



## **METHOD IMPROVEMENTS IN THERMAL ANALYSIS OF MACH 10 LEADING EDGES**

**Ruth M. Amundsen**

National Aeronautics and Space Administration

Langley Research Center

Hampton VA 23681-2199

### **ABSTRACT**

Several improvements have recently been made in the thermal analysis methods for leading edges of a hypersonic vehicle. The leading edges of this vehicle undergo exceptionally high heat loads that incorporate extreme spatial gradients as well as severe transients. Due to the varying flight conditions, complex geometry, and need for thermal loads at many points along the trajectory, full computational fluid dynamics (CFD) analysis of the aeroheating loads is not feasible. Thus, engineering methods must be used to determine the aeroheating on the vehicle surfaces, and that must be utilized in the thermal analysis. Over the last year, the thermal analysis of a hypersonic vehicle has been enhanced in several ways. Two different engineering codes are used to predict aeroheating loads: one over the curve near the stagnation point, and the other on flat surfaces downstream of the leading edge. These two are matched together at the intersection point using a method that allows closer approximation of CFD results. User-developed FORTRAN, which is part of the thermal solver PATRAN Thermal, is used to accomplish this. The customizable FORTRAN code also allows use of many different time- and space-dependent factors, interpolation of the heat load in time and space, and inclusion of both highly swept and unswept grid structures. This FORTRAN is available to other PATRAN users who may want to accomplish a similar objective in analysis. Flux, rather than convective coefficient, is used to define heat loads, which allows more accurate analysis as well as better application of margins. Improvements have also been made in more efficient utilization of imported CAD geometry, by creating faces on solids to facilitate load application.

### **INTRODUCTION**

Earlier work by the author in the thermal analysis of hypersonic leading edges has been described elsewhere.<sup>1,2</sup> This paper will describe improvements in the methodology that have been made recently. Several hypersonic leading edges of varying geometries have been analyzed. The thermal solver used is MSC/PATRAN Thermal. The aeroheating loads for the hypersonic trajectory are generated in other software, and output in a text file format. The challenge is to import these aeroheating loads into PATRAN, interpolate in time and space to allow application to the PATRAN thermal model, and apply factors to the loads that are both time and spatially dependent. The main improvements that have been made in the last year are: to import and interpolate heat flux rather than the convective coefficient and recovery temperature; add flexibility so that the code can correctly interpolate from swept grid structures with varying directionality; incorporate time- and space-dependent factors; and alter the aeroheating loads at specific points on the geometry. The method for applying heat loads on the leading edge that are dependent on nodal temperature has been fully developed. Also, the method for verifying the interpolation has been optimized.

## HEAT FLUX IMPORT

A change was made in the interpolation software to import aeroheating flux, rather than the convective coefficient  $h_c$  and the fluid temperature. This was done for several reasons. First, the flux is the value actually calculated by the aeroheating software. The other two values were determined from flux, introducing possible inaccuracies. The fluid temperature is of questionable value at the range of Mach numbers encountered in this analysis. Second, the flux depends not only directly on the nodal temperature (as it also was in the original method using  $h_c$ ), but also indirectly. This indirect or second-order dependence of flux on the nodal temperature was not fully handled by importing  $h_c$  and computing flux based on a temperature difference. Third, the question of applying factors to the flux becomes more complex when  $h_c$  was used. When  $h_c$  was the value imported, any uncertainty factors could only be applied to  $h_c$ . This would lead to different temperatures at later times than when run without factors, and thus to a different (lower) flux. Thus, the factor was not really being applied to the total input flux. When flux is the imported value, any factor applied is actually increasing the flux by the correct amount at any given time.

In order to import flux rather than  $h_c$ , a different subroutine in the PATRAN ulib files was required. The `uhval.f` subroutine was used for import of  $h_c$  and fluid temperature. For import of flux, the `umicro.f` subroutine was utilized. The logic in the two subroutines is very similar. However, different variables are passed to, and used within, the two subroutines. In `umicro.f`, many of the internal PATRAN variables and arrays, such as `ITLIST`, `IFLIST`, and `MFID`, must be initialized with the correct sizes. Then, within `umicro.f`, the solution time and nodal position are evaluated and used for the time and spatial interpolation. A separate subroutine is called to perform the interpolation. Since the interpolation is based only on the nodal position in PATRAN, this method can be applied equally well to structured (brick) or unstructured (tetrahedral) meshes. It can also be applied to surface meshes.

Since the aeroheating flux is dependent on the temperature of the surface, surface temperatures at each time point must be output to the aeroheating program for calculation of flux. With new predicted fluxes, the PATRAN thermal solution is re-run, and temperatures again transferred for new calculation of aeroheating. This iteration normally only requires about three cycles to achieve closure (matching of input and output temperatures).

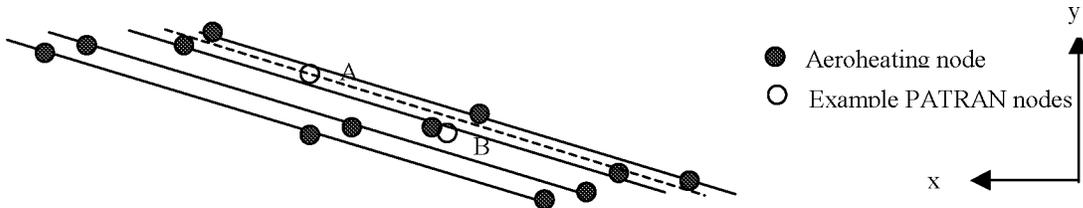
## IMPORT GRID PARAMETERS

The interpolation from one grid set to another would be fairly simple if the grids were both orthogonal to the same axes, and the gradients were small relative to the grid spacing. However, in the aeroheating grid, there are several parameters that make interpolation difficult.

First, the leading edge is sometimes swept at a severe angle, by as much as 70 degrees from normal to the flow. The vehicle is normally modeled with one axis parallel to the flow, and thus the leading edge is not in general parallel with an axis, but can be at a large angle to an axis. The grid of the aeroheating model usually follows the vehicle lines, and thus is swept with the leading edge, although the sweep angle is not constant and decreases substantially toward the aft end of the part. The steep gradient in heating that occurs on a hypersonic leading edge is normal to the line of the leading edge: i.e., the fall-off in heat flux is very abrupt in the direction directly away from the leading edge. If interpolation were done directly on the spatial coordinates of  $x$  and  $y$ , the flux could be interpolated incorrectly, since the gradient is dependent on the distance from the leading edge and not specifically on  $x$  or  $y$ . The sweep also makes it difficult to select the correct set of grid points to interpolate between. Using either  $x$  or  $y$  alone to select the grid point is not sufficient. It is also not enough to base it on the closest aeroheating point, since the point that is physically closest may have a very different flux based on its distance from the leading edge. In the software, an iterative set of equations is used to find the correct starting point in the aeroheating grid. Then, effective coordinates based on the distance from the leading edge are used for interpolation.

An example mesh is shown in Figure 1. The aeroheating grid is represented by closed circles, and two example nodes in the PATRAN mesh by open circles. The PATRAN mesh is much denser than the aeroheating grid, so for clarity only two example nodes are shown. The solid lines indicate the lines of

nodes in the aeroheating grid. One can see that for any given line of  $y$  points, the values are not constant, and for any given line of  $x$  points, the values change substantially for each change in  $y$ . Looking at example PATRAN node A, one can see that going only by the  $x$  coordinate would result in selecting a grid line much farther from the leading edge than is correct. Looking at example PATRAN node B, one can see that the  $y$  coordinate cannot be used as the sole criteria either. In fact, due to the relative coarseness and sweep angle of the aeroheating grid, the position of each PATRAN node must be determined based on an equivalent line, parallel to the aeroheating grid at that point. The dashed line in the figure indicates this equivalent line for node A. Then, the true grid cell containing the node can be determined, and the interpolation can be performed based on the equivalent position.



**Figure 1. Example interpolation mesh (viewed in 2D).**

Second, the aeroheating gridding does not follow a prescribed physical direction. The grid is ordered in the aeroheating program so as to conform to the sweep angle. On a part where part of the leading edge is swept back along positive  $z$ , and the other swept along negative  $z$  (as illustrated below in Figure 3), the physical ordering of the grid when brought into PATRAN will not be constant. The grid may be ordered such that the array increases in  $x$  and decreases in  $y$ , or the other way around. Or both values may increase or decrease together in the array. In the  $y$  axis (normal to the swept lines), the lines are close to constant value, but not quite. A line of nodes at some given position in the array (say, at the second value of  $y$  in each row) may not even be monotonic in  $y$ . The FORTRAN needs to be flexible enough to handle these changes in directionality.

Also, the aeroheating grid may not be oriented with the same  $x$  and  $y$ , and may not have the origin in the same place as the PATRAN model. The software handles changes in the axis orientation and origin position by modifying the values read in from the array, before they are used for interpolation.

## ALTERATIONS TO HEAT FLUX

The `umicro.f` subroutine is where factors can be applied to the heat flux value. There are several different conditions where factors are required. Effects on trailing edges, outer edges, coves, and gap regions, are all not completely predicted by the aeroheating code. Thus, factors must be applied to the loads in these regions to achieve an accurate thermal prediction. Regions that will all receive the same factor can be grouped, and a separate boundary condition applied. Then, the identification number of the boundary condition is used as the flag in `umicro.f` to trigger application of a given factor. The position in any of three axes can be used as a trigger for applying a set factor, or as a variable in calculating a spatially dependent factor to apply. The time in a transient solution can also be used as a trigger for changing factors.

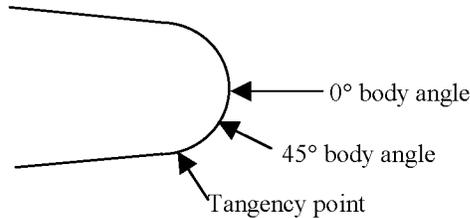
In some cases, the region where a factor is required is not a discrete part of the solid geometry. In most cases, the solid geometry is electronically imported from Pro/Engineer. There are often cases where a region is not broken out as a geometric entity on the solid that is imported, but due to aerodynamic considerations it must be treated with a separate factor. This can be done by manually selecting the affected elements, or by setting up a complex logic network of the spatial variables to define the region in question. These methods are somewhat cumbersome and time-consuming. A more straightforward method is to break the base geometry according to the surfaces that are desired. The base geometry will still exist as a solid, and can be handled as such for application of loads and boundary conditions. Faces may be added to the solid that facilitate loads application. One way to accomplish this is by breaking the solid (for example, with a plane), but this leaves multiple solids instead of the original one. A more elegant method is to break

the solid into its constituent surfaces, and split whichever surfaces are necessary for efficient load application. The solid can then be re-assembled from the desired surfaces, leaving a single solid containing all the required faces for loads application.

The aeroheating flux prediction is not reliable at the forward tangency point where the flat section meets the round of the leading edge. A method was developed to make this flux more consistent with CFD predictions. For sections where the aeroheating grid extends to the forward leading edge, the value of each point along the tangency line was replaced by a value calculated using Fay-Riddell methods. This substitution is done using logic in the subroutine where the array is read in.

## LEADING EDGE METHOD

The leading edge heating requires another method entirely, since the aeroheating code used for the flat acreage sections is not used on the leading edge. The value of heating is dependent on the trajectory, the leading edge geometry, the angular position on the leading edge, and on the temperature at the leading edge. The code uses basic Fay-Riddell methods to calculate stagnation heating, then modifies those values by the specified sweep angle and body angle. See Figure 2 for an illustration of the angular position (body angle). The Fay-Riddell calculation must be done for each time point in the trajectory. For swept leading edges, the temperature of all nodes at a given angular position may not be constant. Thus, in order to use a correct heat flux for all nodes, the heat flux must be altered based on the local temperature.



**Figure 2. Body angle definition.**

For nodes other than the reference node used to calculate heat flux, the flux into the node was factored using the following:

$$Q_{node} = Q_{ref} \frac{T_{stag} - T_{node}}{T_{stag} - T_{ref}} \quad [1]$$

where  $T_{node}$  and  $Q_{node}$  are the temperature of and flux into the given node, and  $T_{ref}$  and  $Q_{ref}$  are the temperature of and flux into the node whose temperature is used for heat flux calculation. This was done in PATRAN by making the heat into the leading edge nodes a product of three functions:

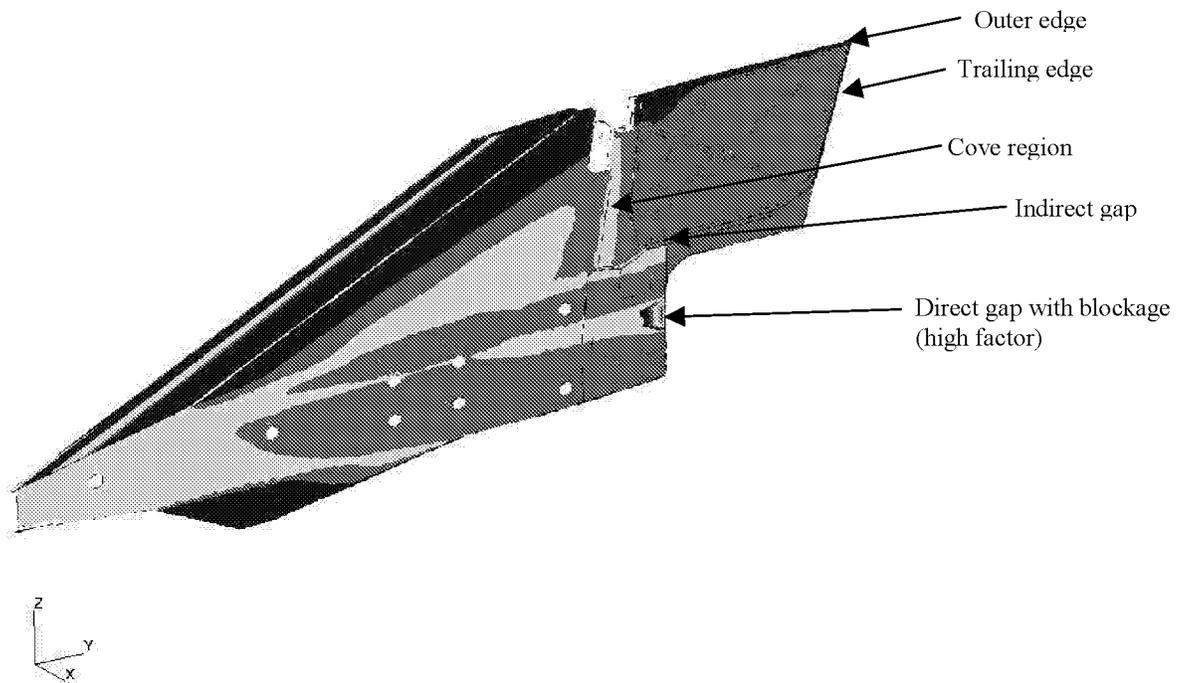
$$[Q_{ref}] * \left[ \frac{1}{T_{stag} - T_{ref}} \right] * [T_{stag} - T_{node}] \quad [2]$$

## VERIFICATION OF FLUX INTERPOLATION

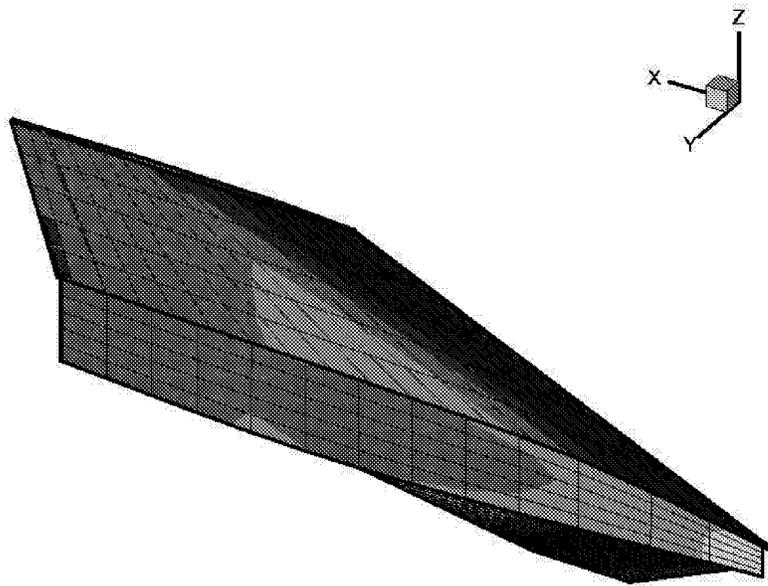
The verification of the interpolation of the flux would at first seem to be a simple matter of evaluating the flux in PATRAN versus the flux from the text formatted input file. However, a difficulty arises in that PATRAN Thermal does not plot flux as one of the standard parameters. The parameter that can be plotted directly from the nodal results file is nodal heating, which is the heat per node. This can be changed to flux

by dividing by the nodal sub-area. To accomplish this, one first needs to generate a file of the nodal sub-areas for each node that receives heating. One simple way to do this is to import the qmacro.dat file into any spreadsheet program, such as Microsoft Excel. The file can be reformatted to give only node number and nodal sub-area. Then, nodes that receive heating via more than one boundary condition must be summed to find their total sub-area. Once this is done, the file can be filtered so that each node appears only once.

The heat per node can be output from PATRAN into a text file by doing a Create\_Report from the Results menu, on only the surface nodes (those receiving aerodynamic heating). This text file can then be pulled into the same spreadsheet described above. Once the nodes are in the same order, it is a simple matter to calculate flux by dividing the heat by the nodal sub-area. The sheet of node number versus heat flux can be saved as a text file and pulled into PATRAN via PATRAN's shareware spreadsheet function. A result case can be created from the spreadsheet data, and the flux plotted on the model. The flux from the original text-formatted aeroheating file can be compared graphically by plotting it in Tecplot or a similar plotting program. The flux on an example PATRAN model is shown in Figure 3. This model utilizes 12 separate boundary conditions for acreage heat flux, to capture the discrete regions on the geometry, as well as five regions for leading edge heat flux. The mesh on the PATRAN model is not shown since it is such a dense mesh that it would completely obscure the flux contours (in this model there are more than 45,000 nodes). The flux directly from the aeroheating code, plotted in Tecplot, is shown in Figure 4. The units and scale are withheld due to concerns with data export; however, when the plots are evaluated in their original color format, on the same scale and units, the interpolation and factor application can be directly verified.



**Figure 3. Flux interpolated onto PATRAN model, with factors applied.**



**Figure 4. Aeroheating load on original grid.**

The flux can also be evaluated versus x, y and z, to obtain a quantitative comparison. The flux from the original file and as interpolated within PATRAN can be output versus x, y and z. Differences between the two can be determined, versus each variable, to assess the interpolation. Exact comparisons of the interpolation can be performed in this manner.

## CONCLUSIONS

The prediction of thermal behavior of hypersonic leading edges using PATRAN has been accomplished by development of the user-customizable FORTRAN available. Performance of accurate thermal analysis requires consideration of many factors. Interpolation of the heating loads from an aerodynamic code must be done carefully due to the highly swept and directional nature of the grid, as well as the steep gradients in heat flux away from a leading edge. Interpolation of heat flux, rather than convective coefficient, has been found to make the aeroheating prediction more accurate, as well as facilitating the application of factors. Leading edges with various sweep angles, grid directionality, and grid orientation are all handled by the FORTRAN developed for this interpolation. Factors that are constant, spatially dependent or time dependent can be applied to discrete areas of the geometry. Discrete areas for load application can be created as needed by breaking the original geometry prior to meshing or load application. A method for robust application of the stagnation point heating at the leading edge has been developed. A method for verifying the flux interpolation by plotting flux on the PATRAN model has been established.

## ACRONYMNS AND SYMBOLS

CFD	computational fluid dynamics
$h_c$	convective coefficient
$Q_{ref, node}$	Flux at a reference node, and at a given node, respectively
$T_{stag, ref, node}$	Temperature at the stagnation point (fluid), at a reference node, and at a given node, respectively

## **ACKNOWLEDGEMENTS**

The author wishes to acknowledge the contributions of Phil Yarrington in providing the original software, Chuck Leonard and Vince Cuda for assistance with aeroheating, and Mike Lindell for general modeling assistance.

## **REFERENCES**

1. Ruth M. Amundsen, "Comparison of Integrated Analysis Methods for Two Model Scenarios," Ninth Conference, Workshop and Product Presentation on Thermal and Fluids Analysis Tools and Methods, Cleveland, Ohio, August 31--September 4, 1998.
2. R. M. Amundsen, A. O. Torres, C. P. Leonard, Thermal Analysis Of The Hyper-X Research Vehicle Wing: Mach 7 Design, 1997 Joint Army NAVY NASA Air Force (JANNAF), West Palm Beach, Florida, October 30, 1997.