A00-16537

AIAA 2000-0678
Internet Based Computational Fluid Dynamics Calculation of Hypersonic Heating

Theodore K. Lee and Xiaolin Zhong
University of California, Los Angeles
Los Angeles, CA

Leslie Gong
NASA Dryden Flight Research Center
Edwards, CA

38th Aerospace Sciences Meeting & Exhibit
January 10–13, 2000 / Reno, NV

For permission to copy or republish, contact the American Institute of Aeronautics and Astronautics
1801 Alexander Bell Drive, Suite 500, Reston, Virginia 20191-4344

Internet Based Computational Fluid Dynamics Calculation of Hypersonic Heating

Theodore K. Lee[*] Xiaolin Zhong[†] and Leslie Gong[‡]
University of California, Los Angeles, California 90095

Abstract

In the last twenty years, the development of computational fluid dynamics (CFD) has advanced tremendously, coinciding with the development of more powerful computers, but CFD numerical codes are still not very easy to use. Current numerical codes still require an advanced knowledge of CFD methods and fluid flow modeling requirements before a successful simulation can be calculated. Visualization of CFD results usually requires the use of a separate graphical post processor. The objective of this work was to develop a new numerical code Internet system that will dramatically improve the way that numerical codes, such as those use in computational fluid dynamics, are currently developed and tested. A package that integrates a high order accurate CFD inviscid flow solver with approximate analytic aerodynamic heating methods was developed and utilized in conjunction with the Internet system which includes advanced graphical features to view the computational results. The Internet system was developed with the Java language, while the CFD inviscid flow solver and heating methods were developed with the Fortran language.

Introduction

The Internet has dramatically changed the speed and accuracy with which information is exchanged and used. While most of the focus on the Internet's capabilities has been in the areas of news media [1] and Web commerce [2], scientific communities are just beginning to see the Internet as a new resource for their research. A large area of scientific computational research has been in the field of computational fluid dynamics (CFD) [3]. CFD is used in aerospace and mechanical engineering, as well as other fields, to simulate fluid flows in situations that are difficult to reproduce experimentally.

The objective of this study was to develop an Internet based computational fluid dynamics (CFD) system using the Java programming language [4], whereby a remote user may access and execute CFD programs through the World Wide Web. Upon completion of the numerical code's execution, the output could be immediately viewed using advanced graphical features over the Internet. This study is motivated by the need for an easier method with which CFD results and methods can be communicated to interested parties, as well as the need for efficient and accurate aerodynamic heating predictions. By using the Internet and Java applets, results can be easily and quickly accessed by outside parties. This Internet system allows a user to remotely run flow test cases and view results for a variety of flow conditions. A database of existing results is also available for the user to download and view.

There are numerous Internet Web sites that utilize user inputs to produce a specific result. Sites such as search engines that ask for key words, business sites that ask for credit card info, and even online message boards all require and react to user inputs. There are even educational engineering programs which have been completely written using the Java language [5]. We have developed our own Web site that allows remote users to remotely execute our CFD and aerodynamic heating codes. The remote users will use the Web site to define flow conditions, choose numerical methods, and input other needed initial parameters. Once these inputs are entered by the remote user, our Web site will automatically execute our CFD programs according to these specifications. When the programs are completed, and the results are output, the users may then download the numerical data file to their home computer for their own use. The results can also be viewed across the Internet, using a built in graphical post processor. Users are able to view complete grids geometries, flow contours, and other types of information in a graphical manner. Not only does our Web site allow the user to execute the CFD programs, but a database of existing results are also available to be downloaded.

The Web site is developed using mainly Java applets. Java applets are portable programs which most Web browsers can execute. These Java applets are responsible for taking the user inputs, executing the programs, and providing results. The Java applets are written using the Java 2 language, and contain advanced graphical capabilities using the Java 3D libraries. They receive inputs in a user friendly graphical manner, and automatically begin execution of the CFD programs. The user is able to track the progress of the CFD programs,

[*] Graduate Student, theodore@seas.ucla.edu, Student Member AIAA

[†] Associate Professor, Mechanical and Aerospace Engineering Department, xiaolin@seas.ucla.edu, Senior Member AIAA

[‡] NASA Dryden Flight Research Center

and upon completion, the user is able to view the results in a graphical manner.

The complete Internet software package is used to provide accurate predictions of surface heating on hypersonic vehicles. The advantages of the Internet system can be described as follows :

- The new Internet system allows us to be able to remotely execute our current CFD and aerodynamic heating code from any computer that is connected to the Internet by using only a Web browser such as Netscape.
- This system does not require any advance knowledge of Internet processes beyond interacting with a Java applet on a Web page.
- This system also allows engineers to download CFD and aerodynamic heating numerical data that has been produced by computations that were previously executed.
- Graphical output is built into the Java applet, allowing users to immediately view the results of a computation in a graphical manner.

The work on integrating a high order accurate CFD inviscid flow solver with analytic heating models is motivated by the need for accurate prediction of aerodynamic heating around hypersonic vehicles by NASA Dryden [6–9]. This work was done in conjunction with the development of the Internet system. Currently, the state-of-the-art CFD methods efficiently solve for the inviscid solutions. However, viscous flows, both laminar and turbulent, are still too expensive for engineering applications. In computing hypersonic flow fields, an accurate inviscid solution can be obtained by any number of conventional high order shock capturing schemes if the main interests are the mean aerodynamic properties such as pressure and temperature. The shape of the hypersonic vehicle may vary from a simple cone to a complex three-dimensional aircraft. There may be complex shock interactions present in the flow field. It is because of this constraint, that the shock capturing schemes are chosen as the numerical methods that are used. The shock capturing schemes do not require a knowledge of the shock structure beforehand, and they have been proven to be very robust. For hypersonic conditions, the number of choices of shock capturing methods is very large. Therefore, it was decided that only two different shock capturing methods would be developed. The shock capturing methods that were developed are the Weighted Essentially Non-Oscillatory [10] (WENO) scheme, and the Total Variation Diminishing [11] (TVD) scheme. Three-dimensional grid geometries have also been developed. The grid generation routines that have already been developed include the spherical body, the elliptical cone, and a swept wing. By integrating the heating methods, along with the inviscid flow results provided by a high order accurate numerical method, highly accurate heating predictions for various hypersonic vehicles will be provided.

CFD Numerical Methods

In computing hypersonic flow fields, an accurate inviscid solution can be obtained by any number of conventional high order shock capturing schemes if the main interests are the mean aerodynamic properties such as pressure and temperature. The accurate calculation of surface heating rates on hypersonic vehicles is dependent on an accurate inviscid CFD numerical solution of the mean aerodynamic flow properties. The shape of the hypersonic vehicle may vary from a simple cone to a complex three-dimensional aircraft. There may, therefore, be complex shock interactions present in the flow field. An example of a hypersonic vehicle is the Hyper-X. A surface grid of the Hyper-X is shown in Figure 1. It is because of this constraint, that the shock capturing schemes are chosen as the numerical methods that are used. The shock capturing schemes do not require a knowledge of the shock structure beforehand, and they have been proven to be very robust. Shock capturing numerical schemes have been used extensively in aerodynamic calculations for flows involving shocks [12] [13] [14] [15] [16].

Various shock capturing numerical schemes include the Essentially Non-Oscillatory (ENO) scheme [13,17] which has been used extensively with flows involving shocks [14,15,18]. The Weighted Essentially Non-Oscillatory (WENO) scheme [10] is an extension of the ENO scheme. The Total Variation Diminishing (TVD) scheme was originally developed by Harten [11] and it also has been used in flows involving shocks [16,19,20].

For hypersonic conditions, the number of choices of shock capturing methods is very large. Therefore, it was decided that only two different shock capturing methods would be developed. The shock capturing methods that were developed are the Weighted Essentially Non-Oscillatory (WENO) scheme, and the Total Variation Diminishing (TVD) scheme.

Governing Equations

The governing equations for simulating fluid flows consist of the Navier-Stokes equations. For an inviscid flow, however, the Navier-Stokes equations simplify into the Euler Equations. The three-dimensional Euler equations are as follows :

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \tag{1}$$

where,

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \tag{2}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uw \\ uE \end{bmatrix}, \tag{3}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ \rho vw \\ vE \end{bmatrix}, \tag{4}$$

$$H = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + P \\ wE \end{bmatrix}. \tag{5}$$

P is the pressure, ρ is the density, u, v and w are the horizontal, vertical and transverse velocities, and E is the energy. The equation of state for a perfect gas is :

$$P = \rho RT \tag{6}$$

where R is the universal gas constant and T is the temperature. For three-dimensional flow cases, Eq (1) is transformed to a body fitted grid.

Weighted Essentially Non-Oscillatory Schemes (WENO)

The WENO schemes are an extension of the well known ENO [13,17] schemes. The main difference between the two methods, is that where the ENO scheme chooses the smoothest stencil in which to approximate the flux, the WENO scheme uses all the possible stencils. The WENO scheme that is used here is the finite-difference method that was developed by Jiang and Shu [10].

To describe the WENO schemes briefly, for a hyperbolic system such as :

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{7}$$

one can solve for $\partial F/\partial x$ using :

$$F_x = \frac{1}{\Delta x}(f_{i+1/2} - f_{i-1/2}). \tag{8}$$

For the fifth-order WENO scheme, the approximation of $f_{i+1/2}$ is based on all three possible stencils. These three stencils are :

$$(i-2, i-1, i),$$

$$(i-1, i, i+1),$$

$$(i, i+1, i+2).$$

The WENO scheme, therefore, uses an approximation to $f_{i+1/2}$ that encompasses all three stencils in a manner such as :

$$f_{i+1/2} = Af_1 + Bf_2 + Cf_3 \tag{9}$$

where f_1 is the approximation to $f_{i+1/2}$ based on the first stencil, f_2 the approximation based on the second stencil, and f_3 the approximation based on the third stencil. The coefficients A, B and C are found by determining the smoothness of each stencil. The smoothest stencil, therefore, would have the largest coefficient value. This ensures that any approximations derived from discontinuities would result in a very small coefficient value.

Total Variation Diminishing (TVD)

The use of TVD schemes in the solution of hyperbolic conservation laws is very popular. It has been a popular method ever since the original concept was introduced by Harten [11]. While many variations on the TVD method have been developed, the basis of all of them remains the same. The TVD method is based on the idea of choosing the smallest flux value if all the other approximations to the flux are of the same sign, and to choose a flux value of zero, if the approximations of the flux are of different sign. This ensures that there will be no new local extrema created, and therefore the TVD schemes are monotonicity preserving.

Because there have been so many variations on the TVD method, a simple one that is used by Furumoto et. al. [16] was chosen which involves the minmod limiter. The approximation to the flux $f_{i+1/2}$ is as follows :

$$f_{i+1/2} = F_{i+1} - \frac{1}{2} minmod[\Delta_{i+1}, \Delta_i] \tag{10}$$

where,

$$\Delta_i = F_{i+1} - F_i. \tag{11}$$

The minmod function is responsible for choosing the smallest stencil if both values of Δ are of the same sign, or zero if the two values are of opposite signs. It can be formulated as :

$$minmod(a, b) = \frac{1}{2}[sgn(a) + sgn(b)]min(|a|, |b|). \tag{12}$$

In considering computational efficiency, the TVD scheme involves much less computational time than the WENO scheme. This translates into a faster, more efficient program. When considering accuracy though, the TVD scheme is second order accurate while the WENO scheme is fifth order accurate. Both schemes, however, reduce to first order accuracy at discontinuities.

Time Stepping

The method that was used for time integration is a standard Runge-Kutta second order scheme. It can be formulated as follows :

$$U^{(0)} = U^{(n)} \tag{13}$$

$$U^{(1)} = U^{(0)} + \Delta t R_1 \tag{14}$$

$$U^{(n+1)} = \frac{1}{2}(U^{(1)} + U^{(0)} + \Delta t R_2) \tag{15}$$

where the R's are the residuals of the spatial integration, and Δt is the time step.

Boundary Conditions

Boundary conditions at exit planes is dealt with using extrapolation of the conservative variable. Usually a second order extrapolation is performed as follows:

$$U_i = 2U_{i-1} - U_{i-2}. \tag{16}$$

Wall boundary conditions are enforced by requiring that there be no velocity normal to the wall. Details of the wall boundary conditions are described by Chakravarthy [21]. The wall boundary conditions are based on determining the conservative variables when the velocity normal to the body is defined as zero. Initial flow conditions consist of dividing the computational domain into two sections. For the outer section, freestream values are set at each grid point. For the inner section around the body, normal shock values are set for each grid point.

Aerodynamic Heating Analysis

The aerodynamic heating analytical methods that are used in the current work originated from NASA Dryden's TPATH aeroheating code [7,8]. These heating methods, however, require an inviscid flow solution to be input. Previously, the inviscid flow would be solved analytically. By utilizing a inviscid flow that has been computed using a high order shock capturing method, the heating results are much more accurate. The heating methods are able to compute both laminar, turbulent and transitional flows, and flows of constant and variable entropy. The methods are also able to compute steady and transient flows.

Stagnation Point Heating

For stagnation point calculations, the heating program will automatically determine the heating at only the stagnation point, as defined by the user. The stagnation point will not always be located at the nose of the vehicle. Depending on the angle of attack, and the orientation of the vehicle, the user must specify the stagnation point location.

The heat transfer coefficient for a three-dimensional stagnation point can be found using Fay and Riddell's [22] method,

$$h = 0.94F(\frac{\rho_{st}\mu_{st}}{\rho_w\mu_w})^{0.4}(\frac{dU}{dx}\rho_w\mu_w G)^{0.5}, \tag{17}$$

where h is the heat transfer coefficient (lbm/ft^2s), ρ is the density $(slugs/ft^3)$, μ is the viscosity (lbm/fts), $\frac{dU}{dx}$ is the velocity gradient $(1/s)$, G is the gravitational constant $(= 32.17 lbmft/lbs)$, and F is a user defined adjustment factor with default values of 1.73 for laminar cases, and 1.15 for turbulent cases. The velocity gradient is determined using :

$$\frac{dU}{dx} = \frac{1}{R}(\frac{2(P_{st} - P_l)}{\rho_{st}})^{0.5} \tag{18}$$

where P is the pressure (lb/ft^2), and R is the nose radius (ft). The subscript st refers to stagnation values, w refers to wall values, and l refers to local values behind the shock.

The heat transfer coefficient for a two-dimensional stagnation point can be found using Beckwith and Gallagher's [23] method,

$$h = 0.704F(\frac{\rho_{st}\mu_{st}}{\rho_w\mu_w})^{0.44}(\frac{dU}{dx}\rho_w\mu_w G)^{0.5}. \tag{19}$$

Non-Stagnation Point Heating

Non-stagnation point heating corresponds to the surface heating along the body away from the stagnation point. In this case, the user must define whether the flow is laminar, turbulent or transitional, and whether a constant or variable entropy assumption is to be used. A constant entropy assumption is based on the idea that the properties at the edge of the boundary layer are determined by the surface pressure and the entropy behind a normal shock wave. This assumption is normally valid for high Reynolds numbers and in the nose region. The variable entropy assumption is based on the idea that as the boundary layer grows along the length of the body, more of the inviscid flow is entrained into the boundary layer. The streamlines which passed through the bow shock are swallowed by the boundary layer, and this can have a large effect on heating rates.

Constant Entropy Assumption

If a constant entropy condition is chosen, the boundary layer displacement thickness is set to equal zero. A constant entropy assumption is based on the idea that the properties at the edge of the boundary layer are determined by the surface pressure and the entropy behind a normal shock wave. The boundary layer edge values are assumed to be the wall values. There is a further distinction between laminar and turbulent calculations. There is one laminar method and two turbulent methods.

For the laminar method, the method is based on the Blasius incompressible skin friction formula which is related to heat transfer with a modified Reynolds analogy. The Stanton number relates the following

$$ST = R_A \frac{C_f}{2} = \frac{h}{\rho V}, \qquad (20)$$

where ST is the Stanton number, R_A is the Reynolds Analogy factor, C_f is the skin friction coefficient, and V is the total velocity. The Stanton number is a dimensionless heat transfer coefficient, the Reynolds analogy is a connection between momentum and heat transfer, and the skin friction coefficient is the ratio between the wall shear stress and momentum transfer. The Blasius skin friction formula [24] is

$$\frac{C_f}{2} = 0.332(Re_l)^{-0.5}, \qquad (21)$$

where Re_l is the local Reynolds number. Compressibility effects are accounted for using Eckert's [25,26] reference enthalpy method. The reference enthalpy method is used due to the fact that in high speed boundary layers, the freestream temperature must be replaced by an adiabatic wall temperature in order to compute accurate heating rates. Based on the above equations, the heat transfer coefficient can then be derived as,

$$h = 0.332 \frac{F \rho_{sr} V_l G}{Re_{sr}^{0.5} Pr_{sr}^{2/3}}, \qquad (22)$$

where Pr is the Prandtl number, the subscripts sr refer to reference enthalpy values and l the wall value. $Pr_{sr}^{2/3}$ is the modified Reynolds analogy factor (R_A).

The first turbulent method is based on the skin friction theory of Van Driest [27]. The skin friction coefficient, C_f, is calculated using the equations found in Van Driest [27]. It involves an iterative routine to solve for C_f. Once the skin friction coefficient is calculated, the heat transfer coefficient can then be calculated using the same Stanton number relation as above for the laminar case. The heat transfer coefficient is calculated using,

$$h = \frac{F C_f \rho_l V_l G}{2 Pr_l^{2/3}}. \qquad (23)$$

The second turbulent method is based on the following incompressible skin friction equation provided by NASA Dryden,

$$\frac{C_f}{2} = \frac{0.185}{(Log(Re_l))^{2.584}}. \qquad (24)$$

The equation is modified by using the Reynolds number based on Eckert's reference enthalpy, resulting in an equation,

$$\frac{C_f}{2} = \frac{0.185}{(Log(Re_{sr}))^{2.584}}. \qquad (25)$$

The heat transfer coefficient is calculated as,

$$h = \frac{0.185 F \rho_{sr} V_l G}{(Log(Re_{sr}))^{2.584}(Pr_{sr})^{2/3}}. \qquad (26)$$

Variable Entropy Assumption

If a variable entropy condition is chosen, the boundary layer displacement thickness is not equal to zero. The variably entropy assumption is based on the idea that as the boundary layer grows along the length of the body, more of the inviscid flow is entrained into the boundary layer. Given the boundary layer displacement thickness, the inviscid values at the boundary layer edge are determined through an interpolation method from the inviscid flow solution provided by the CFD method. These include temperature (T_l), pressure (P_l), density (ρ_l), speed of sound (A_l), enthalpy (E_l), velocities (u_l, v_l, w_l), and specific heat ratio (γ_l). The subscript l now refers to values that are at the outer edge of the displacement boundary layer.

The heat transfer coefficient for the laminar variable entropy case is calculated as follows. A Reynolds number based on the boundary layer momentum thickness is calculated as,

$$Re_\Theta = \frac{\Theta_m \rho_l V_l G}{\mu_l}, \qquad (27)$$

where Θ_m is the boundary layer momentum thickness. The heat transfer coefficient is then computed as,

$$h = 0.22(\frac{\mu_{sr}}{\mu_l})\frac{\rho_l V_l G}{Re_\Theta}(Pr_w)^{-0.6}. \qquad (28)$$

This equation is derived by relating the Blasius incompressible skin friction coefficient [24] to heat transfer by a Reynolds analogy factor and using Eckert's reference enthalpy method.

The heat transfer coefficient for the turbulent variable entropy case is calculated as follows. The skin friction equation for turbulent flow is defined as :

$$C_f = 2C_1(R\Theta_l)^{-m} \qquad (29)$$

and a Reynolds analogy factor of :

$$R_A = (Pr_w)^{-0.4}. \qquad (30)$$

Using Eckert's reference enthalpy, the heat transfer coefficient can be calculated using :

$$h = C_1(\rho_l V_l G)(Re_\Theta)^{-m}(\frac{\mu_{sr}}{\mu_l})^m(\frac{\rho_{sr}}{\rho_l})^{1-m}(Pr_w)^{-0.4}, \qquad (31)$$

where C_1 and m are defined later. The Reynolds number based on momentum thickness is calculated the same as for the laminar case. The boundary layer momentum thickness computation procedure is taken from the method described by Zoby [28].

Computational Results

The validation of the CFD inviscid flow and aeroheating analysis was performed for two different test cases. The first test case was the X-15 swept wing. In this validation test case, the heat transfer coefficients were computed using the integrated software program and compared to experimentally measured values. The second test case was an elliptic lifting body. Both stagnation point heating and heating along the surface of the body were compared to measured experimental values.

X-15 Swept Wing Heating Test Case

The X-15 was a rocket powered plane that was used by NASA to test hypersonic flight conditions back in the 1960's. The X-15 is a very good geometry to further test the CFD methods. It is a geometry with a high amount of three-dimensional effects. There are experimental values of pressure and heat transfer coefficients [7] along the lower surface of the wing, with which a comparison can be made. A schematic of the whole X-15 airplane is shown in Figure 2. This schematic shows the locations that pressure and heat transfer measurements were taken on the lower surface of the wing. The X-15 wing is based on a NACA *66005* airfoil [29]. It has a sweep angle of 36.75° at its leading edge, and a sweep angle of 17.74° at its trailing edge. A figure of the grid of the wing is shown in Figure 3. The geometry of the three-dimensional grid around the wing section is quite different from the grid geometry around the sphere or elliptical cone. The grid is stretched to allow for more points near the leading edge. The CFD program is capable of computing flows over any arbitrary wing as long as the coordinates of the airfoil geometry are given.

The experimentally found pressure values along the lower surface of the X-15 wing is documented by Quinn and Palitz [7] for a variety of different flights at varying Mach numbers and angles of attack. Mach numbers varied from 4.15 to 5.5, and angles of attack varied from 0.9° to 16.3°. There were three separate test cases that were computed for the X-15 using the ideal gas assumption.

A low angle of attack case that was computed consisted of a Mach number of 5.1 with an angle of attack of 2°. The pressure contours of this flow field that were computed with the CFD method are shown in Figure 4. The pressure along the lower surface is compared to experimental values, and is shown in Figure 5. The results show that both the TVD and WENO methods agree very well with the experimental results. The numerical grid size used was 41 × 41 × 11, which is a relatively coarse grid. The grid size, however, seems to be adequate for this computation.

A medium angle of attack case was computed, which consisted of a Mach number of 4.15 and an angle of attack of 10.3°. The pressure along the lower surface is compared to experimental values, and is shown in Figure 6. Again, both the TVD and WENO numerical methods agree very well with the experimental results.

A high angle of attack case was computed, which consisted of a Mach number of 4.98 and an angle of attack of 16.3°. The pressure along the lower surface is compared to experimental values, and is shown in Figure 7. Again, both numerical methods agree very well with the experimental results.

Three different heat transfer calculations have been used for each flight case. All the calculations were computed assuming a turbulent flow. There are two constant entropy calculations based on the analytical models of Van Driest [27] (Eqn. 23) and Eckert [25](Eqn. 26). There is also a variable entropy analytical model(Eqn. 31). The heat transfer coefficient comparisons are shown in Figures 8 to 13.

As can be seen in the figures, the heat transfer coefficients that were calculated using the aerodynamic heating program are very close to the experimentally found heat transfer coefficients. This is true regardless of the Mach number and the angle of attack. The agreement between the computational results and the experimental results, especially for the high angle of attack cases, show that the three-dimensional wing geometry is very good at predicting heating rates.

Elliptical Lifting Body Heating Test Case

The second validation test case that was used was an elliptical lifting body. The grid generation of the lifting body was designed more specifically to the body presented by Hillsamer and Rhudy [30]. The lifting body is similar to the elliptical cone. A picture of the experimental model is shown in Figure 14 with the model schematics shown in Figure 15. Hillsamer and Rhudy [30] present experimental values for heating that are used for comparison. A figure of the numerical grid for this geometry is shown in Figure 16.

Validation tests have been conducted to compare the stagnation point values with the values measured by Hillsamer and Rhudy for three different Reynolds numbers. All of the different Reynolds numbers were computed using Mach 10 ideal gas flow with laminar flow conditions. The three different Reynolds numbers were 0.31×10^6 per foot, 1.0×10^6 per foot, and 2.38×10^6 per foot. The geometry of the elliptical lifting body was based on two different radius' of curvature. Using the aerodynamic heating program, the stagnation heating method of Fay and Riddell [22] (Eqn. 17) was used to compute stagnation heating rates. Because the Fay and Riddell method requires a nose radius, the stagnation heating was computed using both radius' of curvature.

For the lower Reynolds number, the experimentally computed stagnation heating value was approximately 0.0042 Btu/ft^2sR. The numerically found values based on the two different radius' was 0.004155 and 0.004965. For the middle Reynolds number, the experimentally computed value was approximately 0.0082 Btu/ft^2sR. The numerical values were 0.007462 and 0.008917. For the higher Reynolds number, the experimentally computed value was approximately 0.014 Btu/ft^2sR. The numerical values were 0.01151 and 0.01376. The experimental values assumed an isothermal wall temperature of 470° R, therefore the aerodynamic heating code also assumed an isothermal wall temperature.

Hillsamer and Rhudy also presented experimental heating values along the surface of the lifting body. For the low Reynolds number case, the computationally predicted heating is compared with their experimental values in Figures 17 and 18. For the numerical computation, an isothermal wall of 470° R was assumed, as well as laminar, constant entropy flow (Eqn. 22). Figure 17 shows the heating distribution beginning at the stagnation point proceeding along the body on the side with the smaller radius of curvature. Figure 18 shows the heating distribution beginning at the stagnation point proceeding along the body on the side with the larger radius of curvature. There are two computational results shown, the only difference being the grid resolution. The regular grid had a resolution of 41 × 21 × 21 while the fine grid had a resolution of 41 × 21 × 41. The results show a close agreement between the experimental values and the computational results.

Internet System

We have developed an Internet system that is used in conjunction with the CFD and aerodynamic heating codes. The purpose of the Internet system is to allow us to remotely execute our codes through the Internet with a graphical user friendly interface. The Internet system also allows us to plot the output of our numerical codes with a built in graphical post processor. A flowchart of the basic design of the Internet system is shown in Figure 19.

The Internet system was programmed using the Java language [4]. The Java language is one of the most popular computer languages when dealing with many Internet applications. It has many advantages over traditional computing languages such as C++ and Fortran. Java's biggest advantage may be its portability. A program written once in Java, can subsequently be executed on a variety of operating systems, including Windows and UNIX based systems. Another advantage of the Java language is that there are numerous libraries that the programmer can use. These libraries allow for a quicker and more efficient programming strategy, especially when dealing with multimedia applications involving sound, graphics and window interfaces.

Recently, a new addition to the Java language called Java 3D was released for use with applications requiring advanced graphical capabilities. This new feature of the Java language implements advanced graphical libraries which allows us to incorporate three-dimensional scenes in the application. For our CFD and aerodynamic heating program, such uses include viewing the vehicle body and numerical grid, viewing the inviscid flow contours, as well as other plots of desired properties along a defined geometry. The implementation of the Java 3D graphical features with the existing CFD codes, means that we can efficiently develop a fully integrated software package that involves advanced graphical features.

The Internet system was designed using a client-server communication structure. The idea behind a client-server system is that the remote users Web browser will access a Java applet which is classified as a client. This Java applet is embedded on a Web page. This Java applet allows the remote users to input information pertaining to the specific conditions that they want the CFD and aerodynamic heating computations to calculate. The Java applet, then, has the ability to communicate with another Java program that is running on the server's computer. This is classified as the Java server. The Java server receives the information that the remote users have input, and performs the required tasks such as writing data files, reading data files, or beginning executions of the numerical code. The reason that a client-server system must be implemented is that due to security restrictions, a client is not able to directly execute a program that is situated on a server machine. The only way to remotely execute a program is for the client to instruct the Java server to execute a program on the server machine.

The core of this system is the client and server programs which are both written using the Java language. A Java applet is used by the remote users who wish to access the code. It has been designed to respond to the remote user's request to execute our numerical codes. The Java applet will query the remote users to provide the information that is specified. The remote user can choose to view existing solutions, or define parameters for a new computation. Initial flow conditions, grid sizes and a choice of numerical methods are some of the questions that the remote user will have to answer. A chart of the remote user's interaction with the Java applet is shown in Figure 20. Once the remote user has finished inputing all of the needed parameters, the client sends this information to the server. The server executes the needed commands. These commands may be for reading an existing data file, writing the necessary input data files, or beginning execution of the numerical code. Output data file names are designated based on the current time so that they will be unique. A chart of the server operations is shown in Figure 21. The data files containing the numerical results are added to a database containing previously computed results. The server automatically updates the Web page that displays the results, and add any newly computed solution. This database is accessible from a Web page that is designed specifically to display the data files that are available to be downloaded.

The design of the Java applet is to allow the remote user the ability to perform a number of different tasks. The first set of tasks that a user may choose is to generate and view a new grid, or to view a grid that already exists within the database. The user can define the size of a new grid and immediately view it on the Web browser. A screen shot of how the grid is displayed using a Web browser is shown in Figure 22. The displayed grid is a three-dimensional object that can be rotated along all three axes. The user can also zoom in closer to look at specific locations of the grid. The zoom function is done using the buttons on the lower left hand side of the graphical screen. The rotation of the object can be done using both the buttons, or by simply moving the mouse while holding down the right mouse button.

The second set of tasks is to allow the remote user to begin a new CFD inviscid flow computation, or to view existing inviscid flow results. To begin a new inviscid flow computation, the user must define the freestream flow properties, the numerical methods, generate a grid, and define the boundary condition order of accuracy. Once completed, the server automatically begins the execution of the CFD numerical code. The results of the inviscid computation can be displayed using two different formats. The first format is for line plots of various aerodynamic properties along a given grid line. A screen shot of a line plot for temperature is shown in Figure 23. The remote users can also interpolate values for the x and y axes by clicking on the screen the location that they are interested in. These values are displayed in the corresponding boxes on the lower right hand side of the applet. The aerodynamic properties that can be displayed include pressure, temperature, density, Mach number, speed of sound and the velocity. The second format for displaying aerodynamic properties is with color contouring. A screen shot of a temperature contour over a three-dimensional sphere is shown in Figure 24. This method of displaying the aerodynamic properties shows the overall distribution much better than line plots. A color table showing the temperature distribution in relation to numerical values of temperature is also displayed on the left hand side of the graphical window.

The third and last set of tasks involve remote users using the inviscid flow to compute surface heating over the respective geometries. The users can choose to compute a new heating solution, or view an existing heating solution. Users need to input the heating parameters that are needed, such as choosing between laminar and turbulent models, and the region of the vehicle that is to be considered. The server automatically executes the heating program, and produces a data file containing the surface heating results. These surface heating results can be viewed in the same manner as previously shown with the temperature line plots. A screen shot of the display of the surface temperature along a sphere is shown in Figure 25. Again, remote users may click on the graphical window in order to find interpolated values for specific locations.

While there are various simple maintenance duties that we must perform, they are not much more then editing text files, and deleting old data files. For remote users, there is no need for any prior knowledge about the Java programming language, or the internal communication structure of the client-server programs. This system allows any interested party situated at any computer with Internet access to execute our numerical codes for the conditions that they require. The use of a graphical user interface allows people who are not fully familiar with the numerical code to easily input desired initial parameters and obtain results.

Presently, there is no such software package that allows engineers to easily allow remote access to their current numerical codes. In addition, there has not been a large movement towards using the Internet as an effective and efficient communications tool by computational researchers. This Internet system is the first system that is specifically designed to be used with our engineering related numerical codes and to display engineering related results. The use of the Internet for scientific computing is an inevitable extension of current numerical code development practices. The advantages of our Internet system are as follows :

- The new Internet system allows us to be able to remotely execute our current numerical codes from any computer that is connected to the Internet by using only a Web browser such as Netscape.
- This system does not require any advance knowledge of Internet processes beyond embedding a Java applet into a Web page, and editing text files.
- This system also allows engineers to download numerical data that has been produced by computations that were remotely executed.
- The Internet system can be designed so that it can be applied to any general computational code, not only our CFD numerical codes.

Ability for Remote Access

The most important aspect of our Internet system is the ability to remotely execute our numerical codes over the Internet in a secure and easy manner, using our existing executable files. Currently, engineering codes that are remotely executable over the Internet have had to be programmed in an Internet executable language, such as Java [5]. There is, however, an enormous amount of numerical programming that has been written in languages such as Fortran or C++ that is currently still being used. It is impractical for us to reprogram these codes into another language such as Java, since that would involve too much time. An Internet system has been developed where our existing numerical codes that are written in Fortran can be remotely executed over the Internet by combining them with an interface written with the Java language. This entire integrated software is accessed with a Web browser.

Efficient and Accurate Information Exchange

Another important aspect of this new Internet system, is the fact that it takes advantage of the fastest, most efficient and most accurate method of communication that is available to engineers. Traditional methods for the exchange of information in the scientific community have been through conferences, seminars, journals and personal communication. While these methods of communication are useful, they are not very efficient. The Internet allows communication between scientists and engineers to be almost instantaneous, regardless of the distance between them. An example of highly efficient communication can be seen in the email that is used for personal communication, and Web pages that are used to communicate information to a larger audience. Currently, many computational engineering solutions are presented at conferences and in journal papers. It is difficult, however, to present numerical data accurately, even by using data tables and small graphs. Space restrictions also force researchers to limit the amount of data that is to be presented. The results that are presented to other interested parties, therefore, is usually incomplete. With the Internet system, our computed solutions can be made available for interested parties to download. By providing entire numerical data files instead of a select number of graphs and tables, researchers will be able to analyze the data more closely and obtain a clearer picture of the numerical solution.

More Accurate Evaluation of Numerical Methods

The new Internet system can also be used to provide a method for better evaluations of our numerical codes. In many cases, different researchers will have their own version of a numerical code which solves an engineering problem. Researchers may have also developed a new solution method with which to solve an engineering problem. They may want to compare their computational results of some certain known flow conditions with solutions that have been produced by other numerical codes. This validation procedure is standard practice in the development of numerical computer codes, as it is used as a check to ensure that the numerical code is computing the correct solution. Additionally, it is regular practice now for journals to ask that any numerical results that are presented must include an analysis of the numerical accuracy [31]. Currently a clear validation is very difficult to achieve. Validation test cases are obtained from journals and conference papers. Results presented in these papers may be incomplete, may be displayed using a small graph, or may be the result of unknown initial conditions. This makes it very difficult for an accurate comparison to be made to determine if the numerical code is computing the solution accurately and correctly. The only way in which a clear comparison could be made is if both numerical codes were executed with the same flow conditions, and the output data compared.

Our new Internet system improves the validation procedure in two ways.

- The system can provide existing solution data files for interested parties to download and examine for themselves. These data files can be used to compare to data that is produced by the researchers themselves, and a conclusion drawn about the validity of the current methods. By providing entire data files, researchers will be provided more accurate information than what is presented in papers and journals.
- The other way this Internet system improves the validation procedure is for interested parties to remotely execute our numerical codes using initial conditions that they desire. By enabling outside parties access to our numerical codes, they can now choose what initial conditions they wish to study. This method ensures that the data that is generated has been produced by well defined initial conditions.

Availability to Novice Users

The Internet system is designed with a graphical user interface (GUI) that is clear and easy to understand. This feature makes our CFD numerical codes easier for anybody to use. With numerical codes, the user usually needs to have a degree of understanding of both the code itself and the methods that are being used. The input of initial conditions and a choice of methods may be unclear for people who are not fully familiar with the code's abilities. By using a GUI, any casual user will be able to successfully execute our numerical code without extensive prior knowledge of the code's methods or requirements. The GUI also contains help areas, where we have posted information that will assist the remote users in properly executing the code. The integrated code that predicts surface heating over hypersonic vehicles requires a level of knowledge concerning both CFD methods as well as analytical heating methods. By implementing the code over the Internet, and incorporating a user friendly GUI, users are asked to answer specific questions concerning the case which they want to study. User inputs such as freestream flow conditions and grid size are done in an easy to use manner which does not require any extensive prior knowledge. In addition, our numerical codes that are combined with this Internet system can be used instead of expensive commercial numerical codes. People who are interested in solving a particular engineering problem may find that our numerical code that exists on the Internet may be able to compute an accurate solution. This way, interested parties can compute and receive data that solves their engineering problem for virtually no cost. The Internet system can provide a fast, accurate solution that is much more cost effective than one that is provided by a commercial code.

Conclusions

The objective of this study was to use the Java programming language, taking advantage of the advanced graphical libraries, and develop an Internet based computational fluid dynamics (CFD) system. This system is designed to allow remote users the ability to access and execute CFD programs through the World Wide Web using only a Web browser. This study is motivated by the need for an easier method for which CFD results and methods can be communicated to interested parties, as well as the need for efficient accurate aerodynamic heating predictions. By using the Internet and Java applets, results can be easily and quickly accessed by remote users. This Internet system allows a user to remotely run flow cases and obtain and view results for a variety of flow conditions. A database of existing results is also available for the user to download and view.

We have developed an Internet system that not only allows us to remotely execute our CFD programs, but also to immediately view the results in a graphical manner using a built in post processor. The CFD program that we have developed involves a CFD inviscid flow solver integrated along with approximate aerodynamic heating methods. The heating methods require an inviscid flow result as an input parameter in order to accurately compute surface heating on hypersonic vehicles. By incorporating these numerical codes along with the Internet system, we are able to take advantage of the Internet's proven ability for fast and accurate data transfer. The use of the Java language has allowed us to incorporate advanced graphical functions that would have been much more difficult to program using another language. The result is our ability to execute this code and immediately view and download the results, using only a Web browser with the Java plug-in software installed.

Acknowledgements

This research was partially supported by NASA Dryden Flight Research Center under Grant NCC 2-374.

References

[1] Cnn.com. *http://www.cnn.com.*

[2] Amazon.com - earth's biggest selection. *http://www.amazon.com.*

[3] Cfd online. *http://www.cfd-online.com.*

[4] The source for java technology. *http://www.java.sun.com.*

[5] Java applets for engineering education. *http://www.engapplets.vt.edu.*

[6] R. Quinn and A. E. Kuhl. Comparison of flight-measured and calculated turbulent heat transfer on the X-15 airplane at Mach numbers from 2.5 to 6.0 at low angles of attack. *NASA TM X-939*, 1964.

[7] R. Quinn and M. Palitz. Comparison of measured and calculated turbulent heat transfer on the X-15 airplane at angles of attack up to 19.0. *NASA TM X-1291*, 1966.

[8] L. Gong, W. L. Richards, R. C. Monaghan, and R. D. Quinn. Preliminary analysis for a Mach 8 crossflow transition experiment on the Pegasus space booster. *NASA TM-104272*, 1993.

[9] C. D. Engel and S. C. Praharaj. MINIVER upgrade for the AVID system, Vol I: LANMIN user's manual. *NASA CR-172212*, 1983.

[10] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Computational Physics*, 126:202–228, 1996.

[11] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Computational Physics*, 49:357–393, 1983.

[12] S. Yamamoto and H. Daiguji. Higher-order-accurate upwind schemes for solving the compressible Euler and Navier-Stokes equations. *Computers and Fluids*, 22(2/3):259–270, 1993.

[13] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Computational Physics*, 77, 1988.

[14] N. A. Adams and K Shariff. A high resolution hybrid compact-ENO scheme for shock-turbulence interaction problems. *J. Computational Physics*, 127:27–51, 1996.

[15] C. Chiu and X. Zhong. Simulation of transient hypersonic flow using the ENO schemes. *AIAA 95-0469*, 1995.

[16] G. H. Furumoto, X. Zhong, and J. C. Skiba. Unsteady shock-wave reflection and interaction in viscous flows with thermal and chemical nonequilibrium. *AIAA 96-0107*, 1996.

[17] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes III. *J. Computational Physics*, 71:231–303, 1987.

[18] K. Mahesh, P. Moin, and S. K. Lele. The interaction of a shock wave with a turbulent shear flow. *Report No. TF-69, Thermosciences Division, Department of Mechanical Engineering, Stanford University*, 1996.

[19] H. C. Yee and R. F. Warming. Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. Computational Physics*, 57:327–360, 1985.

[20] O. Peroomian, R. E. Kelly, and S. Chakravarthy. Spatial simulations of a confined supersonic shear layer at two density ratios. *AIAA 96-0783*, 1996.

[21] S. Chakravarthy. Euler equations - implicit schemes and boundary conditions. *AIAA J.*, 21(5):699–706, 1983.

[22] J. A. Fay and F. R. Riddell. Theory of stagnation point heat transfer in dissociated air. *J. Aero. Sciences*, 25(2):73–85, 1958.

[23] I. E. Beckwith and J. J. Gallagher. Local heat transfer and recovery temperatures on a yawed cylinder at a Mach number of 4.15 and high Reynolds numbers. *NASA TR R-104*, 1961.

[24] H. Schlichting. *Boundary Layer Theory*, 1960.

[25] E. R. G. Eckert. Survey of heat transfer at high speeds. *WADC Technical Report 54-70*, 1954.

[26] E. R. G. Eckert. Survey of boundary layer heat transfer at high velocities and high temperatures. *WADC Technical Report 59-624*, 1960.

[27] E. R. van Driest. The problem of aerodynamic heating. *Aeron. Eng. Rev.*, 15(10), 1956.

[28] E. V. Zoby, J. N. Moss, and K. Sutton. Approximate convective heating equations for hypersonic flow. *J Spacecraft and Rockets*, 18(1):64–70, 1981.

[29] J. S. Pyle. Flight-measured wing surface pressures and loads for the X-15 airplane at Mach numbers from 1.2 to 6.0. *NASA TN D-2602*, 1965.

[30] M. E. Hillsamer and J. P. Rhudy. Heat-transfer and shadowgraph tests of several elliptical lifting bodies. *AEDC-TDR-64-19*, 1964.

[31] *AIAA Journal.*

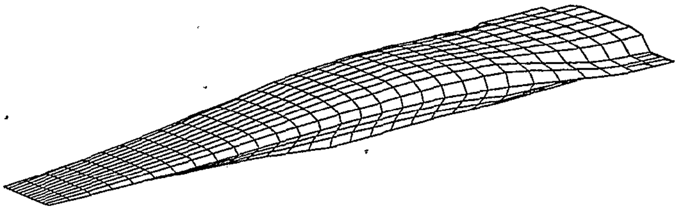


Figure 1: Surface grid of the Hyper-X vehicle.

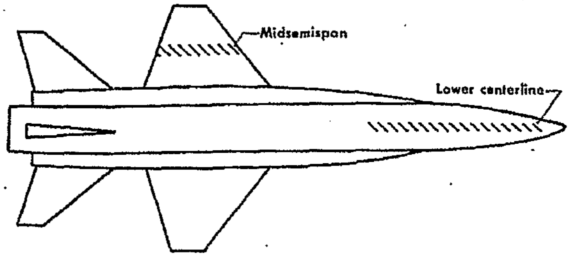


Figure 2: Schematic of the X-15 with experimental measurement locations.

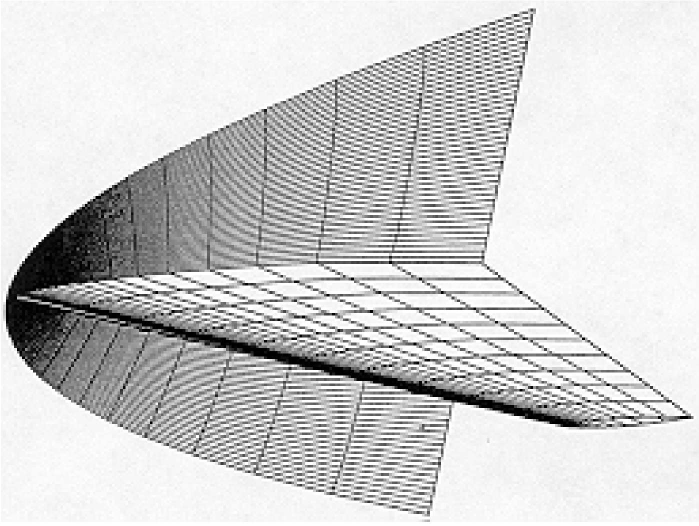


Figure 3: Three-dimensional numerical grid of the X-15 swept wing.

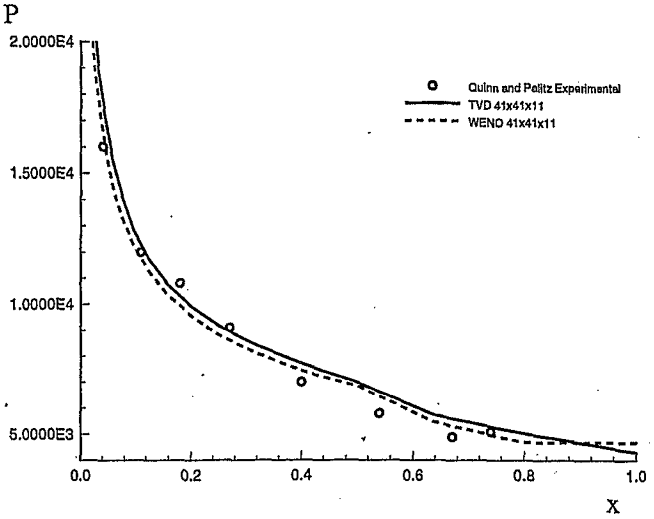


Figure 5: Pressure along the lower surface of the X-15 wing at Mach 5.1 with angle of attack of 2°.

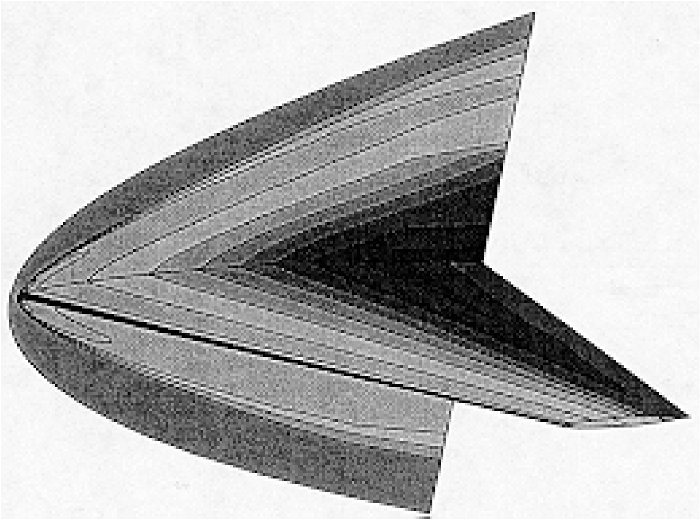


Figure 4: Numerically computed inviscid pressure contours for the X-15 wing at a Mach 5.1 with an angle of attack of 2°.

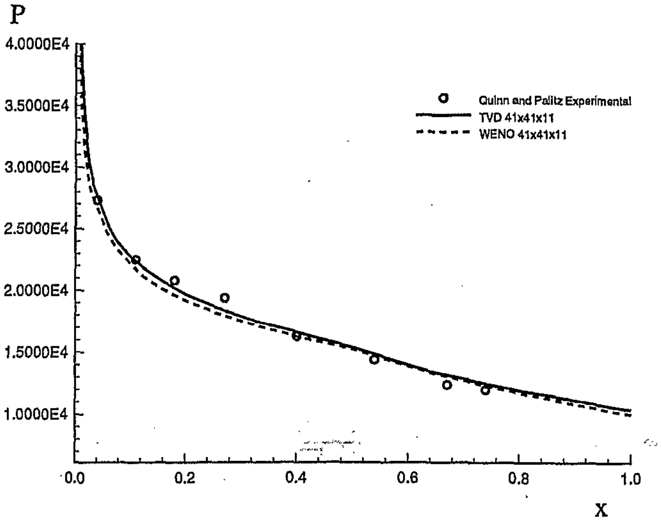


Figure 6: Pressure along the lower surface of the X-15 wing at Mach 4.15 with angle of attack of 10.3°.

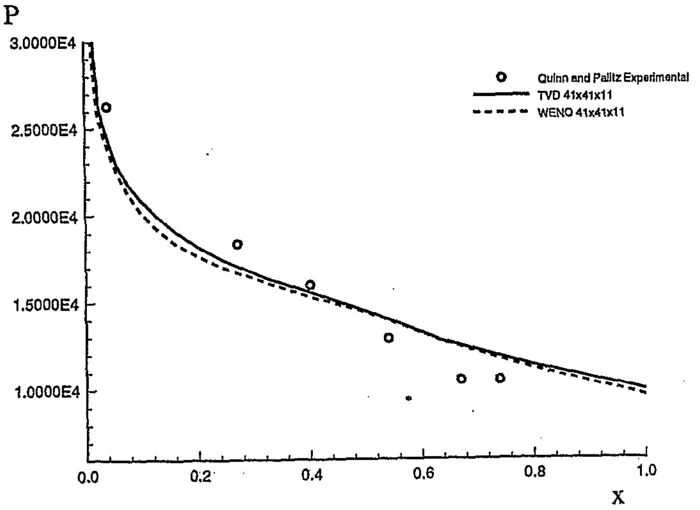


Figure 7: Pressure along the lower surface of the X-15 wing at Mach 4.98 with angle of attack of 16.3°.

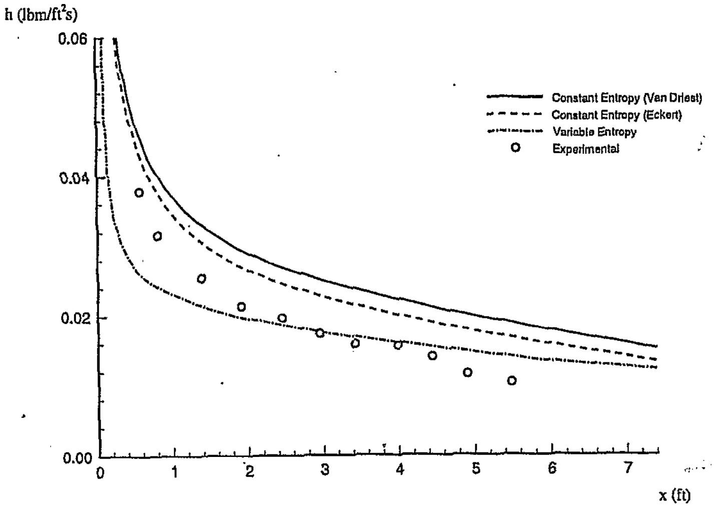


Figure 9: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 5.21 and angle of attack of 2.2°.

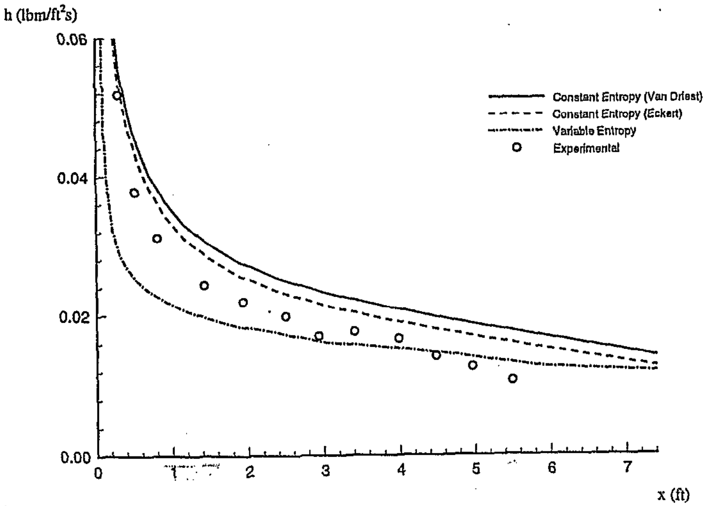


Figure 8: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 5.10 and angle of attack of 2.0°.

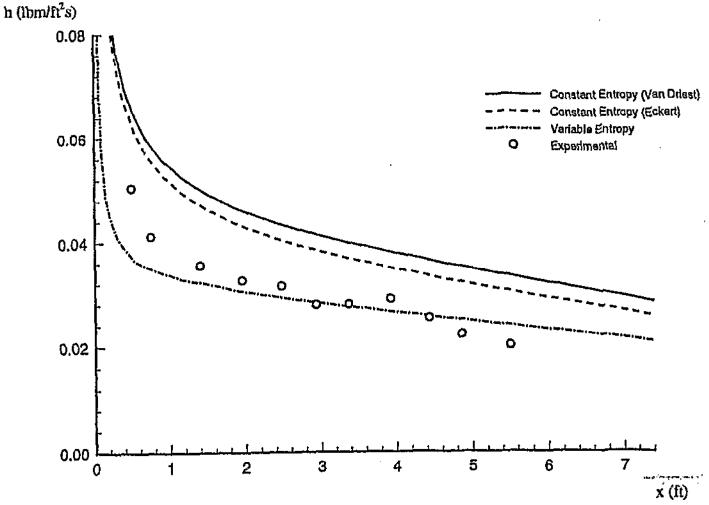


Figure 10: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 4.15 and angle of attack of 10.3°.

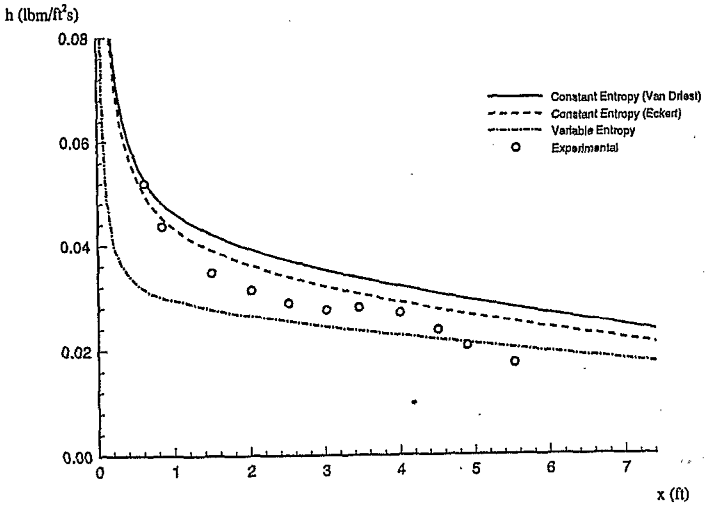


Figure 11: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 5.14 and angle of attack of 11.5°.

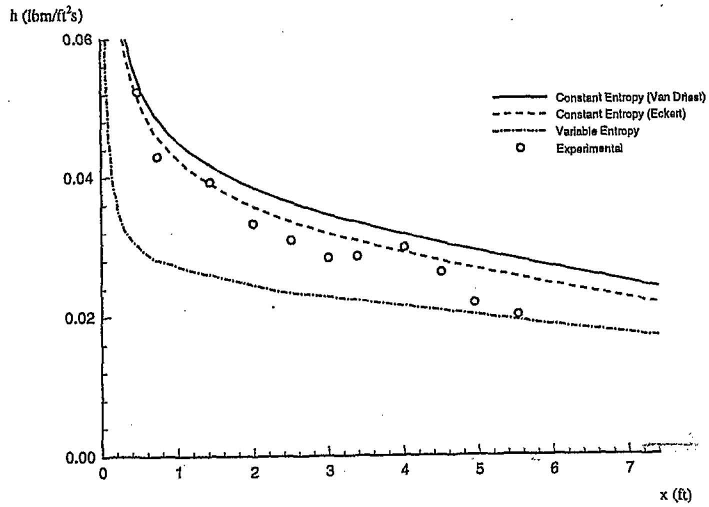


Figure 12: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 4.98 and angle of attack of 16.3°.

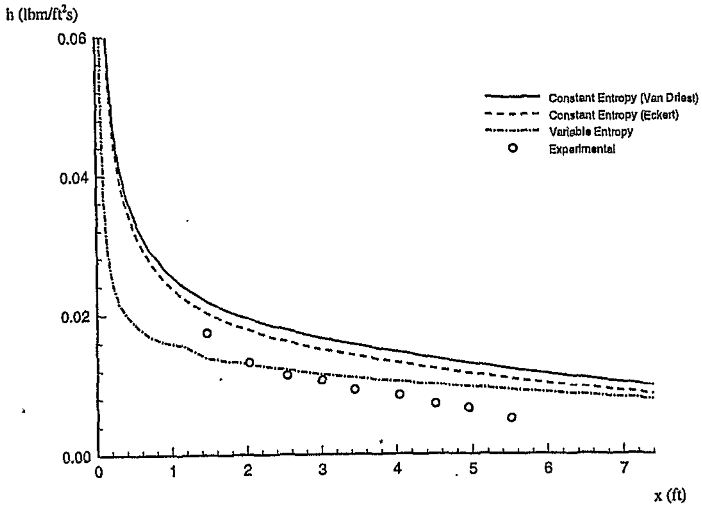


Figure 13: Heat transfer coefficient along the lower surface of an X-15 wing at Mach 5.50 and angle of attack of 0.9°.

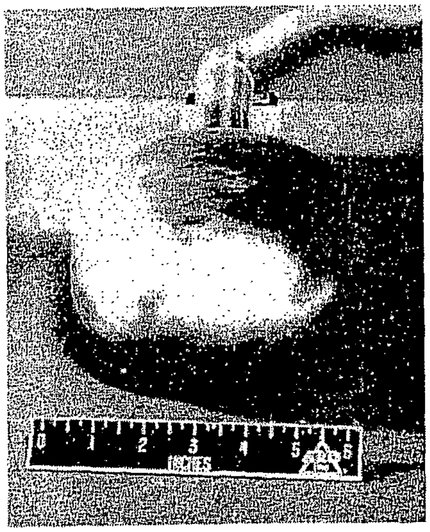


Figure 14: Photo of the experimental elliptical lifting body.

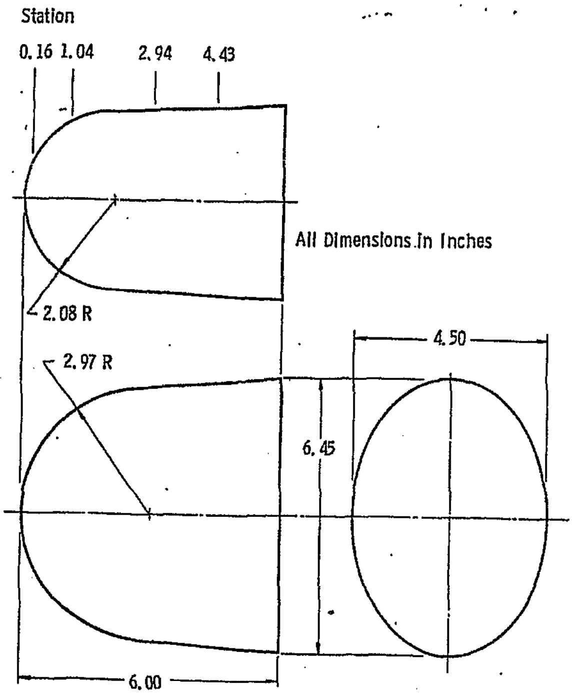


Figure 15: Schematic of the experimental elliptical lifting body heating test case with the two different radius' shown.

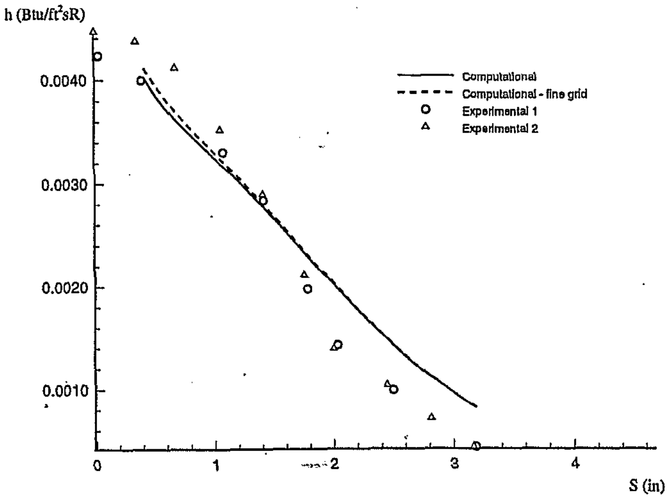


Figure 17: Heating comparison for the Reynolds number of 0.31 × 10^6 per foot at an angle of attack of 0° along the smaller radius streamwise direction.

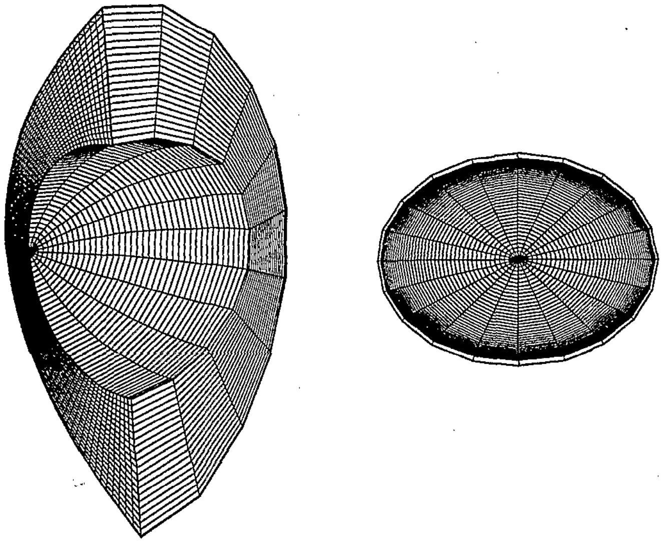


Figure 16: Three-dimensional numerical grid of the elliptical lifting body heating test case of size 41×21×21.

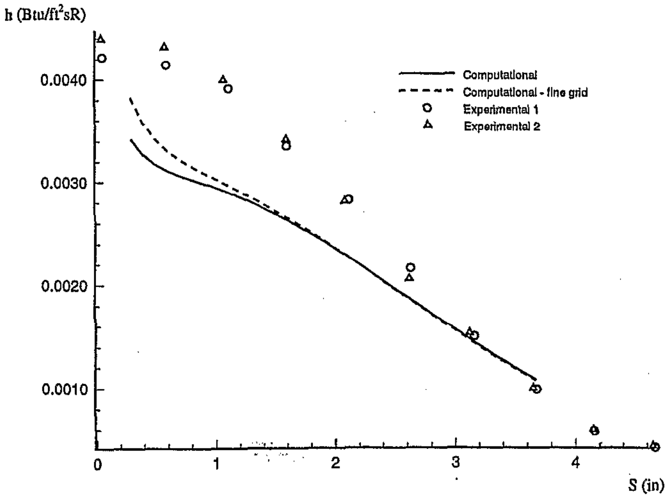


Figure 18: Heating comparison for the Reynolds number of 0.31 × 10^6 per foot at an angle of attack of 0° along the larger radius streamwise direction.

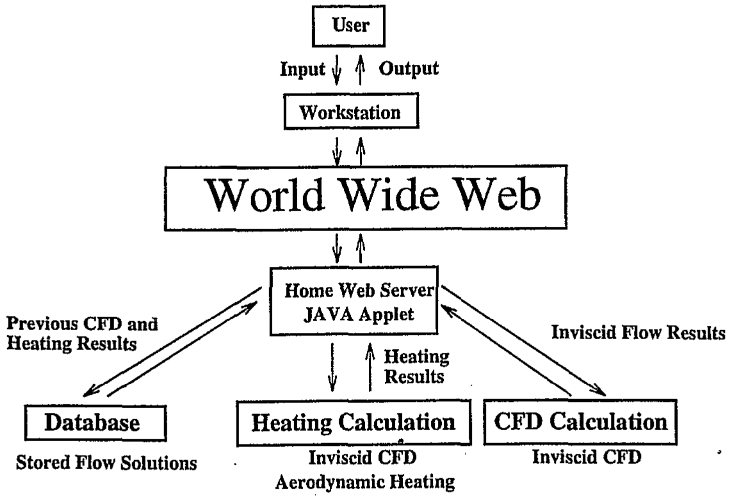
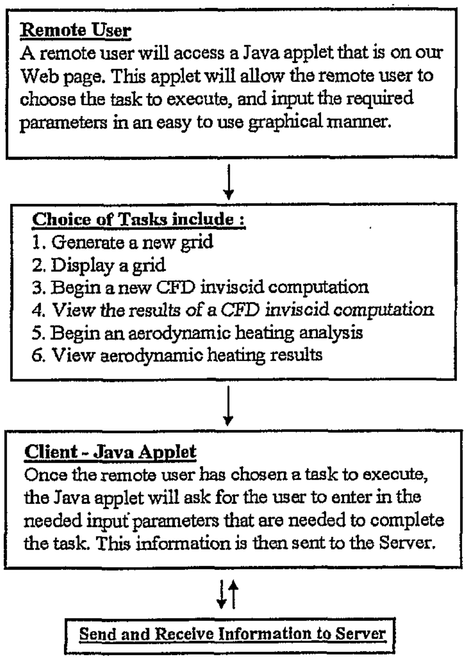


Figure 19: Internet application design structure.

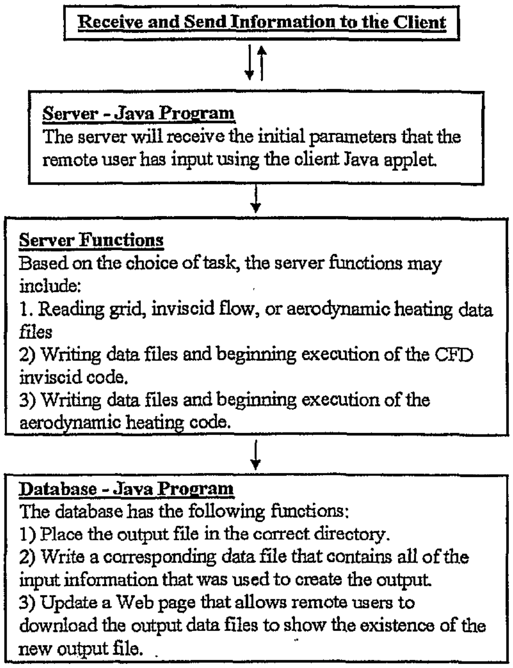

Remote User
A remote user will access a Java applet that is on our Web page. This applet will allow the remote user to choose the task to execute, and input the required parameters in an easy to use graphical manner.

Choice of Tasks include :
1. Generate a new grid
2. Display a grid
3. Begin a new CFD inviscid computation
4. View the results of a CFD inviscid computation
5. Begin an aerodynamic heating analysis
6. View aerodynamic heating results

Client - Java Applet
Once the remote user has chosen a task to execute, the Java applet will ask for the user to enter in the needed input parameters that are needed to complete the task. This information is then sent to the Server.

Send and Receive Information to Server

Figure 20: Internet system of client-server communication for the numerical code programmer.


Receive and Send Information to the Client

Server - Java Program
The server will receive the initial parameters that the remote user has input using the client Java applet.

Server Functions
Based on the choice of task, the server functions may include:
1. Reading grid, inviscid flow, or aerodynamic heating data files
2) Writing data files and beginning execution of the CFD inviscid code.
3) Writing data files and beginning execution of the aerodynamic heating code.

Database - Java Program
The database has the following functions:
1) Place the output file in the correct directory.
2) Write a corresponding data file that contains all of the input information that was used to create the output.
3) Update a Web page that allows remote users to download the output data files to show the existence of the new output file.

Figure 21: Internet system of client-server communication for the remote user.

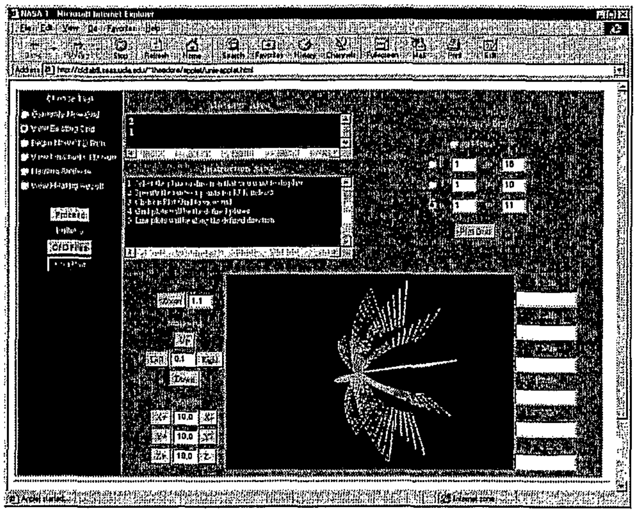


Figure 22: Screen shot of three dimensional sphere grid.

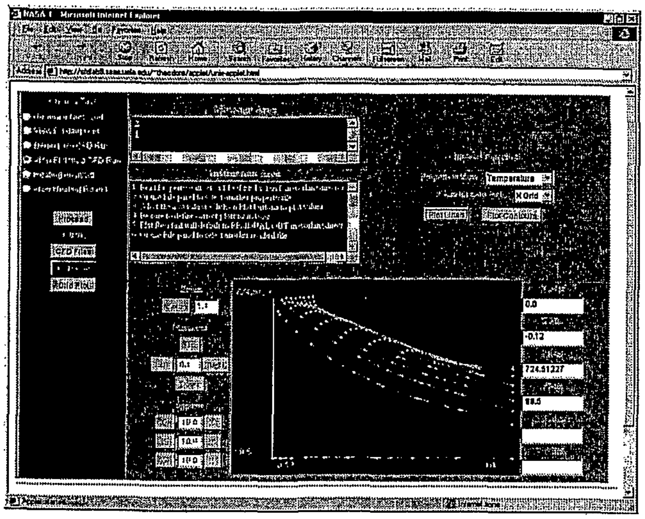


Figure 23: Screen shot of temperature line plot.

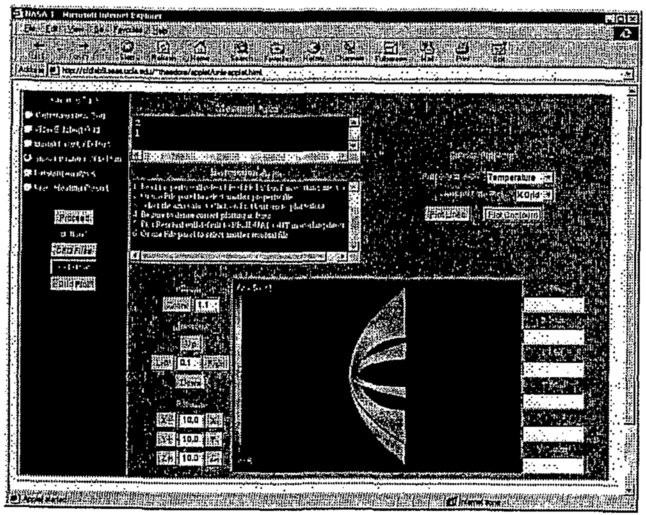


Figure 24: Screen shot of temperature contour over a sphere.

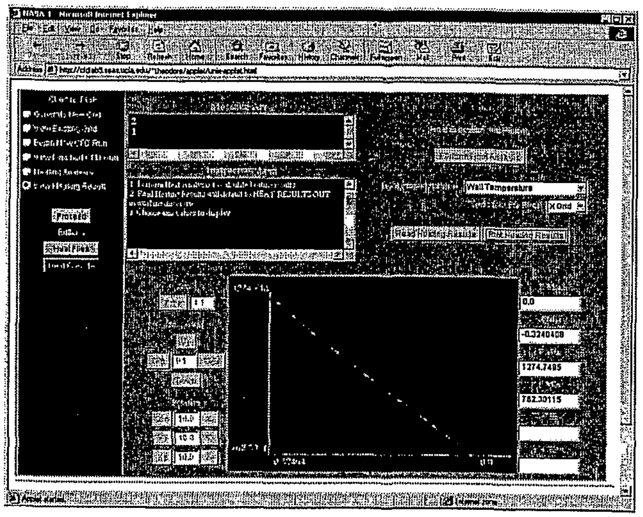


Figure 25: Screen shot of surface temperature heat result.