352.558d0, 1361.882d0,
+ 1370.213d0, 1377.676d0, 1384.674d0,
8 1391.841d0, 1400.015d0, 1410.175d0,
+ 1423.369d0, 1440.629d0, 1462.882d0,
9 1490.871d0, 1525.064d0 /

```

```

data (hctb(n, 3),n=1,50)
1 /1482.86d0, 1482.86d0, 1482.86d0,
+ 1482.86d0, 1482.86d0, 1482.86d0,
2 1482.86d0, 1482.86d0, 1482.866d0,
+ 1482.881d0, 1482.926d0, 1483.031d0,
3 1483.24d0, 1483.616d0, 1484.224d0,
+ 1485.144d0, 1486.445d0, 1488.197d0,
4 1490.465d0, 1493.298d0, 1496.73d0,
+ 1500.793d0, 1505.494d0, 1510.833d0,
5 1516.803d0, 1523.381d0, 1530.536d0,
+ 1538.23d0, 1546.423d0, 1555.067d0,
6 1564.114d0, 1573.51d0, 1583.204d0,
+ 1593.143d0, 1603.276d0, 1613.55d0,
7 1623.922d0, 1655.172d0, 1705.539d0,
+ 1751.14d0, 1790.688d0, 1824.46d0,
8 1853.821d0, 1880.8d0, 1907.725d0,
+ 1936.922d0, 1970.476d0, 2010.027d0,
9 2056.631d0, 2110.681d0 /

```

```

data (hctb(n, 4),n=1,50)
1 / 969.129d0, 971.608d0, 983.714d0,
+ 1003.282d0, 1024.606d0, 1044.792d0,
2 1062.907d0, 1078.852d0, 1092.836d0,
+ 1105.081d0, 1115.888d0, 1125.473d0,
3 1134.025d0, 1141.703d0, 1148.639d0,

```

```

+ 1154.938d0, 1160.692d0, 1165.97d0 ,
4 1170.839d0, 1175.348d0, 1179.54d0 ,
+ 1183.455d0, 1187.122d0, 1190.572d0,
5 1193.827d0, 1196.91d0 , 1199.839d0,
+ 1202.632d0, 1205.306d0, 1207.874d0,
6 1210.35d0 , 1212.746d0, 1215.075d0,
+ 1217.349d0, 1219.576d0, 1221.77d0 ,
7 1223.936d0, 1249.367d0, 1265.608d0,
+ 1282.193d0, 1298.903d0, 1315.488d0,
8 1331.695d0, 1347.299d0, 1362.09d0 ,
+ 1375.907d0, 1388.578d0, 1399.988d0,
9 1410.02d0 , 1418.644d0 /

```

```

data (hctb(n, 5),n=1,50)
1 / 969.125d0, 969.831d0, 975.141d0,
+ 987.227d0, 1003.254d0, 1020.333d0,
2 1036.876d0, 1052.21d0 , 1066.143d0,
+ 1078.701d0, 1089.991d0, 1100.151d0,
3 1109.319d0, 1117.619d0, 1125.164d0,
+ 1132.049d0, 1138.357d0, 1144.16d0 ,
4 1149.52d0 , 1154.491d0, 1159.12d0 ,
+ 1163.45d0 , 1167.52d0 , 1171.364d0,
5 1175.019d0, 1178.512d0, 1181.876d0,
+ 1185.143d0, 1188.336d0, 1191.486d0,
6 1194.62d0 , 1197.761d0, 1200.94d0 ,
+ 1204.176d0, 1207.495d0, 1210.918d0,
7 1214.469d0, 1226.059d0, 1249.44d0 ,
+ 1278.948d0, 1311.095d0, 1366.826d0,
8 1415.495d0, 1467.173d0, 1519.227d0,
+ 1569.608d0, 1616.646d0, 1659.142d0,
9 1696.412d0, 1728.138d0 /

```

```

data (hctb(n, 6),n=1,50)
1 /1038.032d0, 1038.817d0, 1044.651d0,
+ 1057.787d0, 1075.089d0, 1093.45d0 ,
2 1111.179d0, 1127.577d0, 1142.45d0 ,
+ 1155.833d0, 1167.852d0, 1178.653d0,
3 1188.391d0, 1197.198d0, 1205.194d0,
+ 1212.482d0, 1219.155d0, 1225.285d0,
4 1230.94d0 , 1236.174d0, 1241.034d0,
+ 1245.564d0, 1249.797d0, 1253.767d0,
5 1257.501d0, 1261.023d0, 1264.357d0,
+ 1267.524d0, 1270.54d0 , 1273.426d0,

```

```
6 1276.197d0, 1278.87d0 , 1281.463d0,  
+ 1283.988d0, 1286.463d0, 1288.903d0,  
7 1291.323d0, 1298.611d0, 1311.745d0,  
+ 1329.699d0, 1357.905d0, 1391.942d0,  
8 1426.424d0, 1465.677d0, 1508.544d0,  
+ 1553.524d0, 1598.981d0, 1643.349d0,  
9 1685.285d0, 1723.743d0 /
```

```
end
```

Appendix B

UPS EXTERNAL GRID EXTRACTION PROGRAM

```
program laura_ups_grid

c   w.a.wood  9 - 10 - 93
c   modified  12 - 3 - 93

c   copies volume grid from LAURA RESTART.in file and
c   writes as fort.11 external grid for UPS

c   both files are fortran binary

c   conversion factor read from standard input

c   starting point read from standard input

c   xl ( i, j, k, XYZ)   laura grid coordinates

c   xu ( k, l, m, XYZ)   UPS grid coordinates

c   vl ( i, j, k, variables)   laura cell centered variables
c   1-u, 2-v, 3-w, 4-T, 5-Tv,
c   densities: 6-n, 7-o, 8-n2, 9-o2, 10-no, 11-no+, 12-e-

parameter ( iplanes = 121, jplanes = 101, kplanes = 61 )

dimension xl ( iplanes, jplanes, kplanes, 3 ),
_          xu ( jplanes, kplanes, iplanes, 3 ),
_          vl ( iplanes, jplanes, kplanes, 12 )

open ( 10, file = 'RESTART.in', form = 'unformatted',
_      status = 'old' )

c   confac = .3048   ! convert from feet to meters
c   confac = 1.
c   write(6,*) ' enter conversion factor to meters, default=',confac
```

```

read(5,*) confac

c   laura dimensions (cell centered) and number of species
read ( 10 ) ilm, jlm, klm, llms

llm = llms + 5 ! # of laura variables

kum = jlm + 1 ! ups grid dimensions
lum = klm + 1
mum = ilm + 1

write ( 11 ) kum, lum

c   laura variables
read ( 10 ) ((( ( vl ( il, j1, kl, ll), il=1, ilm), j1=1, jlm),
-             kl=1, klm), ll=1, llm),
c   laura grid
-             ((( ( xl(il,j1,kl,ll), il=1, ilm+1), j1=1, jlm+1),
-             kl=1, klm+1), ll=1, 3)

istart = 14 ! first plane in external grid
write(6,*) ' enter starting plane, default=', istart
read(5,*) istart

c   obtain ups grid from laura grid
do 20 mu = 1, mum
il = mu
do 20 ku = 1, kum
jl = kum - ku + 1
do 20 lu = 1, lum
kl = lu
xu ( ku, lu, mu, 1 ) = -xl ( il, jl, kl, 3 ) * confac
xu ( ku, lu, mu, 2 ) = xl ( il, jl, kl, 1 ) * confac
20 xu ( ku, lu, mu, 3 ) = xl ( il, jl, kl, 2 ) * confac

c   ups grid
do 21 mu = istart, mum
21 write ( 11 ) ((( xu ( ku, lu, mu, n ), ku = 1, kum ),
-                 lu = 1, lum ), n = 1, 3 )

stop
end

```

Appendix C

EXTERNAL GRID ADAPTOR ROUTINE FOR UPS SOLUTIONS

```
program grdc2

c   w.a.wood  3 - 10 - 94

c   reads in ups external grid, modifies a plane by stretching,
c   as well as all following planes,
c   and writes out a new external ups grid

c   keeps the inner 40% of points fixed
c   linearly varies stretching on outer 60% of grid to reach
c   the specified increase in size

parameter ( im = 52 )      ! number of planes

dimension x(im, 16, 102), y(im, 16, 102), z(im, 16, 102)

read (11) jm, km
write(21) jm, km

do 10 i=1, im
    read(11) (( x(i, j, k), j = 1, jm), k = 1, km),
-           (( y(i, j, k), j = 1, jm), k = 1, km),
-           (( z(i, j, k), j = 1, jm), k = 1, km)
10 continue

write(*,*) ' enter starting plane number to be modified'
read(*,*) ichg1

c   write(*,*) ' enter ending plane number to be modified'
c   read(*,*) ichg2

write (*,*) ' Enter percentage length increase (0 = same) '
```

```
c      do 20 ichg = ichg1, ichg2
c      write(*,*) ' enter proportional length increase for plane ',
c      -         ichg, ' (0=same)'
```

read(*,*) s1
s1 = s1 * .01
kp1 = km * 2 / 5 ! inner points
kp2 = km - kp1 ! outer points

```
do 12 k = kp1, km
  fac = 1. + s1 * float( k - kp1 ) / float( kp2 )
  do 11 j = 1, jm
    do 20 ichg = ichg1, im
      y(ichg, j, k) = y(ichg, j, k) * fac
      z(ichg, j, k) = z(ichg, j, k) * fac
20    continue
11    continue
12  continue
```

```
do 13 i=1, im
  write(21) ((x(i,j,k), j=1,jm), k=1,km),
-          ((y(i,j,k), j=1,jm), k=1,km),
-          ((z(i,j,k), j=1,jm), k=1,km)
13  continue
```

```
stop
end
```