Improvements in Thermal Protection Sizing Capabilities for TCAT

Final Report

NAG8-1597
NASA Marshall Space Flight Center

For a NASA Cooperative Agreement entitled

Conceptual Design for Advanced Space Transportation Systems

Submitted by:

Dr. John R. Olds (PI)
Stephen Izon (thesis author)

School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

February 2002
Table of Contents

Abstract .......................................................................................................................... iv
List of Tables .................................................................................................................. v
List of Figures ............................................................................................................... vi
Nomenclature ............................................................................................................... vii
List of Symbols .......................................................................................................... viii
1.0 Project Motivation ............................................................................................... 1
   Ground Hold Analysis Tool ....................................................................................... 1
   Crank-Nicolson Method and TCAT .......................................................................... 3
2.0 Ground Hold Analysis Tool Development .......................................................... 5
   Numerical Solution Method ..................................................................................... 5
   Optimization Method .............................................................................................. 8
   Program Execution .................................................................................................. 10
3.0 Ground Hold Analysis Test Results ..................................................................... 12
4.0 Crank-Nicolson Method and TCAT ..................................................................... 19
   Finite Difference Equations ................................................................................... 19
5.0 World Wide Web Interface .................................................................................... 21
   Ground Hold Analysis ............................................................................................ 22
   Aeroheating Analysis ............................................................................................... 24
   MINIVER File Upload ............................................................................................. 24
6.0 Conclusions ........................................................................................................... 26
Appendix A: STS-1 Reentry Trajectory .................................................................... 27
Appendix B: Windward Aeroheating Analysis Input File ............................................ 28
Appendix C: Leeward Aeroheating Analysis Input File .............................................. 29
Appendix D: Windward Ground Hold Analysis for LOx Tank Input File ....................... 30
Appendix E: Leeward Ground Hold Analysis for LOx Tank Input File ......................... 31
Appendix F: Windward Ground Hold Analysis for LH2 Tank Input File ....................... 32
Appendix G: Leeward Ground Hold Analysis for LH2 Tank Input File ......................... 33
Appendix H: ‘single.cgi’ Ground Hold Single Analysis CGI Script ............................... 34
Appendix I: ‘single.cgi’ Ground Hold Single Analysis CGI Script ............................... 34
Appendix I: ‘compare.cgi’ Ground Hold Multiple Analysis CGI Script ............................................. 49
Appendix J: ‘ground.f’ FORTRAN Source Code ............................................................................. 61
Appendix K: ‘ground2.f’ FORTRAN Source Code ......................................................................... 65
Appendix L: ‘one.f’ FORTRAN Subroutine .................................................................................. 69
Appendix M: ‘two.f’ FORTRAN Subroutine .................................................................................. 71
Appendix N: ‘three.f’ FORTRAN ................................................................................................. 73
Appendix O: ‘four.f’ FORTRAN Subroutine ................................................................................... 75
Appendix P: ‘five.f’ FORTRAN Subroutine ................................................................................... 78
References ....................................................................................................................................... 81
Abstract

TCAT, originally developed for the Space Systems Design Lab at the Georgia Institute of Technology is a conceptual design tool capable of integrating aeroheating analysis into conceptual reusable launch vehicle design. It provides TPS unit thicknesses and acreage percentages based on the geometry of the vehicle and a reference trajectory to be used in calculation of the total cost and weight of the vehicle design.

TCAT has proven to be reasonably accurate at calculating the TPS unit weights for in-flight trajectories; however, it does not have the capability of sizing TPS materials above cryogenic fuel tanks for ground hold operations. During ground hold operations, the vehicle is held for a brief period (generally about 2 hours) during which heat transfer from the TPS materials to the cryogenic fuel occurs. If too much heat is extracted from the TPS material, the surface temperature may fall below the freezing point of water, thereby freezing any condensation that may be present at the surface of the TPS. Condensation or ice on the surface of the vehicle is potentially hazardous to the mission and can also damage the TPS. It is questionable whether or not the TPS thicknesses provided by the aeroheating analysis would be sufficiently thick to insulate the surface of the TPS from the heat transfer to the fuel. Therefore, a design tool has been developed that is capable of sizing TPS materials at these cryogenic fuel tank locations to augment TCAT's TPS sizing capabilities.

In addition, the Crank-Nicolson method was substituted for the simple implicit (Laasonen) Method used in the numerical solution for the interior nodal points of TCAT. The simple implicit method has first-order accuracy with a truncation error of $O(\Delta t, \Delta x)$. The Crank-Nicolson Method makes use of trapezoidal differencing to achieve second-order accuracy with a truncation error of $O(\Delta t^2, \Delta x^2)$. This method was selected because it is the same method used in the commercial heating code SINDA. The Crank-Nicolson method is proven to require fewer nodes and less execution time than the simple implicit method. Therefore, the solution process may take less time to complete, and the numerical accuracy should be better than that of the simple implicit method.
List of Tables

Table 1: TPS Thicknesses for Windward Surface of 10° Half-Angle Cone for In-Flight Aeroheating with LI-2200 Tiles ................................................................. 13
Table 2: TPS Thicknesses for Leeward Surface of 10° Half-Angle Cone for In-Flight Aeroheating with CFBI Blankets ............................................................... 13
Table 3: TPS Design Results for 10° Half-Angle Cone for In-Flight Aeroheating ................................................................. 14
Table 4: Ground Hold Analysis Parameters ........................................................................... 14
Table 5: TPS Sizing Results for Ground Hold Analysis Above a LOx Fuel Tank .......... 14
Table 6: TPS Sizing Results for Ground Hold Analysis Above a LH2 Fuel Tank .......... 14
List of Figures

Figure 1: Design Structure Matrix ................................................................. 1
Figure 2: TPS Sizing Process ........................................................................ 2
Figure 3: Schematic of Program Flow Logic ............................................... 11
Figure 4: Schematic of 10° Half-Angle Cone ............................................. 12
Figure 5: Schematic of CFBI TPS Configuration ...................................... 12
Figure 6: Schematic of LI-2200 TPS Configuration .................................. 12
Figure 7: Nodal Temperature Histories for LI-2200 Sized Over LH2 Tank .... 15
Figure 8: Nodal Temperature Histories for CFBI Sized Over LH2 Tank .... 15
Figure 9: Nodal Temperature Histories for LI-2200 Sized Over LOx Tank ... 16
Figure 10: Nodal Temperature Histories for CFBI Sized Over LOx Tank ...... 16
Figure 11: Nodal Mesh Map for Interior Nodes .......................................... 19
Figure 12: Screen Shot of TCAT Welcome Page ....................................... 21
Figure 13: Screen Shot of TCA1 Ground Hold Single Analysis Page ......... 22
Figure 14: Screen Shot of TCAT Ground Hold Multiple Analysis Page .... 23
Figure 15: Screen Shot of TCAT Aeroheating Analysis Page ................. 24
Figure 16: Screen Shot of TCAT MINIVER File Upload Page .................. 25
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADS</td>
<td>Automated Design Synthesis</td>
</tr>
<tr>
<td>CFBI</td>
<td>Composite Flexible Blanket Insulation</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>DSM</td>
<td>Design Structure Matrix</td>
</tr>
<tr>
<td>GrEx</td>
<td>Graphite Epoxy</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>LI-2200</td>
<td>22 lbm/ft³ Rigid Ceramic Tile Insulation</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>RCC</td>
<td>Reinforced Carbon-Carbon</td>
</tr>
<tr>
<td>RLV</td>
<td>Reusable Launch Vehicle</td>
</tr>
<tr>
<td>SiC</td>
<td>Silicon Carbide</td>
</tr>
<tr>
<td>STS</td>
<td>Space Transportation System</td>
</tr>
<tr>
<td>TCAT</td>
<td>Thermal Calculation Analysis Tool</td>
</tr>
<tr>
<td>TPS</td>
<td>Thermal Protection System</td>
</tr>
<tr>
<td>TPSX</td>
<td>Thermal Protection System Expert</td>
</tr>
<tr>
<td>TUF1</td>
<td>Toughened Uni-Piece Fibrous Insulation</td>
</tr>
</tbody>
</table>
List of Symbols

\( \alpha \)  
Thermal diffusivity, \( \left[ \frac{m^2}{s} \right] \) or \( \left[ \frac{ft^2}{s} \right] \)

\( \Delta x \)  
Spatial Step, \( [m] \) or \( [ft] \)

\( \Delta t \)  
Temporal Step, \( [s] \)

\( \frac{\partial T}{\partial t} \)  
Partial derivative of temperature with respect to \( t \)

\( \frac{\partial^2 T}{\partial x^2} \)  
Second partial derivative of temperature with respect to \( x \)

\( \frac{dT}{dx} \)  
Derivative of temperature with respect to \( x \)

\( \varepsilon \)  
Material emissivity

\( \sigma \)  
Boltzmann constant, \( 5.67051 \times 10^{-8} \left[ \frac{W}{m^2 \cdot K^4} \right] \) or \( 4.75644 \times 10^{-13} \left[ \frac{Btu}{s \cdot m^2 \cdot K^4} \right] \)

\( f_i \)  
System of equations formed after discretization of the heat equation

\( i \)  
Spatial Position Index

\( J_{i,j} \)  
Jacobian matrix

\( k \)  
Material thermal conductivity, \( \left[ \frac{W}{m \cdot K} \right] \) or \( \left[ \frac{Btu}{lbm \cdot s \cdot R} \right] \)

\( L \)  
Material stack thickness, \( [m] \) or \( [ft] \)

\( N \)  
Maximum number of nodes

\( n \)  
Time index

\( q \)  
Heat transfer rate, \( \left[ \frac{W}{m^2} \right] \) or \( \left[ \frac{Btu}{ft^2 \cdot s} \right] \)

\( T \)  
Temperature, \( [K] \) or \( [^\circ F] \)

\( t \)  
Time, \( [s] \)

\( x \)  
Spatial position in material normal to exposed surface, \( [m] \) or \( [ft] \)
1.0 Project Motivation

Ground Hold Analysis Tool

Conceptual design of launch vehicles is generally an iterative process in which disciplinary analyses are performed individually. At the Space Systems Design Lab at the Georgia Institute of Technology, research in the conceptual design of third generation launch vehicles flows through a sequence of disciplinary analyses found in a Design Structure Matrix. The results of each disciplinary analysis are sent to other disciplines whose analyses are dependent upon those results. A typical design structure matrix is illustrated in Figure 1.

![Design Structure Matrix](image)

**Figure 1: Design Structure Matrix**

From Figure 1 we can see that the TPS sizing is dependent on the trajectory and configuration disciplines. More specifically, trajectory data such as altitude, relative velocity, and angle of attack versus time as well as geometry data are required in order to perform the aeroheating analysis and generate the TPS thicknesses and unit weights.
Once the TPS material thicknesses and unit weights are determined for the iteration, the data is sent to the weights and sizing discipline.

TCAT, the Thermal Calculation Analysis Tool is a conceptual design tool originally developed for the Space Systems Design Lab to integrate aeroheating analysis into conceptual reusable launch vehicle design. TCAT is a FORTRAN code that uses finite difference methods to perform transient in-depth one-dimensional conduction analysis over the center mold line of the vehicle. TCAT couples with MINIVER, TPSX, and ADS to correctly size the vehicle's thermal protection system. MINIVER is an Aeroheating code that produces centerline radiation equilibrium temperatures, convective heating rates, and head loads over simplified vehicle geometries. TPSX is a NASA Ames material properties database that is available on the World Wide Web. ADS (Automated Design Synthesis) is a numerical optimizer that uses algorithms to solve constrained and unconstrained design problems. A schematic of the TPS sizing process using TCAT is shown in Figure 1.

When TCAT was originally created, its goal was to size TPS materials for in-flight aeroheating. However, sizing the TPS solely for in-flight aeroheating may not guarantee that the vehicle will meet all of its performance requirements. During in-flight operations, the surface of the vehicle experiences very high temperatures as well as high heat rates induced by convective heating. Therefore, TPS materials are selected so that the surface temperature of the materials does not exceed their operating temperature...
limits. The materials are sized so that the temperatures experienced by the rest of the vehicle, namely the vehicle structure or tank structure, do not exceed some critical temperature, which would result in thermal failure.

What TCAT did not account for is that during ground hold operations, the vehicle is held at the launch facility for a brief period while it is prepared for launch. The vehicle is exposed to the ambient temperature of the air, as opposed to the high temperatures caused by aeroheating during flight. Heat is transferred from the atmosphere to the surface of the TPS at a much lower rate and heat transfer occurs within the depth of the TPS material stack. TPS materials applied over a cryogenic tank must be sized to prevent the formation of ice on the outer surface of the TPS material. Since the fuel inside the cryogenic fuel tank is very low (423.67 °F for LH2, -297.67 for LOx), heat is extracted from the tank structure as well as the TPS materials directly in contact with the tank structure, causing temperatures to decrease throughout. If the temperature of the outer surface of the TPS falls below the freezing point of water (32°F) then frosting will occur. Condensation or frosting on the surface of the TPS material is considered hazardous since it will negatively affect the performance of the vehicle or damage the TPS. Therefore, a design tool capable of sizing TPS materials during ground hold operations must be developed to ensure that the overall TPS thicknesses selected for the vehicle will be adequate to satisfy not only the in-flight aeroheating constraints, but also the ground hold constraints.

Crank-Nicolson Method and TCAT

At the conceptual design level, since the level of fidelity is not as high as it would be for detailed design, conceptual design tools that provide a high level of accuracy with low computational times are desired. TCAT is solves the one-dimensional heat equation numerically using a simple implicit (Laasonen) method for the interior nodes of a TPS material stackup. Although this method is unconditionally stable it is only first order accurate with a truncation error of $O((\Delta r)(\Delta x)^2)$. The Crank-Nicolson method is another implicit numerical scheme that is also unconditionally stable; however, it makes use of trapezoidal differencing to achieve second-order accuracy with a truncation error of
$O((\Delta t)^2, (\Delta x)^2)$. This method is the same as that used in the commercial heating code SINDA. The solution method of the resulting system of equations is the same as for the simple implicit method, but generally the Crank-Nicolson method requires fewer nodes and less execution time. Therefore, substitution of the Crank-Nicolson method for the simple implicit method may make TCAT a more valuable and practical conceptual design tool.
2.0 Ground Hold Analysis Tool Development

The original version of TCAT operates by taking aeroheating data from MINIVER (a commercial aeroheating code) and sizing the TPS material for each body point specified by MINIVER. The data provided by MINIVER is the radiation equilibrium temperature and the convective heat rate versus time. During groundhold it is assumed that the heat transfer is constant and determined by the ambient temperature of the air. Since the ambient temperature of the surrounding atmosphere is constant, and the heat transfer rate is constant, it is not necessary to create a file like those provided by MINIVER. Instead, these constants can be entered explicitly into the numerical solution process.

The ground hold analysis only needs to be applied to one representative body point above a cryogenic fuel tank location. This is because it is assumed that the conditions at one body point location would be the same as any other location above the same cryogenic fuel tank. The unit weight calculated from the single body point analysis can then be applied to the total area of the vehicle surface above cryogenic fuel tank locations for which the same TPS material will be used under the same ambient conditions.

Numerical Solution Method

TCAT uses a fully implicit method to solve the parabolic, one-dimensional unsteady heat conduction equation by marching in time.

\[
\frac{\partial T}{\partial t} + \frac{\partial}{\partial x} \frac{\partial T}{\partial x} = 0
\]  

The boundary condition given by equation (2) is applied to the top surface of the TPS material. This equation is the energy balance relationship for the top surface of the TPS material. The convective, radiative, and conductive heat transfers are summed to equal zero in order to preserve the conservation of energy.
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

\[ q_{\text{conv}} - \varepsilon \alpha T^4 + k \frac{dT}{dx} = 0 \text{ at } x = 0 \]  

(2)

However, for groundhold, a constant heat transfer mechanism is assumed, where the heat transfer coefficient is \(1 \text{ BTU/(ft}^2\cdot\text{hour} \cdot ^\circ\text{F})\). Here, the atmosphere is treated as an infinite atmosphere in which heat transfer to the surface TPS material can occur, but the temperature of the atmosphere does not change. Therefore, the boundary condition given in equation (3) is used for the groundhold analysis instead of equation (2).

\[ k \frac{dT}{dx} = q_{\text{in}} \text{ at } x = 0 \]  

(3)

At the backface of the TPS material the original TCAT uses the boundary condition in equation (4), which assumes that there is an adiabatic wall at the backface of the material.

\[ \frac{dT}{dx} = 0 \text{ at } x = L \]  

(4)

However, for groundhold an isothermal boundary condition is applied.

\[ T = T_{\text{fuel}} = \text{const} \text{ at } x = L \]  

(5)

This isothermal assumption is applied because during groundhold it is assumed that the temperature of the fuel remains constant, which is in direct contact with the backface of the cryogenic fuel tank structure. Heat can be transferred from the material to the fuel at the interface of the tank and fuel; however, it is assumed that the heat transfer that does occur does not affect the overall temperature of the fuel. This assumption is valid because the fuel can be treated as an infinite reservoir, due to the large volume of fuel that is contained inside the cryogenic fuel tanks.
At the surface, using the boundary condition given by equation (3) the finite difference equation that results is given in equation (6).

\[ T_i^n + \frac{2\alpha_i \Delta t}{\Delta x} \frac{q}{k_i} = -\frac{2\alpha_i \Delta t}{\Delta x^2} (T_{i+1}^{n+1} - T_i^{n+1}) + T_i^{n+1} \]  

(6)

In equation (6), the constant \( q \) is the heat transfer from the atmosphere, and is equal to the ambient temperature times the heat transfer coefficient.

At the interior nodes, using a simple implicit central finite difference scheme, the resulting discretization is given by equation (7).

\[ T_i^n = -\frac{\alpha_i \Delta t}{\Delta x^2} (T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + T_i^{n+1} \]  

(7)

On the back surface of the material, the isothermal boundary condition of equation (5) is applied and results in equation (8).

\[ T_N^n = T_N^{n+1} = T_{fuel} = const \]  

(8)

Examining these equations, we see that we now have a system of linear equations. In the numerical method used by TCAT for the aeroheating analysis, a system of non-linear equations resulted, because of the fourth-order temperature dependence of the radiative heat transfer term. Since our system is now linear, the solution process can be made much simpler because we no longer have to reiterate to converge to the solution. The system of equations can be rearranged into the form \( Ax = b \) as shown in equation (9) to simplify the solution process, where \( A \) is a matrix of coefficients, \( x \) is an array of temperatures at the next time step, and \( b \) is an array of temperatures at the current time step.
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

\[
\begin{bmatrix}
1 + \frac{\alpha_1 \Delta t}{\Delta x^2} & -\frac{2\alpha_1 \Delta t}{\Delta x^2} & 0 & 0 & 0 & 0 \\
-\frac{\alpha_1 \Delta t}{\Delta x^2} & \frac{2\alpha_1 \Delta t}{\Delta x^2} & -\frac{\alpha_1 \Delta t}{\Delta x^2} & 0 & 0 & 0 \\
0 & -\frac{\alpha_1 \Delta t}{\Delta x^2} & \frac{2\alpha_1 \Delta t}{\Delta x^2} & -\frac{\alpha_1 \Delta t}{\Delta x^2} & 0 & 0 \\
0 & 0 & 0 & -\frac{\alpha_1 \Delta t}{\Delta x^2} & \frac{2\alpha_1 \Delta t}{\Delta x^2} & -\frac{\alpha_1 \Delta t}{\Delta x^2} \\
0 & 0 & 0 & 0 & \frac{2\alpha_1 \Delta t}{\Delta x^2} & -\frac{\alpha_1 \Delta t}{\Delta x^2} \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
T_1 \\
T_2 \\
T_3 \\
\vdots \\
T_{N-1} \\
T_N
\end{bmatrix}
= 
\begin{bmatrix}
T_1^{*+1} \\
T_2 \\
T_3 \\
\vdots \\
T_{N-1} \\
T_N
\end{bmatrix}
\]

This solution of this system is easily obtained using the Thomas Algorithm, which is a numerical algorithm capable of solving tridiagonal banded matrices.

Optimization Method

For the ground hold analysis problem, the only constraint that needs to be enforced is the surface frosting condition. As long as the temperature of the surface of the TPS stays above the designated frosting condition, then the constraint is satisfied. If there were no heat transfer from the atmosphere, then we would expect the temperature profile of the surface of the TPS to decrease with time. In other words, the maximum temperature would be at time \( t = 0 \) and the minimum temperature would be at the end of the analysis. If there is heat transfer from the atmosphere, then we might expect the temperature of the surface to rise slightly, and then decrease to a minimum temperature at the end of the analysis. In both cases, we see that the minimum temperature is always at the end of the ground hold analysis time. Therefore, our only constraint would be to ensure that the temperature of the surface at the end of the analysis remains equal to or greater than our frosting condition temperature. Sizing the TPS to meet this constraint would guarantee that the design would be optimized, because it would be the minimum thickness required to properly insulate the surface of the TPS from the cryogenic fuel tank.

The ADS optimizer used in TCAT's aeroheating analysis is a "general problem optimizer", meaning that it was created to handle many different types of engineering design problems. ADS has many controlling parameters that have to be fine-tuned in
order to achieve an optimal answer. Therefore, a problem specific optimizer was written into the main FORTRAN code to find the optimum design for the analysis. The Newton-Raphson method was implemented to find the root of the function $F$, where $F$ is given by equation (10).

$$ F = T_i^N - \text{temp}mrt(1) $$

In this equation, $T_i^N$ is the surface temperature of the TPS at the final time, and $\text{temp}mrt(1)$ is the frosting condition temperature. Since the temperature $T_i^N$ is a function of the thickness of the material being sized, $F$ is also a function of the same thickness. Therefore, to find the optimal design, we want to drive the function $F$ to zero.

The Newton-Raphson method is an iterative method used to find the root of a function. The Newton-Raphson formula is given in equation (11).

$$ x_{i+1} = x_i - \frac{F(x_i)}{F'(x_i)} $$

In this notation, $F$ is the same as in equation (10), and $x$ is the root. In this case, we do not know the value of $F'(x_i)$, so we can approximate it using a backward finite divided difference. This gives us equation (12).

$$ F'(x_i) \approx \frac{F(x_{i+1}) - F(x_i)}{x_{i+1} - x_i} $$

Finally, substitution into equation (11) results in equation (13), which was integrated into the main FORTRAN code.

$$ x_{i+1} = x_i - \frac{F(x_i)(x_{i+1} - x_i)}{F(x_{i+1}) - F(x_i)} $$
We see that two initial guesses for the root must first be made before any subsequent approximations for the root can be made. Therefore, in the main code, two guesses for the root are made before the main optimization loop begins. Once the difference in consecutive approximations falls below 0.0001, convergence is declared and the optimized thickness is obtained.

Program Execution

The ground hold analysis begins with the input of the ground hold time, time step for the analysis, ambient temperature, fuel temperature, frosting temperature, and TPS stackup configuration data from a web browser. Once these are input, a CGI script (‘single.cgi’ for a single analysis or ‘compare.cgi’ for a stackup comparison) is called. The CGI script takes this data and creates an input file called ‘inputs.in’ which contains the constraints entered by the user, and also TPS material properties based on the materials selected by the user and data taken from the TPSX materials database. The FORTRAN main code (‘ground.f’ if the first layer is optimized or ‘ground2.f’ if the second layer is optimized) is called by the CGI script where the optimization is performed. The main FORTRAN code calls the subroutines ‘one.f’, ‘two.f’, ‘three.f’, ‘four.f’, or ‘five.f’, depending on the number of materials selected in the TPS material configuration. These subroutines solve the one-dimensional heat equation using the numerical method described in the previous section. Results are sent back to the main FORTRAN code, where the Newton-Raphson method is implemented to converge upon the solution. Once the solution is obtained, the results are sent back to the CGI script. These results are then written to the files ‘results.txt’ and ‘thickness.txt’, which are posted back to the user via the web interface. A schematic of the program flow logic is shown in Figure 3.
Figure 3: Schematic of Program Flow Logic
3.0 Ground Hold Analysis Test Results

To illustrate the necessity for sizing TPS during both ground hold and in-flight operations, TPS materials were sized for a 10° half-angle cone. This 10° half-angle cone could be representative of the nosecap of a reusable launch vehicle. Figure 4 illustrates the geometry of the cone used to build the MINIVER input file.

![10° Half-Angle Cone](image)

**Figure 4: Schematic of 10° Half-Angle Cone**

The STS-1 reentry trajectory was used as the flight profile experienced by the cone in the MINIVER input file. The STS-1 reentry trajectory in tabular form can be found in Appendix A. Body points were placed two feet apart on the surface of the cone, resulting in 30 body points for both the leeward and windward sides. The TPS materials used for the analysis were CFBI, Composite Flexible Blanket Insulation, for the leeward surface and LI-2200, 22 lbm/ft³ Rigid Ceramic Tile Insulation, for the windward surface. Schematics of the TPS material configurations are shown in Figures 5 and 6.

<table>
<thead>
<tr>
<th>CFBI Blankets</th>
<th>LI-2200 Tiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTV Adhesive</td>
<td>RTV Adhesive</td>
</tr>
<tr>
<td>Graphite Epoxy Structure</td>
<td>Strain Isolator Pad</td>
</tr>
</tbody>
</table>

**Figure 5: Schematic of CFBI TPS Configuration**

<table>
<thead>
<tr>
<th>LI-2200 Tiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTV Adhesive</td>
</tr>
<tr>
<td>Strain Isolator Pad</td>
</tr>
<tr>
<td>RTV Adhesive</td>
</tr>
<tr>
<td>Graphite Epoxy Structure</td>
</tr>
</tbody>
</table>

**Figure 6: Schematic of LI-2200 TPS Configuration**
The results for the aeroheating analysis are given in Tables 1 – 3.

<table>
<thead>
<tr>
<th>Body Point</th>
<th>Thickness [in.]</th>
</tr>
</thead>
<tbody>
<tr>
<td>pointw1</td>
<td>1.89</td>
</tr>
<tr>
<td>pointw2</td>
<td>1.876</td>
</tr>
<tr>
<td>pointw3</td>
<td>1.817</td>
</tr>
<tr>
<td>pointw4</td>
<td>1.778</td>
</tr>
<tr>
<td>pointw5</td>
<td>1.75</td>
</tr>
<tr>
<td>pointw6</td>
<td>1.732</td>
</tr>
<tr>
<td>pointw7</td>
<td>1.721</td>
</tr>
<tr>
<td>pointw8</td>
<td>1.713</td>
</tr>
<tr>
<td>pointw9</td>
<td>1.706</td>
</tr>
<tr>
<td>pointw10</td>
<td>1.7</td>
</tr>
<tr>
<td>pointw11</td>
<td>1.694</td>
</tr>
<tr>
<td>pointw12</td>
<td>1.688</td>
</tr>
<tr>
<td>pointw13</td>
<td>1.683</td>
</tr>
<tr>
<td>pointw14</td>
<td>1.677</td>
</tr>
<tr>
<td>pointw15</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw16</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw17</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw18</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw19</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw20</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw21</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw22</td>
<td>1.512</td>
</tr>
<tr>
<td>pointw23</td>
<td>1.646</td>
</tr>
<tr>
<td>pointw24</td>
<td>1.644</td>
</tr>
<tr>
<td>pointw25</td>
<td>1.642</td>
</tr>
<tr>
<td>pointw26</td>
<td>1.639</td>
</tr>
<tr>
<td>pointw27</td>
<td>1.637</td>
</tr>
<tr>
<td>pointw28</td>
<td>1.635</td>
</tr>
<tr>
<td>pointw29</td>
<td>1.633</td>
</tr>
<tr>
<td>pointw30</td>
<td>1.631</td>
</tr>
</tbody>
</table>

Table 1: TPS Thicknesses for Windward Surface of 10° Half-Angle Cone for In-Flight Aeroheating with LI-2200 Tiles

<table>
<thead>
<tr>
<th>Body Point</th>
<th>Thickness [in.]</th>
</tr>
</thead>
<tbody>
<tr>
<td>point1</td>
<td>1.427</td>
</tr>
<tr>
<td>point2</td>
<td>1.007</td>
</tr>
<tr>
<td>point3</td>
<td>0.462</td>
</tr>
<tr>
<td>point4</td>
<td>0.25</td>
</tr>
<tr>
<td>point5</td>
<td>0.25</td>
</tr>
<tr>
<td>point6</td>
<td>0.25</td>
</tr>
<tr>
<td>point7</td>
<td>0.25</td>
</tr>
<tr>
<td>point8</td>
<td>0.25</td>
</tr>
<tr>
<td>point9</td>
<td>0.25</td>
</tr>
<tr>
<td>point10</td>
<td>0.25</td>
</tr>
<tr>
<td>point11</td>
<td>0.25</td>
</tr>
<tr>
<td>point12</td>
<td>0.25</td>
</tr>
<tr>
<td>point13</td>
<td>0.25</td>
</tr>
<tr>
<td>point14</td>
<td>0.25</td>
</tr>
<tr>
<td>point15</td>
<td>0.25</td>
</tr>
<tr>
<td>point16</td>
<td>0.25</td>
</tr>
<tr>
<td>point17</td>
<td>0.25</td>
</tr>
<tr>
<td>point18</td>
<td>0.25</td>
</tr>
<tr>
<td>point19</td>
<td>0.25</td>
</tr>
<tr>
<td>point20</td>
<td>0.25</td>
</tr>
<tr>
<td>point21</td>
<td>0.25</td>
</tr>
<tr>
<td>point22</td>
<td>0.25</td>
</tr>
<tr>
<td>point23</td>
<td>0.25</td>
</tr>
<tr>
<td>point24</td>
<td>0.25</td>
</tr>
<tr>
<td>point25</td>
<td>0.25</td>
</tr>
<tr>
<td>point26</td>
<td>0.25</td>
</tr>
<tr>
<td>point27</td>
<td>0.25</td>
</tr>
<tr>
<td>point28</td>
<td>0.25</td>
</tr>
<tr>
<td>point29</td>
<td>0.25</td>
</tr>
<tr>
<td>point30</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 2: TPS Thicknesses for Leeward Surface of 10° Half-Angle Cone for In-Flight Aeroheating with CFBI Blankets
The analysis was then performed using the ground hold analysis tool. For each surface, the TPS was sized for the presence of both a liquid hydrogen (LH2) fuel tank and a liquid oxygen (LOx) fuel tank located inside the nosecap. The parameters of the analysis are given in Table 4 and the results of this analysis are given in Tables 5 and 6.

Table 4: Ground Hold Analysis Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Hold Time</td>
<td>7800 s (2 hours)</td>
</tr>
<tr>
<td>Time Step for Analysis</td>
<td>10 s</td>
</tr>
<tr>
<td>Ambient Temperature</td>
<td>70 °F</td>
</tr>
<tr>
<td>LOx Fuel Temperature</td>
<td>-297.67 °F</td>
</tr>
<tr>
<td>LH2 Fuel Temperature</td>
<td>-423.67 °F</td>
</tr>
<tr>
<td>Frosting Condition Temperature</td>
<td>40 °F</td>
</tr>
</tbody>
</table>

Table 5: TPS Sizing Results for Ground Hold Analysis Above a LOx Fuel Tank

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS Thickness [in.]</td>
<td>1.438</td>
</tr>
<tr>
<td>TPS Unit Weight [lb/ft²]</td>
<td>2.6359</td>
</tr>
</tbody>
</table>

Table 6: TPS Sizing Results for Ground Hold Analysis Above a LH2 Fuel Tank

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS Thickness [in.]</td>
<td>1.796</td>
</tr>
<tr>
<td>TPS Unit Weight [lb/ft²]</td>
<td>3.2929</td>
</tr>
</tbody>
</table>
As a check to see if the results make physical sense, the temperature profiles for each TPS material configuration were graphed versus time. This will also give an indication whether or not the design constraints have been satisfied. Figures 7 and 8 show the nodal temperature profiles for the LH2 sized TPS materials, and Figures 9 and 10 show the nodal temperature profiles for the LOx sized TPS materials.

Figure 7: Nodal Temperature Histories for LI-2200 Sized Over LH2 Tank

Figure 8: Nodal Temperature Histories for CFBI Sized Over LH2 Tank
Figure 9: Nodal Temperature Histories for L1-2200 Sized Over LOx Tank

Figure 10: Nodal Temperature Histories for CFBI Sized Over LOx Tank
In each of these nodal temperature histories, we see that the temperature of the surface never drops below 277.6 K (40 °F), which is the surface frosting constraint. Also, the temperature of the backface remains constant and equal to the fuel temperature specified for each case. Therefore, we can be confident that problems constraints have been met. Also, the temperature profiles are all approaching a steady state value at the end of the analysis. The profiles all increase in temperature through the depth of the stack, starting with the lowest temperature at the backface and the highest temperature at the surface. There is also no overlap or crossing of temperature profiles during the analysis. This provides assurance that the numerical method is accurately modeling the physical heat transfer mechanism.

We see that for the windward surface, the unit weight of the LOx sized TPS is lower than the unit weight obtained from the aeroheating analysis. However, the unit weight of the LH2 sized TPS is higher than that of the aeroheating analysis. This means that the TPS sized for aeroheating is sufficiently thick to prevent surface frosting if a LOx tank is located in the nosecap, but it is not thick enough if a LH2 is located in the nosecap.

Now, examining the results of the leeward surface, we see that the unit weights of both the LOx sized and LH2 sized TPS are higher than that of the aeroheating results. If we consider the thicknesses of the TPS alone, we can see that the TPS sized for aeroheating at the very first body point of the leeward side is thick enough to insulate the TPS from the effects of a LOx tank, but still is not thick enough for the LH2 tank. This means that although the TPS may be able to sustain the heat loads experienced by the in-flight aeroheating, the TPS still is not adequately sized for a complete mission profile. If the TPS results from the aeroheating analysis alone were used to construct the vehicle, then the majority of the surface of the vehicle would experience surface frosting. This would result in adverse operating conditions, which could result in TPS damage or possible mission failure.

The increase in unit weights calculated from the ground hold analysis will of course affect the overall dry weight of the vehicle. This could possibly result in a significant increase in cost. Although cost is an important design driver, performance must not be neglected. Therefore, in order to provide the most accurate results and meet
all performance expectations, the TPS should be sized for both ground hold and in-flight aeroheating, with the TPS thicknesses and unit weights satisfying both sets of constraints selected for the final design.
4.0 Crank-Nicolson Method and TCAT

The simple implicit method used in TCAT’s aeroheating analysis has been replaced with the Crank-Nicolson finite difference method in order to increase numerical accuracy and to decrease computational time.

Finite Difference Equations

The one-dimensional heat equation discretized at the interior nodes, using the nodal map shown in Figure 11, with a simple implicit central finite difference scheme results in equation (14).

\[
T_i^{n+1} = -\frac{\alpha_i \Delta t}{\Delta x^2} \left( T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1} \right) + T_i^{n+1}  
\] (14)

![Figure 11: Nodal Mesh Map for Interior Nodes](image)

Using the same nodal mesh map, the discretization of the one-dimensional heat equation with the Crank-Nicolson method results in equation (15).

\[
\left(1 - \frac{\alpha_i \Delta t}{\Delta x^2}\right)T_i^n + \frac{\alpha_i \Delta t}{2\Delta x^2}\left(T_{i+1}^n + T_{i-1}^n\right) = -\frac{\alpha_i \Delta t}{2\Delta x^2}\left( T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1} \right) + T_i^{n+1}  
\] (15)

Rearranging this equation such that \( f \) is equal to the information at the next time step minus the information at the current time step, we arrive at equation (16).
This equation is now in the same form as the equations used in TCAT's solution process. The resulting system of equations is iteratively solved using the Newton-Raphson method. First, the Jacobian Matrix is formed.

\[ J_{i,j} = \frac{\partial f_i}{\partial T_j} \|_{i,j=1,\ldots,N} \]  

The system of equations is now in the form \( Ax = b \), given in equation (10), where \( A \) is the Jacobian matrix, \( x \) is the temperature change at time \( n+1 \) and \( b = -f \) at time \( n \).

\[
\begin{bmatrix}
\frac{\partial f_1}{\partial T_1^{n+1}} & \frac{\partial f_1}{\partial T_2^{n+1}} & 0 & 0 & 0 & 0 \\
\frac{\partial f_2}{\partial T_1^{n+1}} & \frac{\partial f_2}{\partial T_2^{n+1}} & \frac{\partial f_2}{\partial T_3^{n+1}} & 0 & 0 & 0 \\
0 & \frac{\partial f_3}{\partial T_2^{n+1}} & \frac{\partial f_3}{\partial T_3^{n+1}} & \frac{\partial f_3}{\partial T_4^{n+1}} & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & 0 \\
0 & 0 & 0 & \frac{\partial f_{N-1}}{\partial T_{N-1}^{n+1}} & \frac{\partial f_{N-1}}{\partial T_{N-2}^{n+1}} & \frac{\partial f_{N-1}}{\partial T_{N-1}^{n+1}} \\
0 & 0 & 0 & 0 & \frac{\partial f_N}{\partial T_N^{n+1}} & \frac{\partial f_N}{\partial T_{N-1}^{n+1}} \\
\end{bmatrix}
\begin{bmatrix}
\Delta T_1^{n+1} \\
\Delta T_2^{n+1} \\
\Delta T_3^{n+1} \\
\vdots \\
\Delta T_{N-1}^{n+1} \\
\Delta T_N^{n+1} \\
\end{bmatrix}
= 
\begin{bmatrix}
-f_1 \\
-f_2 \\
-f_3 \\
\vdots \\
-f_{N-1} \\
-f_N \\
\end{bmatrix}
\]  

Solution of this system is performed by making an initial guess for the temperature at the next time level \( n+1 \) and iteratively solving for the residual \( \Delta T^{n+1} \) using the Thomas Algorithm until convergence is achieved.
5.0 World Wide Web Interface

The web version of TCAT has been completely restructured in order to provide a more intuitive user interface. The greeting page of TCAT welcomes the user with an animation in Macromedia Flash format. Following the animation, the user can select the 'TCAT' link that appears in the center of the screen to go to the main content page. It is here that the user can select which version of TCAT to use for the analysis.

Figure 12: Screen Shot of TCAT Welcome Page
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

Ground Hold Analysis

If the groundhold analysis is selected, a menu appears in the content frame of the page where the user can choose to analyze a single TPS configuration, or compare three TPS configurations. If the user selects the single configuration analysis, the single analysis page appears in the content window.

![Figure 13: Screen Shot of TCAT Ground Hold Single Analysis Page](image)

The user must input the duration of the groundhold, the time step for the analysis, the ambient temperature of the air during groundhold, the fuel temperature for the isothermal boundary condition, and the surface frosting condition temperature. The user must also build the TPS configuration using the pull-down menus on the right side of the content window. While building the TPS configuration, the user can select the number of materials to use in the configuration. There is also the option to optimize either the first
or the second layer of the TPS configuration. This option is useful because some TPS configurations, such as RCC, SiC, or TUF1 are comprised of a thin composite surface layer and a diffusion layer. The surface layer remains the same thickness; however, the diffusion layer is sized for the analysis.

If the user selects the multiple configuration comparison, the multiple analysis page appears in the content window.

![Figure 14: Screen Shot of TCAT Ground Hold Multiple Analysis Page](image)

Here, the user can build up to three different TPS material configurations. This option works in the same way as the single stackup option, the only difference being that there are now three sets of pull down menus in which the user can build their own material stackups. The results for each analysis are stored in different files, which are compared to determine which material stackup option provides the minimum material unit weight for the analysis. The results are then returned to the user in a new window.
Aeroheating Analysis

The web pages for the original TCAT aeroheating program have been integrated into the new TCAT website. This means that both the aeroheating analysis and ground hold analysis can be performed at the same location. The original functionality of TCAT's aeroheating analysis has remained unchanged. A screen shot of TCAT's aeroheating analysis configuration is shown in Figure 15.

![TCAT Aeroheating Analysis](image)

Figure 15: Screen Shot of TCAT Aeroheating Analysis Page

MINIVER File Upload

There is now an option available for the user to upload a MINIVER file from their PC to the Space Systems Design Lab's web server. This option was made available because in order to perform aeroheating analyses using TCAT, a MINIVER file is required. To access the file upload interface, the user can simply click on the "File
Upload, link in the main menu, or go to the aeroheating analysis page where a link to the file upload page is also available. When the file upload screen appears, the user can then click on the browse button to locate a MINIVER file from their PC's hard drive. The user can specify up to three files to upload at once, with a maximum file size of 1 MB. Once the upload button is pressed, the file is automatically uploaded to the SSDL web server where the aeroheating analysis files are located. The CGI script to perform the file upload was taken from the "Perl Services," website, which is located at http://www.perlservices.net/en/programs/psupload/index.shtml. The html files associated with the ‘upload.cgi’ script have been modified to conform to TCAT’s site layout.

![Screen Shot of TCAT MINIVER File Upload Page](image)

*Figure 16: Screen Shot of TCAT MINIVER File Upload Page*
6.0 Conclusions

The capabilities of TCAT at its inception, although accurate at calculating TPS material thicknesses when compared to commercial heating codes such as SINDA, did not provide complete results that are required for thermal protection system sizing analyses. TCAT was capable of sizing TPS materials for in-flight aeroheating; however, the capability to size TPS for ground hold operations was still missing. In order to meet this goal, TCAT’s capabilities were augmented with the addition of a ground hold analysis tool. This ground hold analysis tool can size the TPS materials for body points located above cryogenic fuel tanks using an isothermal boundary condition at the backface of the fuel tank structure and a constant heat transfer rate from the atmosphere to the surface of the TPS. The results of this analysis tool can be compared to the results given by the aeroheating analysis to determine the TPS thicknesses required at each body point location to meet both the aeroheating and ground hold temperature constraints.

In our test case, we discovered that in some instances the TPS sized for aeroheating may be sufficiently thick to prevent surface frosting. This was evidenced in the case of the TPS sized for the windward side of the 10° half-angle cone with a LOx tank located inside the cone. However, for the rest of the analyses it was determined that the TPS sized for aeroheating were not sufficient to meet the constraints of the ground hold. Therefore, the design tool created specifically for the sizing of TPS during ground hold should be integrated into the TPS design process.

Even with the ground hold analysis tool developed, TCAT’s capabilities are not yet entirely developed. A method of integrating the ground hold and aeroheating analysis should be developed so that the TPS thicknesses of each individual analysis do not need to be compared on a point-by-point basis. This could be achieved by integrating the ground hold temperature and heat transfer profiles for each body point into the MINIVER file that includes the radiation equilibrium temperature and heat transfer profiles during in-flight trajectory. The entire analysis could be performed using this single input file, with the method of analysis switching from ground hold constraints to in-flight aeroheating constraints, returning the optimal TPS thickness that satisfies both sets of constraints to the user.
## Appendix A: STS-1 Reentry Trajectory

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Altitude (ft)</th>
<th>Velocity (ft/s)</th>
<th>Angle of Attack</th>
<th>Bank Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.3</td>
<td>373800</td>
<td>24590</td>
<td>41.26</td>
<td>0</td>
</tr>
<tr>
<td>90.3</td>
<td>351500</td>
<td>24620</td>
<td>41.21</td>
<td>0</td>
</tr>
<tr>
<td>135.3</td>
<td>329500</td>
<td>24640</td>
<td>40.46</td>
<td>0</td>
</tr>
<tr>
<td>180.3</td>
<td>308000</td>
<td>24660</td>
<td>40.65</td>
<td>0</td>
</tr>
<tr>
<td>225.3</td>
<td>288000</td>
<td>24660</td>
<td>41.9</td>
<td>0</td>
</tr>
<tr>
<td>270.3</td>
<td>269700</td>
<td>24610</td>
<td>39.39</td>
<td>0</td>
</tr>
<tr>
<td>315.3</td>
<td>256300</td>
<td>24470</td>
<td>40.71</td>
<td>0</td>
</tr>
<tr>
<td>360.3</td>
<td>250200</td>
<td>24220</td>
<td>41.74</td>
<td>0</td>
</tr>
<tr>
<td>405.3</td>
<td>247000</td>
<td>23920</td>
<td>39.95</td>
<td>0</td>
</tr>
<tr>
<td>450.3</td>
<td>244600</td>
<td>23610</td>
<td>39.28</td>
<td>0</td>
</tr>
<tr>
<td>495.3</td>
<td>242500</td>
<td>23280</td>
<td>39.66</td>
<td>0</td>
</tr>
<tr>
<td>545.3</td>
<td>240000</td>
<td>22880</td>
<td>39.07</td>
<td>0</td>
</tr>
<tr>
<td>569.3</td>
<td>238800</td>
<td>22680</td>
<td>39.32</td>
<td>0</td>
</tr>
<tr>
<td>593.3</td>
<td>237700</td>
<td>22470</td>
<td>39.02</td>
<td>0</td>
</tr>
<tr>
<td>617.3</td>
<td>236600</td>
<td>22250</td>
<td>39.25</td>
<td>0</td>
</tr>
<tr>
<td>641.3</td>
<td>235300</td>
<td>22020</td>
<td>39.48</td>
<td>0</td>
</tr>
<tr>
<td>665.3</td>
<td>233700</td>
<td>21780</td>
<td>39.97</td>
<td>0</td>
</tr>
<tr>
<td>689.3</td>
<td>232000</td>
<td>21530</td>
<td>40.01</td>
<td>0</td>
</tr>
<tr>
<td>713.3</td>
<td>232300</td>
<td>21260</td>
<td>40.25</td>
<td>0</td>
</tr>
<tr>
<td>737.3</td>
<td>230300</td>
<td>20980</td>
<td>40.46</td>
<td>0</td>
</tr>
<tr>
<td>761.3</td>
<td>227900</td>
<td>20680</td>
<td>40.14</td>
<td>0</td>
</tr>
<tr>
<td>785.3</td>
<td>225200</td>
<td>20360</td>
<td>40.16</td>
<td>0</td>
</tr>
<tr>
<td>809.3</td>
<td>223000</td>
<td>20020</td>
<td>40.16</td>
<td>0</td>
</tr>
<tr>
<td>833.3</td>
<td>219000</td>
<td>19650</td>
<td>40.27</td>
<td>0</td>
</tr>
<tr>
<td>857.3</td>
<td>215300</td>
<td>19240</td>
<td>40.4</td>
<td>0</td>
</tr>
<tr>
<td>881.3</td>
<td>211400</td>
<td>18790</td>
<td>40.12</td>
<td>0</td>
</tr>
<tr>
<td>905.3</td>
<td>207600</td>
<td>18300</td>
<td>40.3</td>
<td>0</td>
</tr>
<tr>
<td>929.3</td>
<td>205600</td>
<td>17760</td>
<td>40.01</td>
<td>0</td>
</tr>
<tr>
<td>953.3</td>
<td>202500</td>
<td>17180</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>977.3</td>
<td>197200</td>
<td>16540</td>
<td>40.9</td>
<td>0</td>
</tr>
<tr>
<td>1001</td>
<td>192200</td>
<td>15870</td>
<td>40.75</td>
<td>0</td>
</tr>
<tr>
<td>1025</td>
<td>187200</td>
<td>15130</td>
<td>39.87</td>
<td>0</td>
</tr>
<tr>
<td>1049</td>
<td>182600</td>
<td>14350</td>
<td>38.43</td>
<td>0</td>
</tr>
<tr>
<td>1073</td>
<td>179600</td>
<td>13570</td>
<td>39.53</td>
<td>0</td>
</tr>
<tr>
<td>1097</td>
<td>176600</td>
<td>12790</td>
<td>39.75</td>
<td>0</td>
</tr>
<tr>
<td>1121</td>
<td>172500</td>
<td>12010</td>
<td>38.96</td>
<td>0</td>
</tr>
<tr>
<td>1145</td>
<td>167500</td>
<td>11220</td>
<td>38.12</td>
<td>0</td>
</tr>
<tr>
<td>1169</td>
<td>161900</td>
<td>10440</td>
<td>36.97</td>
<td>0</td>
</tr>
<tr>
<td>1240</td>
<td>150000</td>
<td>8336</td>
<td>34.07</td>
<td>0</td>
</tr>
<tr>
<td>1302</td>
<td>133900</td>
<td>6757</td>
<td>28.2</td>
<td>0</td>
</tr>
<tr>
<td>1364</td>
<td>117400</td>
<td>5342</td>
<td>23.05</td>
<td>0</td>
</tr>
<tr>
<td>1426</td>
<td>106200</td>
<td>4064</td>
<td>20.27</td>
<td>0</td>
</tr>
<tr>
<td>1488</td>
<td>89100</td>
<td>2917</td>
<td>16.47</td>
<td>0</td>
</tr>
<tr>
<td>1550</td>
<td>76350</td>
<td>1915</td>
<td>10.86</td>
<td>0</td>
</tr>
<tr>
<td>1612</td>
<td>57440</td>
<td>1151</td>
<td>7.79</td>
<td>0</td>
</tr>
<tr>
<td>1674</td>
<td>41670</td>
<td>800</td>
<td>7.72</td>
<td>0</td>
</tr>
<tr>
<td>1736</td>
<td>27600</td>
<td>681</td>
<td>7.09</td>
<td>0</td>
</tr>
<tr>
<td>1860</td>
<td>3337</td>
<td>512</td>
<td>3.78</td>
<td>0</td>
</tr>
<tr>
<td>1925</td>
<td>-3</td>
<td>202</td>
<td>-12.2</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix B: Windward Aeroheating Analysis Input File

10           /* time step
48           /* number of nodes of material 1
12           /* number of nodes of material 2
12           /* number of nodes of material 3
12           /* number of nodes of material 4
12           /* number of nodes of material 5
0.899        /* emissivity of material 1
0.074420     /* thermal conductivity of material 1
0.396700     /* thermal conductivity of material 2
0.042000     /* thermal conductivity of material 3
0.396700     /* thermal conductivity of material 4
0.601300     /* thermal conductivity of material 5
352.400000   /* density of material 1
1410.000000  /* density of material 2
86.500600    /* density of material 3
1410.000000  /* density of material 4
1576.232900  /* density of material 5
0.628000     /* specific heat of material 1
1.168000     /* specific heat of material 2
1.318000     /* specific heat of material 3
1.168000     /* specific heat of material 4
0.833300     /* specific heat of material 5
294.2611111  /* initial temp (deg K)
0.075000     /* TPS thickness of material 1
0.003000     /* TPS thickness of material 2
0.004000     /* TPS thickness of material 3
0.003000     /* TPS thickness of material 4
0.003000     /* TPS thickness of material 5
277.5944444  /* Temperature limit of material 1
1644         /* Temperature limit of material 1
561.100000   /* Temperature limit of material 2
561.100000   /* Temperature limit of material 2
505.222000   /* Temperature limit of material 3
505.222000   /* Temperature limit of material 3
561.100000   /* Temperature limit of material 4
561.100000   /* Temperature limit of material 4
588.889      /* Temperature limit of material 5
19.99999999  /* backface temperature limit
6            /* istrat parameter
5            /* iopt parameter
5            /* ioned parameter
1340         /* iprint parameter
Appendix C: Leeward Aeroheating Analysis Input File

3          \!/number of materials
10.         \!/time step
40          \!/number of nodes in material 1
10          \!/number of nodes in material 2
10          \!/number of nodes in material 3
0.8856      \!/emissivity of material 1
0.03821     \!/thermal conductivity of material 1
0.3967      \!/thermal conductivity of material 2
0.6013      \!/thermal conductivity of material 3
96.118      \!/density of material 1
1410        \!/density of material 2
1576.2329   \!/density of material 3
0.7821      \!/specific heat of material 1
1.168       \!/specific heat of material 2
0.8333      \!/specific heat of material 3
300.        \!/initial temp (deg K)
0.05        \!/TPS thickness of material 1 (meters)
0.003       \!/TPS thickness of material 2 (meters)
0.003       \!/TPS thickness of material 3 (meters)
1477.44     \!/Temperature limit of material 1
1477.44     \!/Temperature limit of material 1
561.1       \!/Temperature limit of material 2
561.1       \!/Temperature limit of material 2
588.889     \!/Temperature limit of material 3
421.889     \!/backface temperature limit
6           \!/istrat parameter
5           \!/iopt parameter
5           \!/ioned parameter
0000        \!/iprint parameter
Appendix D: Windward Ground Hold Analysis for LOx Tank Input File

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.074420</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.396700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.042000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.396700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.601300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>352.400000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1410.000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86.500600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1410.000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1576.232900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.168000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.318000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.168000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.833300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>294.261111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.075000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.003000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.004000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.003000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.003000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>277.594444</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1644</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>561.100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>561.100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>505.222000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>505.222000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>561.100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>561.100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>588.889</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1340</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix E: Leeward Ground Hold Analysis for LOx Tank Input File

3  !/number of materials
10  !/time step
40  !/number of nodes of material 1
10  !/number of nodes of material 2
10  !/number of nodes of material 3
0.8856  !/emissivity of material 1
0.038210  !/thermal conductivity of material 1
0.396700  !/thermal conductivity of material 2
0.601300  !/thermal conductivity of material 3
96.118000  !/density of material 1
1410.000000  !/density of material 2
1576.232900  !/density of material 3
0.782100  !/specific heat of material 1
1.168000  !/specific heat of material 2
0.833300  !/specific heat of material 3
294.2611111  !/initial temp (deg K)
0.050000  !/TPS thickness of material 1
0.003000  !/TPS thickness of material 2
0.003000  !/TPS thickness of material 3
277.5944444  !/Temperature limit of material 1
1477.44  !/Temperature limit of material 1
561.100000  !/Temperature limit of material 2
588.889  !/Temperature limit of material 3
90  !/backface temperature limit
6  !/istrat parameter
5  !/LOPT parameter
5  !/ioned parameter
1340  !/iprint parameter
Appendix F: Windward Ground Hold Analysis for LH2 Tank Input File

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>!/number of materials</td>
</tr>
<tr>
<td>10</td>
<td>!/time step</td>
</tr>
<tr>
<td>48</td>
<td>!/number of nodes of material 1</td>
</tr>
<tr>
<td>12</td>
<td>!/number of nodes of material 2</td>
</tr>
<tr>
<td>12</td>
<td>!/number of nodes of material 3</td>
</tr>
<tr>
<td>12</td>
<td>!/number of nodes of material 4</td>
</tr>
<tr>
<td>12</td>
<td>!/number of nodes of material 5</td>
</tr>
<tr>
<td>0.899</td>
<td>!/emissivity of material 1</td>
</tr>
<tr>
<td>0.074420</td>
<td>!/thermal conductivity of material 1</td>
</tr>
<tr>
<td>0.396700</td>
<td>!/thermal conductivity of material 2</td>
</tr>
<tr>
<td>0.042000</td>
<td>!/thermal conductivity of material 3</td>
</tr>
<tr>
<td>0.396700</td>
<td>!/thermal conductivity of material 4</td>
</tr>
<tr>
<td>0.601300</td>
<td>!/thermal conductivity of material 5</td>
</tr>
<tr>
<td>352.400000</td>
<td>!/density of material 1</td>
</tr>
<tr>
<td>1410.000000</td>
<td>!/density of material 2</td>
</tr>
<tr>
<td>86.500600</td>
<td>!/density of material 3</td>
</tr>
<tr>
<td>1410.000000</td>
<td>!/density of material 4</td>
</tr>
<tr>
<td>1576.232900</td>
<td>!/density of material 5</td>
</tr>
<tr>
<td>0.628000</td>
<td>!/specific heat of material 1</td>
</tr>
<tr>
<td>1.168000</td>
<td>!/specific heat of material 2</td>
</tr>
<tr>
<td>1.318000</td>
<td>!/specific heat of material 3</td>
</tr>
<tr>
<td>1.168000</td>
<td>!/specific heat of material 4</td>
</tr>
<tr>
<td>0.833300</td>
<td>!/specific heat of material 5</td>
</tr>
<tr>
<td>294.2611111</td>
<td>!/initial temp (deg K)</td>
</tr>
<tr>
<td>0.075000</td>
<td>!/TPS thickness of material 1</td>
</tr>
<tr>
<td>0.003000</td>
<td>!/TPS thickness of material 2</td>
</tr>
<tr>
<td>0.004000</td>
<td>!/TPS thickness of material 3</td>
</tr>
<tr>
<td>0.003000</td>
<td>!/TPS thickness of material 4</td>
</tr>
<tr>
<td>0.003000</td>
<td>!/TPS thickness of material 5</td>
</tr>
<tr>
<td>277.5944444</td>
<td>!/Temperature limit of material 1</td>
</tr>
<tr>
<td>1664</td>
<td>!/Temperature limit of material 1</td>
</tr>
<tr>
<td>561.100000</td>
<td>!/Temperature limit of material 2</td>
</tr>
<tr>
<td>561.100000</td>
<td>!/Temperature limit of material 2</td>
</tr>
<tr>
<td>505.222000</td>
<td>!/Temperature limit of material 3</td>
</tr>
<tr>
<td>505.222000</td>
<td>!/Temperature limit of material 3</td>
</tr>
<tr>
<td>561.100000</td>
<td>!/Temperature limit of material 4</td>
</tr>
<tr>
<td>561.100000</td>
<td>!/Temperature limit of material 4</td>
</tr>
<tr>
<td>588.889</td>
<td>!/Temperature limit of material 5</td>
</tr>
<tr>
<td>19.99999999</td>
<td>!/backface temperature limit</td>
</tr>
<tr>
<td>6</td>
<td>!/istrat parameter</td>
</tr>
<tr>
<td>5</td>
<td>!/iopt parameter</td>
</tr>
<tr>
<td>5</td>
<td>!/ioned parameter</td>
</tr>
<tr>
<td>1340</td>
<td>!/iprint parameter</td>
</tr>
</tbody>
</table>
Appendix G: Leeward Ground Hold Analysis for LH2 Tank Input File

3  !/number of materials
10  !/time step
40  !/number of nodes of material 1
10  !/number of nodes of material 2
10  !/number of nodes of material 3
0.8856  !/emissivity of material 1
0.038210  !/thermal conductivity of material 1
0.396700  !/thermal conductivity of material 2
0.601300  !/thermal conductivity of material 3
96.118000  !/density of material 1
1410.000000  !/density of material 2
1576.232900  !/density of material 3
0.782100  !/specific heat of material 1
1.168000  !/specific heat of material 2
0.833300  !/specific heat of material 3
294.2611111  !/initial temp (deg K)
0.050000  !/TPS thickness of material 1
0.003000  !/TPS thickness of material 2
0.003000  !/TPS thickness of material 3
277.5944444  !/Temperature limit of material 1
1477.44  !/Temperature limit of material 1
561.100000  !/Temperature limit of material 2
561.100000  !/Temperature limit of material 2
588.889  !/Temperature limit of material 3
19.99999999  !/backface temperature limit
6  !/istrat parameter
5  !/iopt parameter
5  !/ioned parameter
1340  !/iprint parameter
Appendix H: 'single.cgi' Ground Hold Single Analysis CGI Script

#!/usr/local/bin/perl
read(STDIN, $buffer, $ENV{'CONTENT-LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+/-/; # tr/+/-/ subr was used after split to eliminate the +
    $value =~ s/[a-fA-F0-9]/pack("C", hex($1))/eg;
    $value =~ s/-!/-!/g;
    $FORM($name) = $value;
}

$matnum = $FORM(matnum);
$method = $FORM(method);
material_1 = $FORM(material_1);
material_2 = $FORM(material_2);
material_3 = $FORM(material_3);
material_4 = $FORM(material_4);
material_5 = $FORM(material_5);
holdtime = $FORM(holdtime);
timestep = $FORM(timestep);
ambient = $FORM(ambient);
fueltemp = $FORM(fueltemp);
frost = $FORM(frost);

# obtain inputs from command line
#
# List of variables
# $bodyarea - surface area of the body in square feet
# $bodytile - fraction of body area covered with tiles
# $bodyblanket - fraction of body area covered with blankets
# $filename - name of MINIVER 1.0 s file
# $sharp - indicates the use of SHARP materials (yes or no)
# $bodyside - vehicle side analysis is conducted on (leeward or windward)
# $material - type of material to be used along with backface material
#
#
# Prompt the user for inputs.
#
$holdtime = 7800;
$timestep = 10;
$ambient = 70;
$fueltemp = 423.67;
$frost = 40;

print "Enter the amount of time, in seconds, the vehicle will be in groundhold.\n";
$holdtime = readline(*STDIN);
chop($holdtime);

print "Enter the timestep, in seconds, for the analysis.\n";
$timestep = readline(*STDIN);
chop($timestep);

print "Enter the heat transfer coefficient [BTU/(ft2*hour*F)].\n";
$heatrate = readline(*STDIN);
chop($heatrate);
#print "Enter the ambient temperature [F] of the groundhold location.\n";  
#ambient = readline("STDIN");  
#chop(ambient);  
#print "Enter the temperature [F] of the cryogenic fuel (ie. the isothermal boundary condition).\n";  
#fueltemp = readline("STDIN");  
#chop(fueltemp);  
#print "Enter the frosting condition Temperature [F] of the surface TPS material.\n";  
#frost = readline("STDIN");  
#chop(frost);  

time = "Sholdtime !/Time of Groundhold";  
close(TIME);  

heatrate = 1;  

heattransfer = heatrate * ambient *3.15459074506;  

fueltemp = (fueltemp-32)*(5/9)+273.15;  

frost = (frost-32)*(5/9)+273.15;  

#material-option = readline("STDIN");  
#chop(material-option);  

matnum = readline("STDIN");  
#chop(matnum);  

if (material-option eq 1) {  
  for (loopnum=1; loopnum <= matnum; loopnum++) {  
    print "Please select the surface material of the TPS stackup.\n";  
    print "\n";  
  }  
}
Improvements in Thermal Protection Systems Sizing Capabilities for TCA

...
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

print "Please select fourth material of the TPS stackup.\n";
print "\n";
print "----------- Tiles ---------------\n";
print "TUFI_AETB0_Diffusion_Layer\n";
print "TUFI_AETB12_Diffusion_Layer\n";
print "FRCI2\n";
print "FRCI20\n";
print "LI900\n";
print "LI2200\n";
print "RCC\n";
print "SiC\n";
print "----------- Blankets ---------------\n";
print "AFRSI\n";
print "CFBI\n";
print "DURAFRSI\n";
print "PB\n";
print "TAB1\n";
print "---------- Intermediates \n";
print "Strain_Isolator_Pad\n";
print "RTV_Adhesive\n";
print "Rohacell_Foam\n";
print "---------- Structures \n";
print "GrEx_Structure\n";
print "Al_Structure\n";
print "TiAl_Structure\n";
print "\n";
print "Enter the material name as it appears above.\n";
$material_4 = readline("STDIN") ;
chop($material_4)
;

elsif (Sloopnum eq 5) {

print "Please select fifth material of the TPS stackup.\n";
print "\n";
print "----------- Tiles ---------------\n";
print "TUFI_AETB0_Diffusion_Layer\n";
print "TUFI_AETB12_Diffusion_Layer\n";
print "FRCI2\n";
print "FRCI20\n";
print "LI900\n";
print "LI2200\n";
print "RCC\n";
print "SiC\n";
print "----------- Blankets ---------------\n";
print "AFRSI\n";
print "CFBI\n";
print "DURAFRSI\n";
print "PB\n";
print "TAB1\n";
print "---------- Intermediates \n";
print "Strain_Isolator_Pad\n";
print "RTV_Adhesive\n";
print "Rohacell_Foam\n";
print "---------- Structures \n";
print "GrEx_Structure\n";
print "Al_Structure\n";
print "TiAl_Structure\n";
print "\n";
print "Enter the material name as it appears above.\n";
$material_5 = readline("STDIN") ;
chop($material_5)
;

# print "\n";
# print "Please enter the optimization method.\n" ;
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

# print "1. If you would like to optimize the surface material thickness, or\n";
# print "2. If you would like to optimize the second TPS material thickness.\n";
# print "\n";
# print "NOTE: If either of the laminate materials were selected as the\n";
# print "surface TPS material, the second optimization method will\n";
# print "be automatically selected.\n";
#method = readline("STDIN");
# print "\n";

$material_option=1;
$materialnum=5;
$material_1=AFRS12500;
$material_2=TUF1_AETB12_Diffusion_Layer;
$material_3=RTV_Adhesive;
$material_4=GrEx_Structure;
$material_5=GrEx_Structure;
$material_6=Strain_Isolator_Pad;
$material_7=Rohacell_Foam;
$material_8=GrEx_Structure;

if ($material_option eq 1) {
    for ($loopnum=1; $loopnum <= $materialnum; $loopnum++) {
        if ($loopnum eq 1) {
            open(DAT,"<$material_1";
            $i = 0;
            while(<DAT>) {
                $i = $i + 1;
                my($line) = <$i;
                chomp($line);
                if ($i eq 1) {
                    $nodes[1] = $line;
                } elsif ($i eq 2) {
                    $cond[1] = $line;
                } elsif ($i eq 3) {
                    $density[1] = $line;
                } elsif ($i eq 4) {
                    $specheat[1] = $line;
                } elsif ($i eq 5) {
                    $thick[1] = $line;
                } elsif ($i eq 6) {
                    $stemplimit[1]= $line;
                } elsif ($i eq 7) {
                    $emissivity=$line;
                }
            }
            close(DAT);
        } elsif ($loopnum eq 2) {
            open(DAT,"<$material_2";
            $i = 0;
            while(<DAT>) {
                $i = $i + 1;
                my($line) = <$i;
                chomp($line);
                if ($i eq 1) {
                    $nodes[2] = $line;
                } elsif ($i eq 2) {
                    $cond[2] = $line;
                } elsif ($i eq 3) {
                    $density[2] = $line;
                } elsif ($i eq 4) {
                    $specheat[2] = $line;
                } elsif ($i eq 5) {
                    $thick[2] = $line;
                } elsif ($i eq 6) {
                    $stemplimit[2]= $line;
                } elsif ($i eq 7) {
                    $emissivity=$line;
                }
            }
            close(DAT);
        }
    }
}
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```perl
$density[2] = $line;
elseif ($i eq 4) {
    $specheat[2] = $line;
    $thick[2] = $line;
    $stemlimit[2] = $line;
}
}
close(DAT);

elsif ($loopnum eq 3) {
    open(DAT, "<$material_3'');
    $i = 0;
    while(<DAT>) {
        $i = $i + 1;
        my($line) = $_.
        chomp($line);
        if ($i eq 1) {
            $nodes[3] = $line;
        } elsif ($i eq 2) {
            $cond[3] = $line;
        } elsif ($i eq 3) {
            $density[3] = $line;
        } elsif ($i eq 4) {
            $specheat[3] = $line;
        } elsif ($i eq 5) {
            $thick[3] = $line;
        } elsif ($i eq 6) {
            $stemlimit[3] = $line;
        }
    }
close(DAT);

elsif ($loopnum eq 4) {
    open(DAT, "<$material_4'');
    $i = 0;
    while(<DAT>) {
        $i = $i + 1;
        my($line) = $_.
        chomp($line);
        if ($i eq 1) {
            $nodes[4] = $line;
        } elsif ($i eq 2) {
            $cond[4] = $line;
        } elsif ($i eq 3) {
            $density[4] = $line;
        } elsif ($i eq 4) {
            $specheat[4] = $line;
        } elsif ($i eq 5) {
            $thick[4] = $line;
        } elsif ($i eq 6) {
            $stemlimit[4] = $line;
        }
    }
close(DAT);
```

Stephen James Izen
elsif ($loopnum eq 5) {
    open(DAT, "<material_5");
    $i = 0;
    while(<DAT>) {
        $i = $i + 1;
        my($line) = $i;
        chomp($line);
        if ($i eq 1) {
            $nodes[5] = $line;
        } elsif ($i eq 2) {
            $cond[5] = $line;
        } elsif ($i eq 3) {
            $density[5] = $line;
        } elsif ($i eq 4) {
            $specheat[5] = $line;
        } elsif ($i eq 5) {
            $thick[5] = $line;
        } elsif ($i eq 6) {
            $stemplimit[5] = $line;
        }
    }
    close(DAT);
}

# Create the TPS stackup file by reading the properties of each material selected and entering their values into a input file recognizable by T-CAT.

# system("touch stackup.in"); # create TPS stackup file
# system("chmod 777 stackup.in"); # change permissions of TPS stackup file
open(STACKUP, ">stackup.in");
print STACKUP "$matnum !/number of materials
";
print STACKUP "$timestep !/time step
";
for ($loopnum=1; $loopnum <= $matnum; $loopnum++) {
    printf STACKUP "%i !/number of nodes of material
",$nodes[$loopnum];
    print STACKUP "$emissivity !/emissivity of material 1
";
    for ($loopnum=1; $loopnum <= $matnum; $loopnum++) {
        printf STACKUP "%f !/thermal conductivity of material
",$cond[$loopnum];
    }
    for ($loopnum=1; $loopnum <= $matnum; $loopnum++) {
        printf STACKUP "%f !/density of material
",$density[$loopnum];
    }
    for ($loopnum=1; $loopnum <= $matnum; $loopnum++) {
        printf STACKUP "%f !/specific heat of material
",$specheat[$loopnum];
    }
    print STACKUP "$ambient !/initial temp (deg K)
";

Stephen James Ion
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

for ($loopnum=1; $loopnum <= $smatnum; $loopnum++) {
    printf STACKUP "\$thick[$loopnum]: !TPS thickness of material $loopnum
",
}

print "frost !/Temperature limit of material 1\n"
print "Stemplimit[1] !/Temperature limit of material 1\n"
for ($loopnum=2; $loopnum <= $smatnum-1; $loopnum++) {
    printf STACKUP "\$thick[$loopnum]: !TPS thickness of material $loopnum
",
    printf STACKUP "\$tlim[$loopnum]: !Temperature limit of material $loopnum
",
    printf STACKUP "Frost !/Temperature limit of material $loopnum
",
    printf STACKUP "Stemplimit[$loopnum]: !Temperature limit of material $loopnum\n",
}

print STACKUP "Stemplimit[$smatnum]: !Temperature limit of material $smatnum\n",
print STACKUP "$fueltemp: !/strat parameter\n",
print STACKUP "$iop: !/iopt parameter\n",
print STACKUP "$ion: !/ioned parameter\n",
print STACKUP "$iprint: !iprint parameter\n",
close(STACKUP);
# system("chmod 777 stackup.in");
}
else {
    print "Enter the material for TPS:\n",
    print "--------- Tiles ---------\n",
    print "--------- Blankets ---------\n",
    print "---------- Al\n",
    print "---------- GrEx\n",
    print "---------- SiC\n",
    print "---------- RCC\n",
    print "---------- TUFI\n",
    print "---------- FRC\n",
    print "---------- CFB\n",
    print "---------- PBI\n",
    print "---------- TAB\n",
    print "Enter the material name as it appears above.\n",
    $material = readline(*STDIN);
    chop($material);
}

####################################################
# Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```bash
$numtimes = $holdtime/$stimestep + 1;
$st_time[1] = 0;
for ($i=2; $i <= $numtimes; $i++) {
    $st_time[$i] = $st_time[$i-1] + $stimestep;
}

# create mock MINIVER file
# change permissions of file
open(MINI, '<l_stackup.s');
print MINI "Mock MINIVER file for input into ground hold analysis
1.0n";
for ($i=2; $i <= $numtimes; $i++) {
    print MINI "$st_time[$i] Ambient Heat transfer\n";
}
print MINI "-100.0";
close(MINI);
```

---

Stephen James Ion
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

printf NEWFILE "%f \n", Stradeq$1);
  } #end of for loop
close(NEWFILE);
$1=0;
goto START
} #end of if statement
} #end of while loop
} #end of until loop
close(FILE);

# incrementally move each body point file into "miniver.in" and execute tcat
# perl script that creates the filenames needed in order to batch execute tcat
# filenames are created via input from the user
# inputs (file_start_number, file_end_number, body_side)
# ..................................................................................

$start=1; #start of the file index
$end=($filecount-1)/2; #end of the file index
$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "\n";
close(FILE);

system("touch thickness.txt");
#system("chmod 777 thickness.txt");
open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS Ground hold design results\n\n";
# ..................................................................................

if ($material_option eq 1) {

$inputfile = "stackup.in";
}
else {
  if ($material eq "AETBB_A1") {
    $inputfile="AETB_B_AL_5inputs.in";
  }
  if ($material eq "AETBB_GrEx") {
    $inputfile="AETB_B_GrEx_5inputs.in";
  }
  if ($material eq "AETB12_A1") {
    $inputfile="AETB_12_AL_5inputs.in";
  }
  if ($material eq "AETB12_GrEx") {
    $inputfile="AETB_12_GrEx_5inputs.in";
  }
  if ($material eq "FRCI12_A1") {
    $inputfile="FRCI_12_AL_5inputs.in";
  }
  if ($material eq "FRCI12_GrEx") {
    $inputfile="FRCI_12_GrEx_5inputs.in";
  }
  if ($material eq "FRCI20_A1") {
    $inputfile="FRCI_20_AL_5inputs.in";
  }
  if ($material eq "FRCI20_GrEx") {
    $inputfile="FRCI_20_GrEx_5inputs.in";
  }
}
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```cpp
if ($material eq "Li900_Al") {
    $inputfile="Li_900_AL_5inputs.in";
}
if ($material eq "Li900_GrEx") {
    $inputfile="Li_900_GrEx_5inputs.in";
}
if ($material eq "Li2200_Al") {
    $inputfile="Li_2200_AL_5inputs.in";
}
if ($material eq "Li2200_GrEx") {
    $inputfile="Li_2200_GrEx_5inputs.in";
}
if ($material eq "RCC_GrEx") {
    $inputfile="RCC_GrEx_5inputs.in";
}
if ($material eq "RCC_Al") {
    $inputfile="RCC_AL_5inputs.in";
}
if ($material eq "SiC_GrEx") {
    $inputfile="SiC_GrEx_5inputs.in";
}
if ($material eq "SiC_Al") {
    $inputfile="SiC_AL_5inputs.in";
}
if ($material eq "TUF1_GrEx") {
    $inputfile="TUF1_GrEx_5inputs.in";
}
if ($material eq "AFRSI_Al") {
    $inputfile="AFRSI_AL_3inputs.in";
}
if ($material eq "AFRSI_GrEx") {
    $inputfile="AFRSI_GrEx_3inputs.in";
}
if ($material eq "CFBI_Al") {
    $inputfile="CFBI_AL_3inputs.in";
}
if ($material eq "CFBI_GrEx") {
    $inputfile="CFBI_GrEx_3inputs.in";
}
if ($material eq "DURAFRSI_Al") {
    $inputfile="DURAFRSI_AL_3inputs.in";
}
if ($material eq "DURAFRSI_GrEx") {
    $inputfile="DURAFRSI_GrEx_3inputs.in";
}
if ($material eq "PBI_Al") {
    $inputfile="PBI_AL_3inputs.in";
}
if ($material eq "PBI_GrEx") {
    $inputfile="PBI_GrEx_3inputs.in";
}
if ($material eq "TABI_Al") {
    $inputfile="TABI_AL_3inputs.in";
}
if ($material eq "TABI_GrEx") {
    $inputfile="TABI_GrEx_3inputs.in";
}

# print "input file = $inputfile\n";
# ---------------------------------------------------
# Execute TCAT and send ADS output into file named "junkoutput"
# ---------------------------------------------------

system("touch inputs.in");
system("chmod 777 inputs.in");
system("cp $inputfile inputs.in");
if ($material_option eq 1) {

Stephen James ISON

Page 44 of 81
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

if ($method eq 1) {
    system("./ground.exe");
} else {
    system("./ground2.exe");
}

else {
    if ($material eq "RCC_GrEx") {
        system("./ground2.exe"); # run TCAT
    }
    elsif ($material eq "RCC_Al") {
        system("./ground2.exe"); # run TCAT
    }
    elsif ($material eq "SiC_GrEx") {
        system("./ground2.exe"); # run TCAT
    }
    elsif ($material eq "SiC_Al") {
        system("./ground2.exe"); # run TCAT
    }
    elsif ($material eq "TUPI_GrEx") {
        system("./ground2.exe"); # run TCAT
    } else {
        system("./ground.exe"); # run TCAT
    }
}

# system("touch $name[$loop_index]-temphist");
# system("chmod 777 $name[$loop_index]-temphist");
# system("mv fort.90 groundhold_temphist");

# system("touch $name[$loop_index]-heatratehist");
# system("chmod 777 $name[$loop_index]-heatratehist");
# system("mv fort.91 groundhold_heatratehist");

open(TCATOUTPUT,"<fort.12"); # obtain the current thickness of TPS file
$thickness<$TCATOUTPUT; /
$thickness=$1;
close(TCATOUTPUT);
Sthkns=$thickness;

printf FINALOUTPUT "Groundhold - TPS thickness = %6.3f in.\n",$thickness*100/2.54;

#system("rm point");

# End of heating analysis
# ---------------------------------------------------------------

$avg_thickness=$thkns;
#for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
# $avg_thickness=$avg_thickness+$thkns[$loop_index];
#}
#$avg_thickness=$avg_thickness/$end;

open(DEN,"<material_density"); # obtain the density of TPS material
$Stps_density=$1;
close(DEN);

if ((($sharp eq "yes") && ($bodyside eq "windward")) {
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

$sharp_weight=4.6e-3*bodyarea;  #SHARP weight in lbm
if ($material eq "RCC_GrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*15 77.8347)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    elsif ($material eq "RCC_A1") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    elsif ($material eq "SiC_GrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    elsif ($material eq "SiC_Al") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    elsif ($material eq "TUFI_GrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*13 13.53)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }else {
    $tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    $tile_area_to_body_area=bodytile;
    printf FINALOUTPUT "\n";
    printf FINALOUTPUT "Chosen TPS material was $material\n";
    printf FINALOUTPUT "Nose SHARP TPS: 6.2f lbm\n", $sharp_weight;
    printf FINALOUTPUT "$material unit weight: 6.4f lbm/ft2\n", $tile_unit_weight;
    printf FINALOUTPUT "$material TPS area to body area ratio: 6.2f\n"
    "$tile_area_to_body_area;
    }

if (($sharp eq "yes") && ($bodyside eq "leeward") {
    $blanket_weight=4.6e-3*bodyarea;  #SHARP weight in lbm
    $blanket_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*15 77.8347)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    printf FINALOUTPUT "\n";
    printf FINALOUTPUT "Chosen TPS material was $material\n";
    printf FINALOUTPUT "Nose SHARP TPS: 6.2f lbm\n", $sharp_weight;
    printf FINALOUTPUT "$material unit weight: 6.4f lbm/ft2\n", $tile_unit_weight;
    printf FINALOUTPUT "$material TPS area to body area ratio: 6.2f\n"
    "$tile_area_to_body_area;
    }

if (($sharp eq "no") && ($bodyside eq "windward") {
    if ($material eq "RCC_GrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*15 77.8347)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }else {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }else {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }else {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }else {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0.00254*24 00.00)/0.00254+avg_thickness)/0.4536/3.2808/3.2808/3.2808;  #obtain density of tps
    }
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```perl
elsif ($material eq "TUFI_GrEx") {
    $tile_unit_weight = ($avg_thickness + 0.00254) * 3.2808 * ($avg_thickness + $tps_density + 0.00254 * 13.53) / (0.00254 + $avg_thickness) / 0.4536 / 3.2808 / 3.2808 / 3.2808;
}
else {
    $tile_unit_weight = $avg_thickness * 3.2808 * $tps_density / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
$tile_area_to_body_area = $body_area / $tile_area;
print FINALOUTPUT "Chosen TPS material was $material \n \n ";
print FINALOUTPUT "Material unit weight: \$6.2f lbm/ft2 \n ", $tile_unit_weight;
print "Material TPS area to body area ratio: \$6.2f \n ", $tile_area_to_body_area;

if (($sharp eq "no") && ($bodieside eq "leeward") && $blanket_unit_weight == $avg_thickness * 3.2808 * $tps_density / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
else {
    $blanket_area_to_body_area = $body_area / $blanket_area;
    print FINALOUTPUT "Chosen TPS material was $material \n \n ";
    print FINALOUTPUT "Material unit weight: \$6.2f lbm/ft2 \n ", $blanket_unit_weight;
    print "Material TPS area to body area ratio: \$6.2f \n ", $blanket_area_to_body_area;
}
print "Average thickness: \$avg_thickness\n;"
print "TPS density: \$tps_density\n;"

if ($material_1 eq "RCC_Laminated_Surface") {
    $tile_unit_weight = ($avg_thickness + 0.00254) * 3.2808 * ($avg_thickness + $tps_density + 0.00254 * 15 + 0.00) / (0.00254 + $avg_thickness) / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
else if ($material_1 eq "SiC_Laminated_Surface") {
    $tile_unit_weight = ($avg_thickness + 0.00254) * 3.2808 * ($avg_thickness + $tps_density + 0.00254 * 24 + 0.00) / (0.00254 + $avg_thickness) / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
else if ($material_1 eq "TUFI_Composite_Layer") {
    $tile_unit_weight = ($avg_thickness + 0.00254) * 3.2808 * ($avg_thickness + $tps_density + 0.00254 * 13 + 0.00) / (0.00254 + $avg_thickness) / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
else {
    $tile_unit_weight = $avg_thickness * 3.2808 * $tps_density / 0.4536 / 3.2808 / 3.2808 / 3.2808;
    # Obtain density of TPS
}
print FINALOUTPUT "$\n;"
if ($method eq 1) {
    print FINALOUTPUT "Chosen TPS material was $material_1 \n \n ";
} else {
    print FINALOUTPUT "Chosen TPS material was $material_2 \n \n ";
}
print FINALOUTPUT "TPS unit weight: \$8.4f lbm/ft2 \n ", $tile_unit_weight;

close(FINALOUTPUT);
```

Stephen James Iron
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

# print needed information to screen and file named "thickness.txt"

open(OUT,"<thickness.txt");
open(WEB,"<tcat_output.html");

while(<WEB>){
  if(/Insert stuff here/){
    print "<BR><BR>
    print "$_;
  }
  else{
    print $_;
  }
}
close(WEB);
close(OUT);

#}
Appendix I: ‘compare.cgi’ Ground Hold Multiple Analysis CGI Script

#!/usr/local/bin/perl

# Obtain Information From Web Browse
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM($name).

$PATH='pwd';
chop($PATH);

read(STDIN, $buffer, `ENV('CONTENT_LENGTH')`);
$pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
   ($name, $value) = split(/=/, $pair);
   $value =~ s/\+/\+/g;
   $value =~ s/%([a-fA-FD-g])\([a-fA-FD-g]\)//ig;
   $FORM($name) = $value;
}

# assign information from input website to appropriate variable

$material_option=$FORM(material_option);
$matnum[1]=$FORM(matnum_1);
$matnum[2]=$FORM(matnum_2);
$matnum[3]=$FORM(matnum_3);
$material_1[1]=$FORM(material_1_1);
$material_2[1]=$FORM(material_2_1);
$material_3[1]=$FORM(material_3_1);
$material_4[1]=$FORM(material_4_1);
$material_5[1]=$FORM(material_5_1);
$material_1[2]=$FORM(material_1_2);
$material_2[2]=$FORM(material_2_2);
$material_3[2]=$FORM(material_3_2);
$material_4[2]=$FORM(material_4_2);
$material_5[2]=$FORM(material_5_2);
$material_1[3]=$FORM(material_1_3);
$material_2[3]=$FORM(material_2_3);
$material_3[3]=$FORM(material_3_3);
$material_4[3]=$FORM(material_4_3);
$material_5[3]=$FORM(material_5_3);
$method[1]=$FORM(method_1);
$method[2]=$FORM(method_2);
$method[3]=$FORM(method_3);

$holdtime=$FORM(holdtime);
$timestep=$FORM(timestep);
$ambient=$FORM(ambient);
$fueltemp=$FORM(fueltemp);
$frost=$FORM(frost);

open(TIME, ">time");
print TIME "$holdtime //Time of Groundhold";
close(TIME);

$shearrate = 1;
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

\[ \text{Heat transfer} = \text{Heat rate} \times \text{Ambient} \times 3.15459074506; \]

```
open(HEATXFER,"> heat");
print HEATXFER " heat transfer";  // Heat Flux);
close(HEATXFER);

\text{Ambient} = (\text{Ambient} - 32) \times (5/9) + 273.15;
\text{Frost} = (\text{Frost} - 32) \times (5/9) + 273.15;
\text{Fuel temp} = (\text{Fuel temp} - 32) \times (5/9) + 273.15;
```

```
open(TANK,"> fueltemp");
print TANK " fuel temp";  // Temperature of fuel
close(TANK);
```

```
for ($case = 1; $case <= 3; $case++) { 
  if ($material_option eq 1) {
    for ($loopnum = 1; $loopnum <= $matnum[$case]; $loopnum++) {
      if ($loopnum eq 1) {
        open(DAT, "<material_1[$case]");
        $i = 0;
        while(<DAT>) { 
          $i = $i + 1;
          my ($line) = $_;
          chomp($line);
          if ($i eq 1) { 
            $nodes[1] = $line;
          } elsif ($i eq 2) {
            $cond[1] = $line;
          } elsif ($i eq 3) {
            $density[1] = $line;
          } elsif ($i eq 4) {
            $specheat[1] = $line;
          } elsif ($i eq 5) {
            $chick[1] = $line;
          } elsif ($i eq 6) {
            $temlimit[1] = $line;
          } elsif ($i eq 7) {
            $emissivity = $line;
          }
        }
        close(DAT);
      } elsif ($loopnum eq 2) {
        open(DAT, "<material_2[$case]");
        $i = 0;
        while(<DAT>) { 
          $i = $i + 1;
          my ($line) = $_;
          chomp($line);
          if ($i eq 1) { 
            $nodes[2] = $line;
          } elsif ($i eq 2) {
            $cond[2] = $line;
          } elsif ($i eq 3) {
            $density[2] = $line;
          } elsif ($i eq 4) {
```

Stephen James Lyon
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```perl
$specheat[2] = $line;
} elsif ($i eq 5) {
    $thick[2] = $line;
} elsif ($i eq 6) {
    $stemlimit[2] = $line;
}
}
close(DAT);
}
elsif ($loopnum eq 3) {
    open(DAT, "<$material_3[$case]");
    $i = 0;
    while(<DAT>) {
        $i = $i + 1;
        my($line) = $_;
        chomp($line);
        if ($i eq 1) {
            $nodes[3] = $line;
        } elsif ($i eq 2) {
            $cond[3] = $line;
        } elsif ($i eq 3) {
            $density[3] = $line;
        } elsif ($i eq 4) {
            $specheat[3] = $line;
        } elsif ($i eq 5) {
            $thick[3] = $line;
        } elsif ($i eq 6) {
            $stemlimit[3] = $line;
        }
    }
close(DAT);
}
elsif ($loopnum eq 4) {
    open(DAT, "<$material_4[$case]");
    $i = 0;
    while(<DAT>) {
        $i = $i + 1;
        my($line) = $_;
        chomp($line);
        if ($i eq 1) {
            $nodes[4] = $line;
        } elsif ($i eq 2) {
            $cond[4] = $line;
        } elsif ($i eq 3) {
            $density[4] = $line;
        } elsif ($i eq 4) {
            $specheat[4] = $line;
        } elsif ($i eq 5) {
            $thick[4] = $line;
        } elsif ($i eq 6) {
            $stemlimit[4] = $line;
        }
    }
close(DAT);
}
elsif ($loopnum eq 5) {

Stephen James Ison
Page 51 of 81
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```plaintext
open(DAT, "<material_5[$case]");
$i = 0;

while <$DAT> {
    $i = $i + 1;
    my($line) = <$_;
    chomp($line);
    if ($i eq 1) {
        $nodes[$i] = $line;
    } elsif ($i eq 2) {
        $cond[$i] = $line;
    } elsif ($i eq 3) {
        $density[$i] = $line;
    } elsif ($i eq 4) {
        $specheat[$i] = $line;
    } elsif ($i eq 5) {
        $thick[$i] = $line;
    } elsif ($i eq 6) {
        $temp_limit[$i] = $line;
    }
}

close(DAT);
```

Create the TPS stackup file by reading the properties of each material selected and entering their values into a input file recognizable by T-CAT.

```
# system("touch stackup.in"); #create TPS stackup file
# system("chmod 777 stackup.in"); #change permissions of TPS stackup file
open(STACKUP, ">stackup.in");
print STACKUP "$matnum[$case] /number of materials
";
print STACKUP "$timestep /time step
"

for ($loopnum=1; $loopnum <= $matnum[$case]; $loopnum++) {
    print STACKUP "$i /number of nodes of material $loopnum
", $nodes[$loopnum];
    print STACKUP "$emissivity /emissivity of material $loopnum
";
    print STACKUP "$cond[$i] /thermal conductivity of material $loopnum
";
    print STACKUP "$density[$i] /density of material $loopnum
", $density[$loopnum];
    print STACKUP "$specheat[$i] /specific heat of material $loopnum
", $specheat[$loopnum];
    print STACKUP "$ambient /initial temp (deg K)
";
}
```

Stephen James Icon
printf STACKUP "%f
"  !/TPS thickness of material $loopnum
",$thick[$loopnum];

printf STACKUP "$frost
"  !/Temperature limit of material 1

printf STACKUP "Templimit[1]
"  !/Temperature limit of material 1

for ($loopnum=2; $loopnum <= $matnum[$case]-1; $loopnum++) {
  printf STACKUP "%f
"  !/Temperature limit of material 1
  printf STACKUP "$frost
"  !/Temperature limit of material 1
  printf STACKUP "Templimit[$loopnum]
"  !/Temperature limit of material 1
}

printf STACKUP "Templimit[$matnum[$case]]
"  !/Temperature limit of material 1

print STACKUP "$fueltemp
"  !/backface temperature limit

print STACKUP "$strat
"  !/istrat parameter

print STACKUP "$iopt
"  !/iopt parameter

print STACKUP "$ioned
"  !/ioned parameter

print STACKUP "$iprint
"  !/iprint parameter

close(STACKUP);

This section of the program creates a mock MINIVER file. The three columns of data created are:
- the time,
- ambient temperature,
- rate of heat transfer.

$numtimes = $holdtime/$stimestep + 1;
$time[1] = 0;
for ($i=2; $i <= $numtimes; $i++) {
  $time[$i] = $time[$i-1] + $stimestep;
}

#system("touch l_stackup.s");  #create mock MINIVER file
#system("chmod 777 l_stackup.s");  #change permissions of file
open(MINI,"<l_stackup.s");
print MINI "Mock MINIVER file for input into ground hold analysis
1.0
";
for ($i=1; $i <= $numtimes; $i++) {
  print MINI "Stime[$i] $ambient $heattransfer
";
}

print MINI "-100.0"

# MINIVER file is partitioned into body point files
# partition the MINIVER file into separate body point files that include time,
# heat rate, and radiation equilibrium temperature
# -----------------------------------------------

$i=0;
$filecount=0;
open(FILE,"<l_stackup.s");  #open statement with filehandle FILE
until(eof FILE) {
  #goes through the MINIVER file until the end is reached
  print MINI 
"#initialization of counter index "$i"
  $i=$i+1;
  $filecount=$filecount+1;
  open(FILE,"<l_stackup.s");
  for ($j=2; $j <= $stprel; $j++) {
    print FILE "Stime[$j] $ambient $heatrate
";
  }
  close(FILE);
  open(FILE,"<l_stackup.s");
  for ($j=2; $j <= $stprel; $j++) {
    print FILE "Stime[$j] $ambient $heatrate
";
  }
  close(FILE);
}
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

START: $==<FILE>; #skips the first line of the data file
if (/\.*(\S*)\/\*/) { #searches for the bodypoint label
$filecount-0;

# system(\"touch groundhold\"); #create body point file
# system(\"chmod 777 groundhold\"); #change permissions inorder to write to body point file

$filecount-$filecount+1;
} # end of if statement

while(<FILE>) { #while loop that reads thru each line of the file
$i-0;

$readline-

$filecount-$filecount+1;
} #end of while loop

# ------------------------
# incrementally move each body point file into \"miniver.in\" and execute tcat perl script that creates the filenames needed inorder to batch execute tcat
# filenames are created via input from the user
# inputs (file_start_number, file_end_number, body_side)
# ------------------------

$start-1; #start of the file index
$end-($filecount-1)/2; #end of the file index

open(FILE, \">inputs_for_outputs\"); #open statement for new file "newsample"
printf FILE \"Stimesteps \n\#: formatted print statement"
for($i=1; $i<$n; $i++) {
printf FILE \"$timesteps $time[$i] $i\n\"; print NEWFILE \"$timesteps \n\#: formatted print statement"
printf NEWFILE \"$timesteps $time[$i] $i\n\"; # end of for loop

$readline-

$end-1;
open(NEWFILE, \">groundhold\"); #open statement for new file \"newsample\"$stimesteps-$n-1;
printf NEWFILE \"Stimesteps \n\#: formatted print statement"
for($i=1; $i<$n; $i++) {
printf NEWFILE \"$timesteps $time[$i] $i\n\"; print NEWFILE \"$timesteps $time[$i] $i\n\"; } # end of for loop

close(NEWFILE);
$i-0;
go to START
} #end of if statement

) #end of while loop
) #end of until loop
close(FILE);

# ------------------------
# assign material properties
# ------------------------

if ($material_option eq 1) {

Stephen James Izon
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

$inputfile = "stackup.in";

} else {

if ($material eq "AETB8_Al") {
    $inputfile="AETB_8_AL_5Inputs.in";
}

if ($material eq "AETB8_GrEx") {
    $inputfile="AETB_8_GrEx_5Inputs.in";
}

if ($material eq "AETB12_Al") {
    $inputfile="AETB_12_AL_5Inputs.in";
}

if ($material eq "AETB12_GrEx") {
    $inputfile="AETB_12_GrEx_5Inputs.in";
}

if ($material eq "FRCI12_Al") {
    $inputfile="FRCI_12_AL_5Inputs.in";
}

if ($material eq "FRCI12_GrEx") {
    $inputfile="FRCI_12_GrEx_5Inputs.in";
}

if ($material eq "FRCI20_Al") {
    $inputfile="FRCI_20_AL_5Inputs.in";
}

if ($material eq "FRCI20_GrEx") {
    $inputfile="FRCI_20_GrEx_5Inputs.in";
}

if ($material eq "LI900_Al") {
    $inputfile="LI_900_AL_5Inputs.in";
}

if ($material eq "LI900_GrEx") {
    $inputfile="LI_900_GrEx_5Inputs.in";
}

if ($material eq "LI2200_Al") {
    $inputfile="LI_2200_AL_5Inputs.in";
}

if ($material eq "LI2200_GrEx") {
    $inputfile="LI_2200_GrEx_5Inputs.in";
}

if ($material eq "RCC_GrEx") {
    $inputfile="RCC_GrEx_5Inputs.in";
}

if ($material eq "RCC_Al") {
    $inputfile="RCC_AL_5Inputs.in";
}

if ($material eq "SiC_GrEx") {
    $inputfile="SiC_GrEx_5Inputs.in";
}

if ($material eq "SiC_Al") {
    $inputfile="SiC_AL_5Inputs.in";
}

if ($material eq "TUFI_GrEx") {
    $inputfile="TUFI_GrEx_5Inputs.in";
}

if ($material eq "AFRSI_Al") {
    $inputfile="AFRSI_AL_3Inputs.in";
}

if ($material eq "AFRSI_GrEx") {
    $inputfile="AFRSI_GrEx_3Inputs.in";
}

if ($material eq "CFBI_Al") {
    $inputfile="CFBI_AL_3Inputs.in";
}

if ($material eq "CFBI_GrEx") {
    $inputfile="CFBI_GrEx_3Inputs.in";
}

if ($material eq "DURAFRSI_Al") {
    $inputfile="DURAFRSI_AL_3Inputs.in";
}

if ($material eq "DURAFRSI_GrEx") {
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```bash
$inputfile="DURAFRSI_GrEx_3inputs.in";
if ($material eq "PBI_A1") {
    $inputfile="PBI_A1_3inputs.in";
} else
    if ($material eq "PBI_GrEx") {
        $inputfile="PBI_GrEx_3inputs.in";
    } else
        if ($material eq "TABI_A1") {
            $inputfile="TABI_A1_3inputs.in";
        } else
            if ($material eq "TABI_GrEx") {
                $inputfile="TABI_GrEx_3inputs.in";
            }
}

# Execute TCAT and send ADS output into file named "junkoutput"

system("touch inputs.in");
system("chmod 777 inputs.in");
system("cp $inputfile inputs.in >> error 2>&1");

system("touch inputs[$case]");
system("chmod 777 inputs[$case]");
system("cp inputs.in inputs[$case]");

if ($material_option eq 1) {
    if ($method[$case] eq 1) {
        system("./ground.exe");
    } else {
        system("./ground2.exe");
    }
} else {
    if ($material eq "RCC_GrEx") {
        system("go4.exe>junkoutput");  # run TCAT
    } elsif ($material eq "RCC_A1") {
        system("go4.exe>junkoutput");  # run TCAT
    } elsif ($material eq "SiC_GrEx") {
        system("go4.exe>junkoutput");  # run TCAT
    } elsif ($material eq "SiC_A1") {
        system("go4.exe>junkoutput");  # run TCAT
    } elsif ($material eq "TUFI_GrEx") {
        system("go4.exe>junkoutput");  # run TCAT
    } else {
        system("go3.exe>junkoutput");  # run TCAT
    }
}

system("touch $name[$loop_index]_temphist");
system("chmod 777 $name[$loop_index]_temphist");
system("mv fort.90 groundhold_temphist[$case] >> error3");

system("touch $name[$loop_index]_heatratehist");
system("chmod 777 $name[$loop_index]_heatratehist");
system("mv fort.91 groundhold_heatratehist[$case] >> error4");
```
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```c
open(TCATOUTPUT, "<fort.12");  # obtain the current thickness of TPS file
S = <TCATOUTPUT>;
\[ S = (S')^2/8 + (S')^2; \]
$thickness'=S1;
.close(TCATOUTPUT);
$thkns=$thickness;
printf FINALOUTPUT "Groundhold - TPS thickness = %6.3f in.\n", $thickness*100/2.54;

#system("rm point");
# End of heating analysis
#

$avg_thickness=$thkns;
$for (Sloop_index=$start; $loop_index <$end; $loop_index++) {
  $avg_thickness=$avg_thickness+$thkns[$loop_index];
}$avg_thickness=$avg_thickness/$end;
open(DEN, "<material_density");  # obtain the density of TPS material
$ = <DEN>;
$tps_density=S1;
close(DEN);
if ($sharp eq "yes") && ($bodyside eq "windward") {
  $sharp_weight=4.6e-3*$bodyarea;  #SHARP weight in lbm
  if ($material eq "RCCtGrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*15
    77.8347/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } elsif ($material eq "RCCtAl") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*15
    77.8347/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } elsif ($material eq "SiCtGrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*24
    00.00)/0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } elsif ($material eq "SiCtAl") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*24
    00.00)/0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } elsif ($material eq "TUFItGrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*13
    13.53)/0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } elseif ($material eq "PTUFI-GrEx") {
    $tile_unit_weight=($avg_thickness+0.00254)*3.2808*(Savg_thickness*$tps_density+0.00254*13
    13.53)/0.00254+$avg_thickness)/0.4536/3.2808/3.2808;  #obtain density of tps
  } else {
    $tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808;  #obtain density of tps
  }
  $tile_area_to_body_area=$bodytile;
  printf FINALOUTPUT "\n";
  printf FINALOUTPUT "Chosen TPS material was $material\n";
  printf FINALOUTPUT "Nose SHARP TPS: %.2f lbm\n", $sharp_weight;
  printf FINALOUTPUT "Material unit weight: %.4f lbm/ft2 \n ", $tile_unit_weight;
  printf FINALOUTPUT "Material TPS area to body area ratio: %.4f \n ", $tile_area_to_body_area;
}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
  $sharp_weight=4.6e-3*$bodyarea;  #SHARP weight in lbm
```
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

$blanket_unit_weight=$avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
$blanket_area_to_body_area=$bodyblanket:
printf FINALOUTPUT "$\n";
printf FINALOUTPUT "Chosen TPS material was $material\n";
printf FINALOUTPUT "Nose SHARP TPS: %6.2f lbm/ft2 \n",$sharp_weight;
printf FINALOUTPUT "Material unit weight: %6.2f lbm/ft2 \n ",$blanket_unit_weight;
printf FINALOUTPUT "Material TPS area to body area ratio: %6.2f \n",$blanket_area_to_body_area;
)
if (($sharp eq "no") && ($bodyside eq "windward")) {  
if ($material eq "RCC_GrEx") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
}else ($material eq "RCC_Al") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
}else ($material eq "SiC_GrEx") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
}else ($material eq "SiC_Al") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
} else {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
}  
$tile_area_to_body_area=$bodytile;
printf FINALOUTPUT "$\n";
printf FINALOUTPUT "Material unit weight: %6.2f lbm/ft2 \n ",$tile_weight;
printf FINALOUTPUT "Material TPS area to body area ratio: %6.2f \n",$tile_area_to_body_area;
)
if (($sharp eq "no") && ($bodyside eq "leeward")) {  
$blanket_unit_weight=$avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
$blanket_area_to_body_area=$bodyblanket;
printf FINALOUTPUT "Chosen TPS material was $material\n";
printf FINALOUTPUT "Material unit weight: %6.2f lbm/ft2 \n ",$blanket_unit_weight;
printf FINALOUTPUT "Material TPS area to body area ratio: %6.2f \n",$blanket_area_to_body_area;
)
printf FINALOUTPUT "\n";
printf FINALOUTPUT "Material unit weight: %6.2f lbm/ft2 
 
",$blanket_unit_weight;
printf FINALOUTPUT "Material TPS area to body area ratio: %6.2f 
",$blanket_area_to_body_area;
}

if ($material[0] eq "RCC_Laminate_Surface") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
} else if ($material[0] eq "SiC_Laminate_Surface") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
} else if ($material[0] eq "TUFI_GrEx") {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
} else {
$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density/0.4536/3.2808/3.2808/3.2808;  
# obtain density of tps
}
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

```bash
if ($material-1[$case] eq "TUFI-Composite_Layer") {
    $tile_unit_weight = ($avg_thickness*0.00254)*3.2808*($avg_thickness*$tps_density*0.00254*13/13.53)/0.4536/3.2808/3.2808;  # Obtain density of tps
} else {
    $tile_unit_weight = $avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808;  # Obtain density of tps
}
printf FINALOUTPUT "\n";
if ($method[$case] eq 1) {
    printf FINALOUTPUT "Chosen TPS material was $material-1[$case]\n\n";
} else {
    printf FINALOUTPUT "Chosen TPS material was $material-2[$case]\n\n";
}
printf FINALOUTPUT "TPS unit weight: %8.4f lbm/ft2 \n ", $tile_unit_weight;
close(FINALOUTPUT);
$racchick[$case] = $avg_thickness;
$case_density[$case] = $tps_density;
$unitweight[$case] = $tile_unit_weight;
system("mv thickness.txt thickness[$case].txt >> error4");
}
    $min = 3;
    $min = 2;
    $min = 3;
    $min = 1;
    $min = 1;
    $min = 2;
}
open(RESULTS,">results.txt");
printf RESULTS "The optimum design for this analysis is configuration $min\n";
printf RESULTS "The configuration consists of the following materials:\n";
printf RESULTS "Material 1: $material-1[$min]\n";
if ($matnum[$min] gt 1) {
    printf RESULTS "Material 2: $material-2[$min]\n";
} else if ($matnum[$min] gt 0) {
    printf RESULTS "Material 3: $material-3[$min]\n";
}
```

Stephen James Izen
if ($matnum[$min] gt 3) {
    printf RESULTS " Material 4: $material_4[$min]\n";
}

if ($matnum[$min] gt 4) {
    printf RESULTS " Material 5: $material_5[$min]\n";
}

$finalweight=$casethick[$min]*3.2808*$casedensity[$min]/0.4536/3.2808/3.2808/0.4536;
printf RESULTS "\n";
printf RESULTS "------TPS Ground hold design results------\n";
printf RESULTS "You chose to optimize layer $method[$min] of the TPS stack.\n";

if ($method[$min] eq 1) {
    printf RESULTS "The TPS material for layer $method[$min] is $material_1[$min].\n";
    printf RESULTS "$material_1[$min]\n" thickness = $6.3f in.\n",$casethick[$min]*100/2.54;
    printf RESULTS "TPS unit weight: %8.4f lbm/ft2 \n ",$finalweight;
} else {
    printf RESULTS "The TPS material for layer $method[$min] is $material_2[$min].\n";
    printf RESULTS "$material_2[$min]\n" thickness = $6.3f in.\n",$casethick[$min]*100/2.54;
    printf RESULTS "TPS unit weight: %8.4f lbm/ft2 \n ",$finalweight;
}

close(RESULTS);

# ____________________--------------------------------------------------------------
# print needed information to screen and file named "thickness.txt"
# _____________________------------------------_------------------------------------
open(OUT, "<results.txt"); # opens the input file for TCAT script
print "Content-type:text/html\n"
while (<WEB>){
    if (/Insert stuff here/) {
        print "<BR>\n";
        while (<OUT>){
            print "<BR>\n";
            print ";
        }
    } else {
        print ";
    }
}
close(WEB);
close(OUT);

##
Appendix J: ‘ground.f’ FORTRAN Source Code

program ground

************************************************************
* Program to solve the one dimensional heat equation through a stackup *
* of disparate TPS materials. This program finds the optimum thickness *
* of the surface layer of the TPS material to reduce the TPS unit *
* weight, while eliminating frosting at the surface of the TPS stackup. *
************************************************************

implicit none
integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
integer iter,maxiter
real*8 dt,k (6),den (6),cp (6),initT,L (6),dx(6),sigma,epsil,eps
1,alpha(6),lam(6),a(400),b(400),c(400),T(10000,400),normf
1,normdelT,f(400),delt(400),twonorm,time(10000)
1,qcond(400),grad(400),qcv(400),g(24),obj,x(12),dTTemp
1,kair,Tb(1000),error,diff(1000),temmlmt(10),backT,heat
1,root(1000),toler

************************************************************
* Obtains information from the file 'inputs.in' which contains TPS *
* material properties, used in the solution of the heat equation. *
************************************************************

open(13,file='inputs.in')
read(10,*) matnum        !/number of materials
read(10,*) dt            !/time step
do i=1,matnum
read(10,*) nodes(i)      !/number of nodes per material
enddo
read(10,*) epsil         !/emissivity of surface
do i=1,matnum
read(10,*) k(i)          !/thermal conductivities
enddo
do i=1,matnum
read(10,*) den(i)         !/densities
enddo
do i=1,matnum
read(10,*) cp(i)          !/specific heats
enddo
read(10,*) initT          !/initial temperatures
do i=1,matnum
read(10,*) L(i)           !/thicknesses
enddo
do i=1,2*matnum
read(10,*) temmlmt(i)     !/temp limits for each material
enddo

************************************************************
* Writes the density of the surface TPS material for use in calculating *
* the TPS material unit weight in the perl script. *
* Also writes the heat transfer rate and the backface fuel temperature *
* to files for use in the one-dimensional heat equation solver. *
************************************************************

open(40, file='heat')
read(40,*) heat
close(40)
open(41, file='fueltemp')
read(41,*) backT
close(41)
open(9, file='material_density')
write(9,*) den(1)
close(9)
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

* Initializes optimization parameters. *

```plaintext
toler=0.0001
error=1.
maxiter=100
```

* Solves the heat equation for using two initial guesses for the *
  * thickness of the TPS. Values obtained for the objective function *
  * will be sent to the next while loop which implements the *
  * Newton-Raphson method to converge to the solution. *

```plaintext
if (matnum.eq.1) then
    call one(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
              time, qcv, qcond, qrad, backT, heat)
elsife (matnum.eq.2) then
    call two(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
              time, qcv, qcond, qrad, backT, heat)
elsife (matnum.eq.3) then
    call three(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
              time, qcv, qcond, qrad, backT, heat)
elsife (matnum.eq.4) then
    call four(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
              time, qcv, qcond, qrad, backT, heat)
elsife (matnum.eq.5) then
    call five(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
              time, qcv, qcond, qrad, backT, heat)
endif

```plaintext
diff(1)-T(n,1)-templmt(1)
diff(2)-T(n,1)-templmt(1)
```

* While loop to call the heat equation solver. The while loop *
  * reiterates until the temperature of the surface lies above the *
  * frosting condition (40F), to within the specified convergence *
  * criteria (error = 0.009).

```plaintext
iter=2
```
```
dowhile((error.gt.toler).and.(iter.lt.maxiter))
```
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

\[
\text{iter} = \text{iter} + 1
\]

\[
\text{root(iter)} = \text{root(iter-1)} - \left( (\text{diff(iter-1)} \times (\text{root(iter-2)} - \text{root(iter-1)}) \right) / (\text{diff(iter-2)} - \text{diff(iter-1)})
\]

\[
\text{L}(1) = \text{root(iter)}
\]

if (matnum == 1) then
  call one(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, time, qcv, qcond, qrad, backT, heat)
else if (matnum == 2) then
  call two(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, time, qcv, qcond, qrad, backT, heat)
else if (matnum == 3) then
  call three(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, time, qcv, qcond, qrad, backT, heat)
else if (matnum == 4) then
  call four(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, time, qcv, qcond, qrad, backT, heat)
else if (matnum == 5) then
  call five(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, time, qcv, qcond, qrad, backT, heat)
endif

\[
\text{error} = \text{abs}((\text{root(iter)} - \text{root(iter-1)}) / \text{root(iter)})
\]

enddo

<formatted file output statements to write temperature profiles at key nodes of the TPS material stackup.>

time(1) = 0.
do j = 2, n
  time(j) = time(j-1) + dt
enddo

write(90, 2)
do j = 1, n
  write(90, 4) time(j), qcv(j), qrad(j), qcond(j)
endo if (matnum == 1) then
  write(90, 1)
do j = 1, n
  write(90, 3) time(j), T(j, 1), T(j, nodes(1))
endo
else if (matnum == 2) then
  write(90, 5)
do j = 1, n
  write(90, 6) time(j), T(j, 1), T(j, nodes(1)), T(j, nodes(1) + nodes(2))
endo
else if (matnum == 3) then
  write(90, 7)
do j = 1, n
  write(90, 8) time(j), T(j, 1), T(j, nodes(1)), T(j, nodes(1) + nodes(2) + nodes(3))
endo
else if (matnum == 4) then
  write(90, 9)
do j = 1, n
  write(90, 10) time(j), T(j, 1), T(j, nodes(1)), T(j, nodes(1) + nodes(2)), T(j, nodes(1) + nodes(2) + nodes(3)), T(j, nodes(1) + nodes(2) + nodes(3) + nodes(4))
endo
else if (matnum == 5) then
  write(90, 11)
do j = 1, n
  write(90, 12) time(j), T(j, 1), T(j, nodes(1)), T(j, nodes(1))
endo
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

1+nodes(2)),T(j,nodes(1)+nodes(2)+nodes(3)),T(j,nodes(1)+nodes(2)
1+nodes(3)+nodes(4)),T(j,nodes(1)+nodes(2)+nodes(3)+nodes(4)+
1nodes(5))
enddo
endif
write(12,*)L(1)

******************************************************
* Formatting parameters for file output.
******************************************************
1 format (2x, 'time (j)', 7x, 'T(j,1)', 5x, 'T(j,nodes(1))')
2 format (2x, 'time', 7x, 'qconv', 5x, 'qrad', 5x, 'qcond')
3 format (f12.2, f12.2, f12.2)
4 format (f12.2, 3x, f11.2, 3x, f11.2, 3x, f11.2)
5 format (5x, 'time', 5x, 'T(j,1)', 5x, 'T(j,nodes1)', 5x
1, 'T(j,totnodes)')
6 format (f12.2, f12.2, f12.2, f12.2)
7 format (2x, 'time (j)', 7x, 'T(j,1)', 5x, 'T(j,nodes(1))', 2x
1, 'T(j,sum1)', 5x, 'T(j,sum2)')
8 format (f8.2, 7x, f8.2, 5x, f8.2, 4x, f8.2, 5x, f8.2)
9 format (2x, 'time (j)', 7x, 'T(j,1)', 5x, 'T(j,nodes(1))', 2x
1, 'T(j,sum1)', 5x, 'T(j,sum2)', 5x, 'T(j,totnodes)')
10 format (f8.2, 7x, f8.2, 5x, f8.2, 4x, f8.2, 5x, f8.2, 5x, f8.2)
11 format (2x, 'time (j)', 7x, 'T(j,1)', 5x, 'T(j,nodes(1))', 2x
1, 'T(j,sum1)', 5x, 'T(j,sum2)', 5x, 'T(j,sum3)', 5x, 'T(j,totnodes)')
12 format (f8.2, 7x, f8.2, 5x, f8.2, 4x, f8.2, 5x, f8.2, 5x, f8.2, 5x, f8.2)
end
Appendix K: 'ground2.f' FORTRAN Source Code

program ground

**************************************************************************
* Program to solve the one dimensional heat equation through a stackup *
* of disparate TPS materials. This program finds the optimum thickness *
* of the surface layer of the TPS material to reduce the TPS unit weight, *
* while eliminating frosting at the surface of the TPS stackup. *
**************************************************************************
implicit none
integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
integer itcr,maxiter
real(kind=8) dt,k
den(6),cp(6),initT,L(6),dx(6),sigma,eps,eps1,alp(6),lam(6),a(400),b(400),c(400),T(10000,400),normf
l,normdelT,f(400),delT(400),twonorm,time(10000)
1.gcond(400),qrad(400),qcv(400),g(24),h(12),dTemp
l,kair,Tb(1000),error,diff(1000),templmt(10),backT,heat
1.root(1000),toler

**************************************************************************
* Obtains information from the file 'inputs.in' which contains TPS *
* material properties, used in the solution of the heat equation. *
**************************************************************************
open(10, file='inputs.in')
read(10,*) matnum
read(10,*) dt
do i=1,matnum
read(10,*) nodes(i)
do j=1,nodes(i)
den(i)
do i=1,matnum
read(10,*) cp(i)
do i=1,matnum
read(10,*) initT
enddo
do i=1,matnum
read(10,*) L(i)
do i=1,2
read(10,*) templmt(i)
do i=1,matnum
enddo
enddo
read(10,*) epsi
enddo
do i=1,matnum
read(10,*) k(i)
do i=1,matnum
enddo
do i=1,matnum
read(10,*) dx(i)
doo i=1,matnum
read(10,*) sigma
enddo
do i=1,matnum
enddo
read(10,*) eps1
enddo
do i=1,matnum
enddo
read(10,*) alp(i)
doo i=1,matnum
read(10,*) lam(i)
doo i=1,matnum
read(10,*) a(i)
doo i=1,matnum
enddo
do i=1,matnum
enddo
read(10,*) b(i)
doo i=1,matnum
enddo
do i=1,matnum
enddo
read(10,*) c(i)
doo i=1,matnum
enddo
read(10,*) T(i)
doo i=1,matnum
enddo

**************************************************************************
* Writes the density of the surface TPS material for use in calculating *
* the TPS material unit weight in the perl script. *
* Also writes the heat transfer rate and the backface fuel temperature *
* to files for use in the one-dimensional heat equation solver. *
**************************************************************************
open(40, file='heat')
read(40,*) heat
close(40)

open(41, file='fueltemp')
read(41,*) backT
close(41)

open(9, file='material_density')
write(9,*) den(2)
close(9)
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

* Initializes optimization parameters.

```plaintext
* toler=0.0001
* error=1.
* maxiter=100
```

* Solves the heat equation for using two initial guesses for the thickness of the TPS. Values obtained for the objective function will be sent to the next while loop which implements the Newton-Raphson method to converge to the solution.

```plaintext
open(44,file='opt')

if (matnum.eq.1) then
    call one(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.2) then
    call two(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.3) then
    call three(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.4) then
    call four(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.5) then
    call five(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
endif

diff(1)=T(n,1)-templt(1)
root(1)=L(2)+0.006

if (matnum.eq.1) then
    call one(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.2) then
    call two(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.3) then
    call three(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.4) then
    call four(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
else if (matnum.eq.5) then
    call five(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n,
    + time,qcv,qcond,grad,backT,heat)
endif

diff(2)=T(n,1)-templt(1)
root(2)=L(2)+0.006
```

* While loop to call the heat equation solver. The while loop reiterates until the temperature of the surface lies above the frosting condition (40F), to within the specified convergence.
* criteria (error = 0.009).

*--------------------------------------------------------------------------*

iter = 2

dowhile (error.gt.toler).and.(iter.lt.maxiter)

   iter = iter + 1

   root(iter) = root(iter-1) - ((diff(iter-1) * (root(iter-2)
   + root(iter-1))) / (diff(iter-2) - diff(iter-1)))

   L(2) = root(iter)

   if (matnum.eq.1) then
      call one(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
      + time, qcv, qcond, qrad, backT, heat)

   elseif (matnum.eq.2) then
      call two(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
      + time, qcv, qcond, grad, backT, heat)

   elseif (matnum.eq.3) then
      call three(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
      + time, qcv, qcond, grad, backT, heat)

   elseif (matnum.eq.4) then
      call four(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
      + time, qcv, qcond, grad, backT, heat)

   elseif (matnum.eq.5) then
      call five(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n,
      + time, qcv, qcond, grad, backT, heat)

   endif

   diff(iter) = T(n,1) - templt(1)

   error = abs((root(iter) - root(iter-1))/root(iter))

   write(44,*) diff(iter)

endo

close(44)

*--------------------------------------------------------------------------*

* Formatted file output statements to write temperature profiles
* at key nodes of the TPS material stackup.

*--------------------------------------------------------------------------*

time(1) = 0.

do j=2,n
   time(j) = time(j-1) + dt
endo

write(91,2)

do j=1,n
   write(91,4) time(j), qcv(j), grad(j), qcond(j)
endo

if (matnum.eq.1) then
   write(90,1)
   do j=1,n
      write(90,3) time(j), T(j,1), T(j,nodes(1))
   enddo
else
   write(90,5)
   do j=1,n
      write(90,6) time(j), T(j,1), T(j,nodes(1)), T(j, nodes(1)
   enddo
endif

1+nodes(2))

else (matnum.eq.3) then
   write(90,7)
   do j=1,n
      write(90,8) time(j), T(j,1), T(j,nodes(1)), T(j, nodes(1)
   enddo
endif

Stephen James Ion
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

elseif (matnum.eq.4) then
  write(90,9)
  do j=1,n
    write(90,10) time(j), T(j,1), T(j,nodes(1)), T(j,nodes(1))
    1+nodes(2), T(j,nodes(1)+nodes(2)+nodes(3)), T(j,nodes(1)+nodes(2))
    1+nodes(3)+nodes(4))
  enddo
elseif (matnum.eq.5) then
  write(90,11)
  do j=1,n
    write(90,12) time(j), T(j,1), T(j,nodes(1)), T(j,nodes(1))
    1+nodes(2), T(j,nodes(1)+nodes(2)+nodes(3)), T(j,nodes(1)+nodes(2))
    1+nodes(3)+nodes(4)), T(j,nodes(1)+nodes(2)+nodes(3)+nodes(4)+
    1nodes(5))
  enddo
endif
write(12,*)L(2)

*****************************************************************************
* Formatting parameters for file output.                                    *
*****************************************************************************
1 format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))')
2 format(2x,'time',7x,'qconv',5x,'qgrad',5x,'qcond')
3 format(f12.2,f12.2,f12.2)
4 format(f12.2,3x,f11.2,3x,f11.2,3x,f11.2)
5 format(5x,'time',5x,'T(j,1)',5x,'T(j,nodes(1))',5x
1,'T(j,totnodes)')
6 format(f12.2,f12.2,f12.2,f12.2)
7 format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
1,'T(j,sum1)',5x,'T(j,sum2)')
8 format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2)
9 format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
1,'T(j,sum1)',5x,'T(j,sum2)',5x,'T(j,totnodes)')
10 format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2,5x,f8.2)
11 format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
1,'T(j,sum1)',5x,'T(j,sum2)',5x,'T(j,sum3)',5x,'T(j,totnodes)')
12 format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2,5x,f8.2,5x,f8.2)
end
Appendix L: 'one.f' FORTRAN Subroutine

subroutine one(matnum,dt,nodes,eps, k,den,cp,initT,L,T,n,
+ time,qcv,qcond,qrad,backT,heat)

******************************************************************************
* Subroutine to solve the one dimensional heat equation for a TPS         *
* material stackup consisting of one materials.                          *
* This subroutine incorporates an isothermal boundary condition at the    *
* backface of the TPS material stackup (generally the cryogenic fuel)     *
* temperature) and also a constant heat flux boundary condition at the   *
* surface of the TPS material stackup. The constant heat flux is         *
* determined solely by the temperature of the ambient air.               *
******************************************************************************

implicit none
integer i,j,n,nodes(6),suml,sum2,sum3,totnodes,matnum,count
real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,eps,eps
alpha(6),lam(6),a(400),b(400),c(400),T(10000,400),normf
1,normeltf(400),delt(400),twnorm,Tradeq(10000),time(10000)
1,qcond(10000),qrad(10000),qcv(10000),g(24),obj,x(12),Tb(400)
1,backT,heat

******************************************************************************
* Calls linint subroutine to linearly interpolate convective heat flux    *
* and Radiation Equilibrium temperature values using user specified       *
* time and time step for analysis.                                       *
******************************************************************************
call linint(n,dt,time,qcv,Tradeq)

******************************************************************************
* Define the mathematical constants to be used in the calculations.       *
******************************************************************************

eps=1.e-6
sigma=5.67051e-8
dx(i)=L(i)/(nodes(i)-1.)
alpha(i)=k(i)/(den(i)*cp(i)*1000.)
lam(i)=alpha(i)*dt/(dx(i)**2)
enddo
suml=nodes(1)+nodes(2)
sum2=sum1+nodes(3)
sum3=sum2+nodes(4)
totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)

******************************************************************************
* Specify the initial conditions of the problem.                         *
******************************************************************************
do i=1,totnodes
T(1,i)=initT
enddo
do i=1,totnodes
Tb(i)=T(1,i)
enddo
Tb(totnodes)=backT
do i=1,totnodes
T(1,i)=Tb(i)
enddo

******************************************************************************
* Initialize time array.                                                 *
******************************************************************************
time(1)=0.

Stephen James Icon
* Do loop to solve for the temperature profiles for each time step j

```
do j=2,n

* Setup the tridiagonal matrix to be used in the Thomas Algorithm.

```

```
a(1)=-2.*lam(1)
b(1)=1.+2.*lam(1)
c(1)=0.
do i=2,nodes(l)-1
  a(i)=-1.*lam(l)
  b(i)=1.+2.*lam(1)
  c(i)=-1.*lam(1)
enddo
```

```
a(totnodes)=0.
b(totnodes)=1.
c(totnodes)=0.
time(j)=time(j-1)+dt
Tb(1)=Tb(1)+2.*lam(1)*heat*dx(1)/k(1)
call tridag(a,b,c,Tb,delT,totnodes)
do i=1,totnodes
  T(j,i)=delT(i)
  Tb(i)=delT(i)
enddo
```

```
* Calculation of the radiative and conductive heat fluxes.

```
```
do j=1,n
  qrad(j)=epsi*sigma*T(j,1)**4
  qcond(j)=qv(j)-qrad(j)
enddo
```

```
* End of subroutine one.

```

```
return
```
```
Appendix M: `two.f` FORTRAN Subroutine

```
subroutine two(matnum,dt,nodes,epsilon,k,den,initT,L,n,
time,qcv,qcond,qrad,backT,heat)

* Subroutine to solve the one dimensional heat equation for a TPS *
* material stackup consisting of two disparate materials. *
* This subroutine incorporates an isothermal boundary condition at the *
* backface of the TPS material stackup (generally the cryogenic fuel *
* temperature) and also a constant heat flux boundary condition at the *
* surface of the TPS material stackup. The constant heat flux is *
* determined solely by the temperature of the ambient air. *

implicit none
integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
real*8 dt, k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsilon,epsilon1,
alpha(6),lam(6),a(400),b(400),c(400),T(10000,400),normf
1,normdelT,f(400),delT(400),twonorm,Tadeq(10000),time(10000)
1,qcond(10000),qrad(10000),qcv(10000),g(24),obj,x(12),Tb(400)
1,backT,heat

* Calls linint subroutine to linearly interpolate convective heat flux *
* and Radiation Equilibrium temperature values using user specified *
* time and time step for analysis. *

call linint(n,dt,time,qcv,Tadeq)

* Define the mathematical constants to be used in the calculations. *
eps1=1.e-6
sigma=5.67036e-8
do i=1,matnum
   dx(i)=L(i)/(nodes(i)-1.)
   alpha(i)=k(i)/(den(i)*cp(i)**1000.)
   lam(i)=alpha(i)*dt/(dx(i)**2)
endo
sum1=nodes(1)+nodes(2)
sum2=sum1+nodes(3)
sum3=sum2+nodes(4)
totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)

* Specify the initial conditions of the problem. *
do i=1,totnodes
   T(1,i)=initT
endo

do i=1,totnodes
   Tb(i)=T(1,i)
endo

* Initialize time array. *
time(1)=0.
```

Stephen James Izon
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

**********************************************************************
* Do loop to solve for the temperature profiles for each time step j *
**********************************************************************

do j=2,n

**********************************************************************
* Setup the tridiagonal matrix to be used in the Thomas Algorithm. *
**********************************************************************

        a(1) = -2.*lam(1)
        b(1) = 1.*lam(1)
        c(1) = 0.

do i=2,nodes(1)-1
        a(i) = 1.*lam(1)
        b(i) = 1.*lam(1)
        c(i) = 1.*lam(1)
.enddo

        a(nodes(1)) = -2.*lam(2)
        b(nodes(1)) = 1.*lam(2) + 2.*lam(2)*k(1)/k(2)
        c(nodes(1)) = -2.*lam(2)*k(1)/k(2)

do i=nodes(1)+1,sum1=1
        a(i) = -1.*lam(2)
        b(i) = 1.*lam(2)
        c(i) = 1.*lam(2)
.enddo

a(totnodes) = 0.
b(totnodes) = 1.
c(totnodes) = 0.
time(j) = time(j-1) + dt

Tb(1) = Tb(1) + 2.*lam(1)*heat*dx(1)/k(1)
call tridag(a,b,c,Tb,delT,totnodes)
do i=1,totnodes
    T(j,i) = delT(i)
    Tb(i) = delT(i)
.enddo

enddo

**********************************************************************
* Calculation of the radiative and conductive heat fluxes. *
**********************************************************************

do j=1,n
    qrad(j) = epsi*sigma*T(j,1)**4
    qcond(j) = qcv(j) - qrad(j)
.enddo

**********************************************************************
* End of subroutine two. *
**********************************************************************

return
end
Appendix N: 'three.f' FORTRAN Subroutine

```fortran
subroutine three(matnum, dt, nodes, epsi, k, den, cp, initT, L, T, n, 
    time, qcv, qcond, qrad, backT, heat)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* Subroutine to solve the one dimensional heat equation for a TPS    *
* material stackup consisting of three disparate materials.           *
* This subroutine incorporates an isothermal boundary condition at the *
* backface of the TPS material stackup (generally the cryogenic fuel  *
* temperature) and also a constant heat flux boundary condition at the *
* surface of the TPS material stackup. The constant heat flux is      *
* determined solely by the temperature of the ambient air.           *
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
implicit none
integer i, j, n, nodes(6), sum1, sum2, sum3, totnodes, matnum, count
real*8 dt, k(6), den(6), cp(6), initT, L(6), dx(6), sigma, epsi, eps
1, alpha(6), lam(6), a(400), b(400), c(400), f(10000, 400), normf
1, normdelT, f(400), delT(400), twonorm, Tradeq(10000), time(10000)
1, qcond(10000), qrad(10000), qcv(10000), q(24), obj, x(12), Tb(400)
1, backT, heat

call linint(n, dt, time, qcv, Tradeq)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* Calls linint subroutine to linearly interpolate convective heat flux *
* and Radiation Equilibrium temperature values using user specified  *
* time and time step for analysis.                                   *
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

eps=1.e-6
sigma=5.6706e-8
do i=1, matnum
    dx(i)=L(i)/(nodes(i)-1.)
    alpha(i)=k(i)/(den(i)*cp(i)*1000.)
    lam(i)=alpha(i)*dt/(dx(i)**2)
enddo
sum1=nodes(1)+nodes(2)
sum2=sum1+nodes(3)
sum3=sum2+nodes(4)
totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* Specify the initial conditions of the problem.                    *
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
do i=1, totnodes
    T(1,i)=initT
enddo
do i=1, totnodes
    Tb(i)=T(1,i)
enddo
Tb(totnodes)=backT
do i=1, totnodes
    T(1,i)=Tb(i)
enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* Initialize time array.                                           *
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
time(1)=0.
```

---

Stephen James Izon
**Improvements in Thermal Protection Systems Sizing Capabilities for TCAT**

* Do loop to solve for the temperature profiles for each time step j

```fortran
* Setup the tridiagonal matrix to be used in the Thomas Algorithm.
```

```
   a(1) = -2.0*lam(1)
   b(1) = 1.0 + 2.0*lam(1)
   c(1) = 0.
   do i=2,nodes(1)-1
      a(i) = -1.0*lam(i)
      b(i) = 1.0 + 2.0*lam(i)
      c(i) = -1.0*lam(i)
   enddo

   a(nodes(1)) = -2.0*lam(2)
   b(nodes(1)) = 1.0 + 2.0*lam(2) + 2.0*lam(2)*k(1)/k(2)
   c(nodes(1)) = -2.0*lam(2)*k(1)/k(2)
   do i=nodes(1)+1,sum1-1
      a(i) = -1.0*lam(3)
      b(i) = 1.0 + 2.0*lam(3)
      c(i) = -1.0*lam(3)
   enddo

   a(sum1) = -2.0*lam(3)
   b(sum1) = 1.0 + 2.0*lam(3) + 2.0*lam(3)*k(2)/k(3)
   c(sum1) = -2.0*lam(3)*k(2)/k(3)
   do i=sum1,sum2-1
      a(i) = -1.0*lam(3)
      b(i) = 1.0 + 2.0*lam(3)
      c(i) = -1.0*lam(3)
   enddo

   a(totnodes) = 0.
   b(totnodes) = 1.
   c(totnodes) = 0.

   time(j) = time(j-1) + dt
   Tb(1) = Tb(1) + 2.0*lam(1)*heat*dx(1)/k(1)

   call tridag(a,b,c,Tb,delt,totnodes)
   do i=1,totnodes
      T(j,i) = delt(i)
      Tb(i) = delt(i)
   enddo
```

* Calculation of the radiative and conductive heat fluxes.

```fortran
* End of subroutine three.
```

```
   qrad(j) = epsi*sigma*T(j,1)**4
   qcond(j) = qcv(j) - qrad(j)
```

* End of subroutine three.

```
return
end
```
Appendix O: ‘four.f’ FORTRAN Subroutine

```fortran
subroutine four(matnum,dt,nodes,epsi,den,dp,initT,L,T,n,i,
time,qcv,qcond,qrad,backT,heat)

* Subroutine to solve the one dimensional heat equation for a TPS
* material stackup consisting of four disparate materials.
* This subroutine incorporates an isothermal boundary condition at the
* backface of the TPS material stackup (generally the cryogenic fuel
* temperature) and also a constant heat flux boundary condition at the
* surface of the TPS material stackup. The constant heat flux is
* determined solely by the temperature of the ambient air.

implicit none
integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
real*8 dt,dx(6),den(6),dp(6),initT,L(6),sigma,epsi,eps1
alpha(6),lam(6),e(400),x(12),Tb(400)
normTf(400),delT(400),time(10000),T(10000,400),normTf
1,qcond(10000),qrad(10000),qcv(10000),g(24),obj,x(12)
1,backT,heat

* Calls linint subroutine to linearly interpolate convective heat flux
* and equilibrium temperature values using user specified
* time and time step for analysis.

call linint(n,dt,time,qcv,Tradeq)

* Define the mathematical constants to be used in the calculations.
eps=1.e-6
sigma=5.67061e-8
do i=1,matnum
dx(i)=L(i)/(nodes(i)-1)
alpha(i)=k(i)/(den(i)*dp(i)**2)
enddo
sum1=nodes(1)+nodes(2)
sum2=sum1+nodes(3)
sum3=sum2+nodes(4)
totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)

* Specify the initial conditions of the problem.

do i=1,totnodes
T(1,i)=initT
enddo
do i=1,totnodes
Tb(i)=T(1,i)
enddo
Tb(totnodes)=backT

do i=1,totnodes
T(1,i)=Tb(i)
enddo
```

Stephen James Ison
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

time(1)=0.

* Do loop to solve for the temperature profiles for each time step j *

**.................................................................**

* Setup the tridiagonal matrix to be used in the Thomas Algorithm. *

**.................................................................**

```
!improve 1

do j=2,n

!improve 2

!improve 3

a(1)=2.+lam(1)
b(1)=1.+2.*lam(1)
c(1)=0.
do i=2,nodes(1)-1
   a(i)=-1.*lam(1)
b(i)=1.+2.*lam(1)
c(i)=-1.*lam(1)
endo

a(nodes(1))=-2.*lam(2)
b(nodes(1))=1.+2.*lam(2)+2.*lam(2)*k(1)/k(2)
c(nodes(1))=-2.*lam(2)*k(1)/k(2)
do i=nodes(1)+1,sum1-1
   a(i)=-1.*lam(2)
b(i)=1.+2.*lam(2)
c(i)=-1.*lam(2)
endo

a(sum1)=-2.*lam(3)
b(sum1)=1.+2.*lam(3)+2.*lam(3)*k(2)/k(3)
c(sum1)=-2.*lam(3)*k(2)/k(3)
do i=sum1+1,sum2-1
   a(i)=-1.*lam(3)
b(i)=1.+2.*lam(3)
c(i)=-1.*lam(3)
endo

a(sum2)=-2.*lam(4)
b(sum2)=1.+2.*lam(4)+2.*lam(4)*k(3)/k(4)
c(sum2)=-2.*lam(4)*k(3)/k(4)
do i=sum2+1,sum3-1
   a(i)=-1.*lam(4)
b(i)=1.+2.*lam(4)
c(i)=-1.*lam(4)
endo

a(sum3)=-1.
b(sum3)=1.
c(sum3)=0.

time(j)=time(j-1)+dt

Tb(1)=Tb(1)+2.*lam(1)*heat*dx(1)/k(1)
call tridag(a,b,c,Tb,delt.totnodes)
do i=1,totnodes
   T(j,i)=delt(i)
   Tb(i)=delt(i)
endo
```

* Calculation of the radiative and conductive heat fluxes. *

```
!improve 4

do j=1,n
   qrad(j)=epsi*sigma*T(j,1)**4
   qcond(j)=qcv(j)-qrad(j)
endo
```
Improved in Thermal Protection Systems Sizing Capabilities for TCAT

* End of subroutine four. *

return
end
Appendix P: ‘five.f’ FORTRAN Subroutine

```
subroutine five(matnum,dt,nodes,epsilon,k,den,cp,initT,L,T,n,
    time,qcv,qcond,qrad,backT,heat)

* Subroutine to solve the one dimensional heat equation for a TPS material stackup consisting of five dissparate materials.
* This subroutine incorporates an isothermal boundary condition at the backface of the TPS material stackup (generally the cryogenic fuel temperature) and also a constant heat flux boundary condition at the surface of the TPS material stackup. The constant heat flux is determined solely by the temperature of the ambient air.

implicit none
integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsilon,epsilon_l
alpha(6),lam(6),a(400),b(400),c(400),T(10000,800),normf
1,normdelT,f(400),delt(400),twnorm,Tradeq(10000),time(10000)
1,qcond(10000),qgrad(10000),qcv(10000),g(24),obj,x(12),Tb(400)
1,backT,heat

* Calls linint subroutine to linearly interpolate convective heat flux and Radiation Equilibrium temperature values using user specified time and time step for analysis.

call linint(n,dt,time,qcv,Tradeq)

* Define the mathematical constants to be used in the calculations.

eps=1.e-6
sigma=5.67061e-8
do i=1,matnum
dx(i)=L(i)/(nodes(i)-1.)
alpha(i)=k(i)/(den(i)*cp(i)*1000.)
lam(i)=alpha(i)*dt/(dx(i)**2)
endo
sum1=nodes(1)+nodes(2)
sum2=sum1+nodes(3)
sum3=sum2+nodes(4)
totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)

do i=1,totnodes
T(1, i)=initT
endo

do i=1,totnodes
Tb(i)=T(1, i)
endo
Tb(totnodes)=backT

do i=1,totnodes
T(1, i)=Tb(i)
endo

* Initialize time array.

time(1)=0.
```

Stephen James Fzon
* Do loop to solve for the temperature profiles for each time step j

* Setup the tridiagonal matrix to be used in the Thomas Algorithm.

```plaintext
a(1) = -2.0 * lam(1)
b(1) = 1.0 + 2.0 * lam(1)
c(1) = 0.0

do i = 2, nodes(1) - 1
    a(i) = -1.0 * lam(i)
b(i) = 1.0 + 2.0 * lam(i)
c(i) = 1.0 * lam(i)
endo

a (nodes(1)) = -2.0 * lam(2)
b (nodes(1)) = 1.0 + 2.0 * lam(2) + 2.0 * lam(2) * k(1) / k(2)
c (nodes(1)) = -2.0 * lam(2) * k(1) / k(2)

do i = nodes(1) + 1, sum1 + 1
    a(i) = -1.0 * lam(2)
b(i) = 1.0 + 2.0 * lam(2)
c(i) = 1.0 * lam(2)
endo

a (sum1) = -2.0 * lam(3)
b (sum1) = 1.0 + 2.0 * lam(3) + 2.0 * lam(3) * k(2) / k(3)
c (sum1) = -2.0 * lam(3) * k(2) / k(3)

do i = sum1 + 1, sum2 - 1
    a(i) = -1.0 * lam(3)
b(i) = 1.0 + 2.0 * lam(3)
c(i) = 1.0 * lam(3)
endo

a (sum2) = -2.0 * lam(4)
b (sum2) = 1.0 + 2.0 * lam(4) + 2.0 * lam(4) * k(3) / k(4)
c (sum2) = -2.0 * lam(4) * k(3) / k(4)

do i = sum2 + 1, sum3 - 1
    a(i) = -1.0 * lam(4)
b(i) = 1.0 + 2.0 * lam(4)
c(i) = 1.0 * lam(4)
endo

a (sum3) = -2.0 * lam(5)
b (sum3) = 1.0 + 2.0 * lam(5) + 2.0 * lam(5) * k(4) / k(5)
c (sum3) = -2.0 * lam(5) * k(4) / k(5)

do i = sum3 + 1, totnodes - 1
    a(i) = -1.0 * lam(5)
b(i) = 1.0 + 2.0 * lam(5)
c(i) = 1.0 * lam(5)
endo

a (totnodes) = 0.0
b (totnodes) = 1.0

c (totnodes) = 0.0

time (j) = time (j - 1) + dt

tb (1) = tb (1) + 2.0 * lam(1) * heat * dx(1) / k(1)

call tridiag (a, b, c, Tb, delT, totnodes)

do i = 1, totnodes
    Tb (i) = delT (i)
endo

[enddo]
```

* Calculation of the radiative and conductive heat fluxes.
Improvements in Thermal Protection Systems Sizing Capabilities for TCAT

********************************************************************************

do j=1,n
   qrad(j)=eps*sigma*T(j,1)**4
   qcond(j)=qcv(j)-qrad(j)
enddo

********************************************************************************

* End of subroutine four.  *
********************************************************************************

return
end
References


