THERMAL ANALYSIS METHODS FOR AEROBRAKING HEATING

Ruth M. Amundsen, Joe F. Gasbarre, John A. Dec
NASA Langley Research Center

ABSTRACT
As NASA begins exploration of other planets, a method of non-propulsively slowing vehicles at the planet, aerobraking, may become a valuable technique for managing vehicle design mass and propellant. An example of this is Mars Reconnaissance Orbiter (MRO), which will launch in late 2005 and reach Mars in March of 2006. In order to save propellant, MRO will use aerobraking to modify the initial orbit at Mars. The spacecraft will dip into the atmosphere briefly on each orbit, and during the drag pass, the atmospheric drag on the spacecraft will slow it, thus lowering the orbit apoapsis. The largest area on the spacecraft, and that most affected by the heat generated during the aerobraking process, is the solar arrays. A thermal analysis of the solar arrays was conducted at NASA Langley, to simulate their performance throughout the entire roughly 6-month period of aerobraking. Several interesting methods were used to make this analysis more rapid and robust. Two separate models were built for this analysis, one in Thermal Desktop for radiation and orbital heating analysis, and one in MSC.Patran for thermal analysis. The results from the radiation model were mapped in an automated fashion to the Patran thermal model that was used to analyze the thermal behavior during the drag pass. A high degree of automation in file manipulation as well as other methods for reducing run time were employed, since toward the end of the aerobraking period the orbit period is short, and in order to support flight operations the runs must be computed rapidly. All heating within the Patran Thermal model was combined in one section of logic, such that data mapped from the radiation model and aeroheating model, as well as skin temperature effects on the aeroheating and surface radiation, could be incorporated easily. This approach calculates the aeroheating at any given node, based on its position and temperature as well as the density and velocity at that trajectory point. Run times on several different processors, computer hard drives, and operating systems (Windows versus Linux) were evaluated.

INTRODUCTION
Several missions to other planets have already employed aerobraking in order to lessen the amount of propellant carried. Thermal analysis of several of these missions has been performed at NASA Langley.[1, 2] The most recent mission under analysis is Mars Reconnaissance Orbiter (MRO), which will aerobrake at Mars from March to September 2006. The thermal team for this type of aerobraking mission is often part of the navigation team, since thermal effects are the determining factor in how low into the atmosphere the spacecraft can progress without damage. On MRO, the area most affected by the heating during aerobraking is the solar arrays. These arrays are the lightest part of the spacecraft, and present a very large area (over 23 m^2) that is directly facing the generated wind. The exposed skin of the solar array is a composite material, unprotected by multi-layer insulation (MLI), and generally has a relatively low temperature limit (for example, on MRO, this limit is 175°C). This limit on the heating defines the trajectory that is planned by the MRO project. In order to facilitate flight operations, the thermal team must be

able to respond quickly to planned maneuvers, and to off-nominal situations such as higher-than-expected atmospheric density. When a situation is encountered that could lead to high heating of the solar array, the thermal team must be able to quickly perform an analysis of both the expected situation, and also any projected recovery maneuvers, to determine the potential thermal outcomes. For these reasons, it is critical that the thermal analysis be robust, flexible, and able to produce new predictions quickly. The analysis scheme for MRO includes several features designed to allow the thermal team to respond quickly to changes in the orbit, atmospheric density, or other parameters, and complete an updated prediction as rapidly as possible.

Analysis Approach

There are several types of thermal analysis necessary to support a navigation team in flight operations for an aerobraking mission. The most frequent is simply to predict the nominal temperatures for the next orbit pass.   Another is to respond to a perceived or expected change in trajectory, atmospheric density, velocity, spacecraft orientation, or model accuracy and produce an updated prediction. Another function is to produce a "limit line" prediction. This prediction entails running several heating cases for each of a selected set of orbit passes, to produce a smooth curve defining the limiting heating for each orbit pass. This allows the navigation team to effectively plan maneuvers and drag passes so as to get the most benefit from aerobraking without wasting maneuvering fuel and without excessive heating of the solar arrays. In each of these analysis scenarios, the main intent is to produce a prediction quickly, re-running only the necessary portions.

The thermal solver used was MSC.Patran Thermal[3], from MSC Software. This software was used for thermal analysis on earlier aerobraking missions, thus the method and customized software had already been developed. Since Patran Thermal does not incorporate orbital (solar or planetary) radiation calculation, it is necessary to use another solver as well. Thermal Desktop[4] RadCAD module, from Cullimore and Ring Technologies, was the software chosen for orbital heating calculations (as was done on earlier missions as well). Normally, the use of two separate thermal software packages and the development of two separate thermal models would not be seen as the most robust and efficient method. However, in this case, the preliminary preparation time is not as critical as the ability to respond quickly during flight operations once the models are developed.

The Thermal Desktop RadCAD software allows the analyst to produce maps of solar and planetary heating over the area of the solar arrays, as a function of time, over the entire orbit. On earlier analyses, these maps were interpolated onto the Patran nodal grid during the thermal run. This adds unnecessary time to the Patran Thermal analysis.   In order to minimize the time in the thermal solver execution, a change was made to the method. In current analyses, these heating gradients are mapped to the Patran nodal grid using an external macro, prior to the Patran Thermal run, thus allowing them to be used directly with no interpolation. This removes the time for this step from daily computations during flight operations. Since the Thermal Desktop results will only change if there is a significant change to the orbit trajectory, this step will rarely have to be repeated.

The flow of information among the segments of the thermal analysis process is shown in Figure 1.
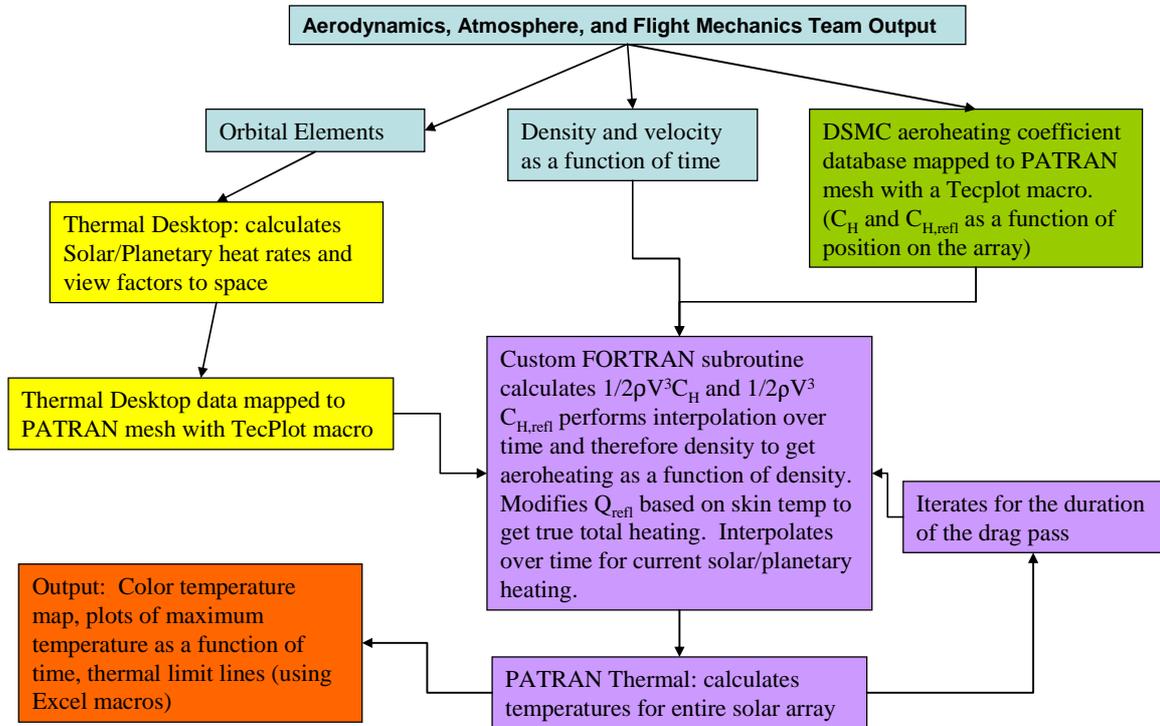

Figure 1. Aerobraking thermal analysis sequence.

Aeroheating is provided for each drag pass in the following way. Aeroheating is calculated as

$$Q = \tfrac{1}{2} * \rho * V^3 * C_H$$

where ρ is the atmospheric density, V is the spacecraft velocity relative to the atmosphere, and C_H is the heat transfer coefficient. The C_H maps are calculated by Direct Simulation Monte Carlo (DSMC) Analysis Code (DAC) and free-molecular methods (DAC-FREE) for a selection of densities, as shown in Figure 2. DSMC is used to calculate $C_{H\ incident}$ as well as $C_{H\ reflected}$. In DSMC, the incident heating is that imparted to the surface by incoming molecules. Reflected heating is that leaving the surface due to outgoing (reflected) molecules. In order to maintain accuracy, the $C_{H\ reflected}$ was corrected for the skin temperature of the given portion of the array at the given point in time using

$$C_{H_{refl}} = C_{H_{refl-DSMC}} * \frac{T_{wall}}{300}$$

where T_{wall} is in K. The value of 300 is present because the DSMC calculations were made with a constant skin temperature of 300K. The $C_{H\ total}$ is then $C_{H\ incident} - C_{H\ reflected}$. (In Figure 2 the $C_{H\ reflected}$ has not yet been corrected for actual skin temperature.)

In earlier analyses, these C_H maps were pulled into Patran Thermal, and at each time step a double interpolation was made – the spatial interpolation from the DSMC grid to the Patran Thermal grid, and the temporal interpolation from the density timeline to the actual solver time. In order to optimize solution time, part of this step was, similar to the Thermal Desktop mapping, pulled out and run prior to the analysis as a discrete macro. The mapping of C_H from the DSMC grid to the Patran nodes is accomplished before the runs are made. This removes the time for that step from the thermal runs, and it does not have to be repeated unless the Patran Thermal mesh changes or the C_H prediction changes. Since these two things will be happening with much less frequency than the normal runs, it makes sense to do this step in advance.
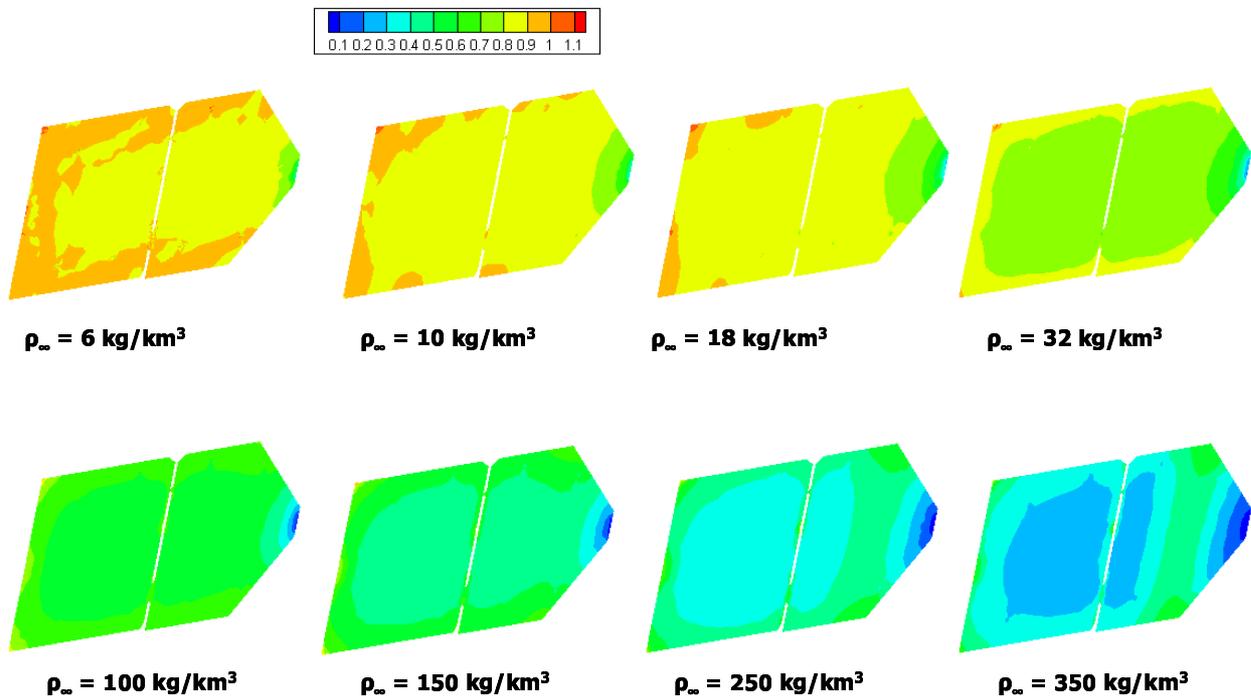

| 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 |

$\rho_\infty = 6$ kg/km^3 $\rho_\infty = 10$ kg/km^3 $\rho_\infty = 18$ kg/km^3 $\rho_\infty = 32$ kg/km^3

$\rho_\infty = 100$ kg/km^3 $\rho_\infty = 150$ kg/km^3 $\rho_\infty = 250$ kg/km^3 $\rho_\infty = 350$ kg/km^3

Figure 2. Aeroheating coefficients ($C_{H\ total}$) from DSMC.

The time history of ρ and V for the actual drag pass are provided by the flight mechanics team. In the thermal solution, the heating at each point in time is interpolated from the ρ and V for that time, and the C_H at that density.

Another change that has been made in the modeling method is to sum all the heat loads in a single loop of the Fortran module within Patran Thermal that evaluates the boundary conditions. Thus, the orbital heat fluxes, aeroheating heat flux, and radiation off the surface are all handled within the same loop. This allows application of the view factors, orbital and aeroheating fluxes mapped on the Patran nodal grid within the same software loop.

MODEL DEVELOPMENT
Both the Patran Thermal and Thermal Desktop models were developed manually from the geometry and drawings of the MRO spacecraft. The Thermal Desktop model is shown in Figure 3. The spacecraft bus is included in this model, in order to properly simulate the shadowing effects and reflections to the solar arrays. In the Thermal Desktop model, only exterior optical

properties are included. The output from this model is a set of solar and planetary heating maps for one solar array, around the entire orbit for each of ten different orbits, mapped to the Patran nodal grid. The thermal analysis is done on a solar array only, not the entire spacecraft. Cullimore and Ring created a custom function in Thermal Desktop that allows all cases to be run, mapped and output with a single button click. The Thermal Desktop model has 725 surfaces in the entire model, with 4488 nodes on each solar array. The run of a single orbit's heat rate in Thermal Desktop takes approximately 100 minutes to complete.
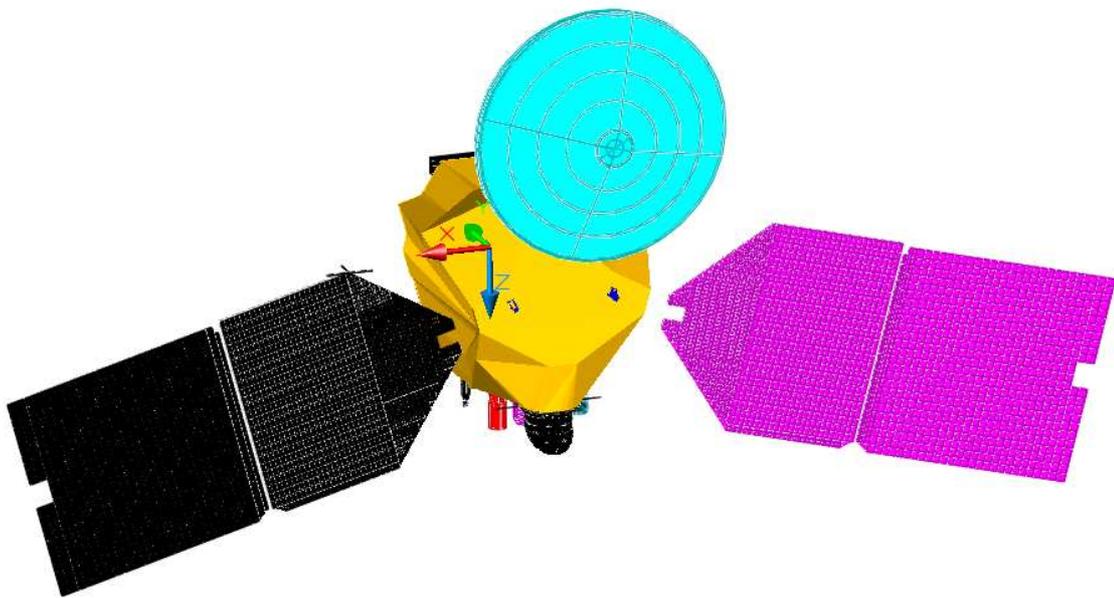

Figure 3. Thermal Desktop model of MRO spacecraft and solar arrays.

The Patran Thermal model for the MRO solar array is shown in Figure 4. The model is of only a single solar array; it contains 88,650 nodes and 73,300 elements. The model consists of five layers: two for the composite facesheets, one for a central aluminum honeycomb core, and one for the solar cell layer. Each layer is connected by a contact conductance that approximates the adhesive bond. Masses of cabling, solder, and adhesive for the solar cells are smeared evenly in the solar cell layer. The as-built mass of the MRO solar array is 48.3 kg, while the Patran model mass is 47.5 kg. The model is thus slightly conservative in terms of mass, being 1.6% light. Incorporated in the model are the differing thickness of facesheets and core between the inboard and outboard sections, differing densities and thickness at the hard points and hinge connections, and a Kapton layer around the outer circumference of each section, which adds a small amount of mass, and is intended to protect the edges of the array from the higher heating rates.

An example of the aeroheating flux mapped onto the Patran mesh is shown in Figure 5. This aeroheating example has been corrected for $C_{H\,reflected}$, and the actual skin temperature. An example of temperature predictions for the MRO solar array is shown in Figure 6.
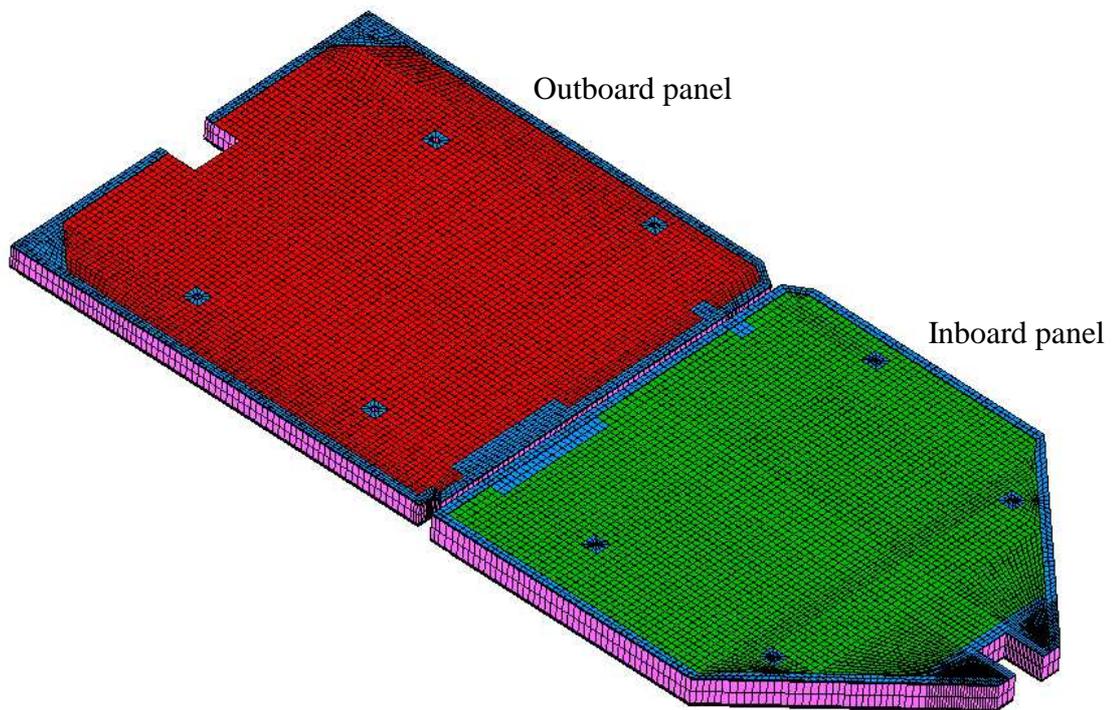
Outboard panel

Inboard panel

Figure 4. Patran Thermal model of MRO solar array.

.48
.44
.40
.37
.33
.29
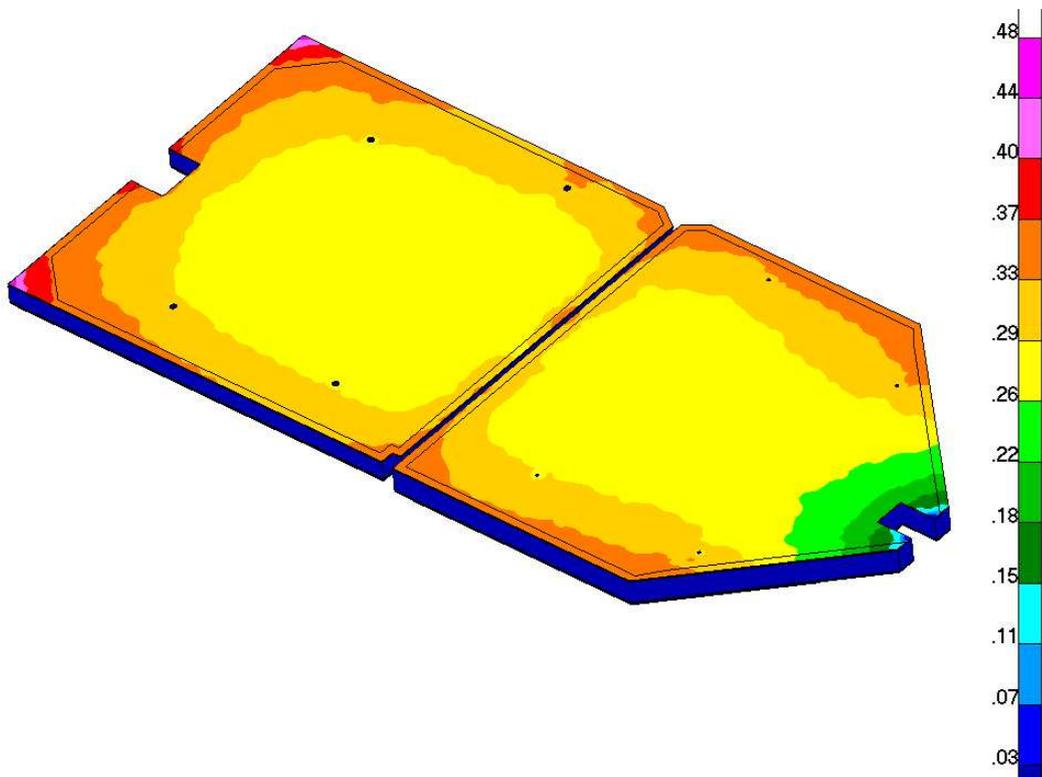.26
.22
.18
.15
.11
.07
.03

Figure 5. Example aeroheating for MRO (W/cm^2).
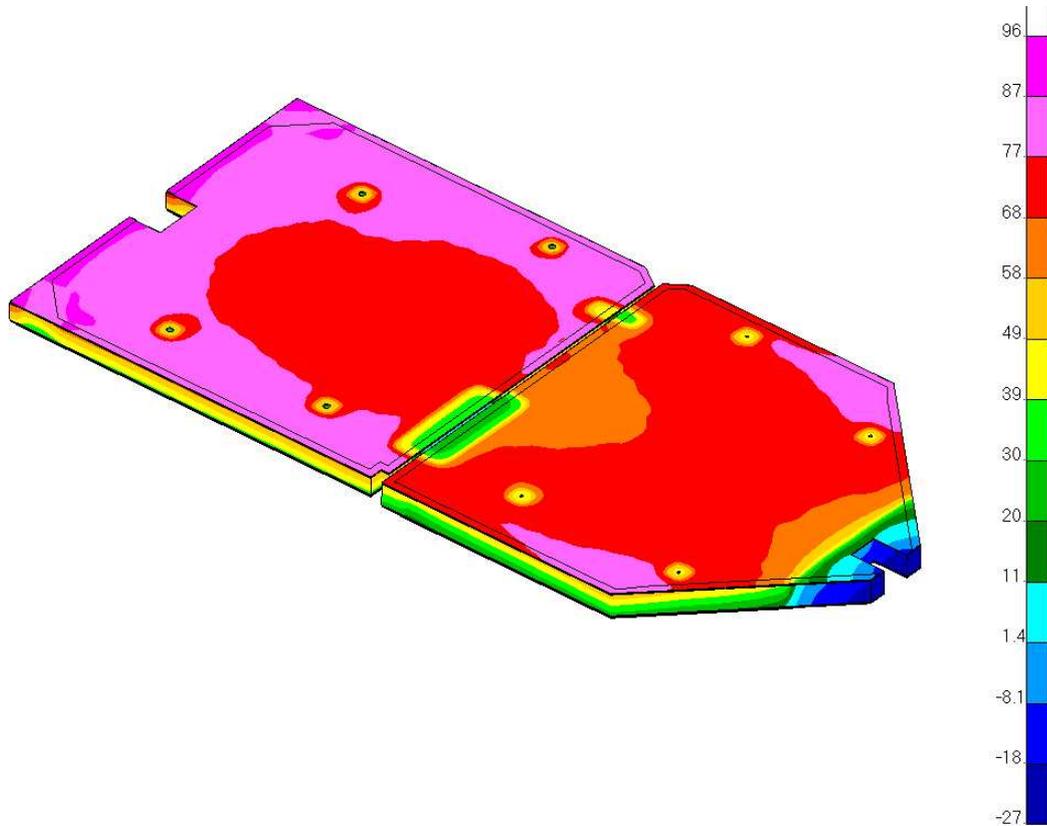
Figure 6. Example temperature distribution for MRO solar array (°C).

Many parametric studies are planned for this MRO mission, as would be useful for any aerobraking analysis. One that has been performed already is to evaluate the effect of the of radiative sink temperature assumption. During aerobraking, the heated surface of the solar array radiates to a variable temperature atmospheric sink. The temperature of this sink is transient and complex to determine, being a function of the transparency of the atmosphere in the radiating wavelength of the solar array, the temperature of the array, the amount of atmosphere viewed versus deep space, the time history of spacecraft depth into the atmosphere, the orientation of the spacecraft, etc. Since this parameter is difficult to determine, it was decided to bracket the possible values that this parameter could have to determine its effect. The lowest conceivable value for the average sink temperature of the Martian atmosphere was assumed to be 10K (slightly above the deep space constant of 3K). This value, as well as 20, 50 and 100K, was used as potential average sink temperatures. The value of 100K was taken as the absolute maximum that this variable could be, since the atmospheric temperature of Mars is low, the spacecraft will remain high in the atmosphere where the atmosphere is thin (fairly transparent in the IR), and at least half the viewing will be of deep space. The radiation from the planet does not need to be included in this sink, since that is already calculated in Thermal Desktop. Values of 10, 20 and 50K caused changes in peak array temperature of less than 0.1°C. Even the highest value of 100K yielded only an increase of 1.4°C in the peak temperature. Thus, this parameter was assumed to have negligible effect.

Another parametric study was performed to evaluate the effect of the accommodation coefficient. This coefficient is an empirical value that alters the aeroheating reflected from the surface ($Q_{reflected}$ as calculated by DSMC) in the following manner:

$$Q_{reflected} = \frac{1}{2} * \rho * V^3 * C_H * AC$$

where AC is the accommodation coefficient. The accommodation coefficient is dependent on the surface characteristics, such as material type, roughness, and catalytic behavior. This parameter is not well-known, so again bracketing values were used. The maximum possible value of AC is 1.0. Values of 0.8 and 0.5 were also used. The maximum range of AC resulted in a change in peak temperature of 5°C for the highest heating cases, and 3°C for more nominal cases. A value of AC=0.8 will be used for future analyses, as this should capture the vast majority of cases (3σ).

AUTOMATION METHODS

There have been several efficiency improvements incorporated in the analysis methodology over the last few years. In order to support flight operations as efficiently as possible, there has been an effort to automate everything feasible, in order to minimize the labor needed to turn around a new analysis. Simple scripts have been developed that perform an entire sequence of 40 runs with a single button click. The only intervention by the user is to change the name of the necessary file(s) for the new portion of the run. The needed changes to the macro are minimized because of the modular nature of the analysis. If only a change to the vacuum phase of the orbit (such as shadow time) is made, then only the file location of the orbital heating file is changed in the analysis – everything else can remain the same. If a change is made only in the aeroheating, then all the orbital analysis files remain unchanged. These macros also delete unneeded files after the run is complete, to cut down on the space required for storage. Use of these macros allows a larger number of people to help support flight operations, as the analyst does not have to necessarily understand every detail of the models, as long as they understand which files need to be changed to run the scripts correctly.

The entire process of mapping the DSMC and Thermal Desktop result onto the Patran mesh is also automated within a single macro. If the Patran mesh is changed, this macro can still be used, by just changing a single file that defines the Patran node locations.

These macros have been written as DOS scripts, although some of them reference Tecplot macros within them (for the nodal mapping, for example). This allows them to be easily moved from system to system, and run on any Windows machine. Also, by using the 'at' command in DOS, a sequence of run scripts can be lined up, and several days worth of analysis can be performed autonomously on a single computer.

The Patran Thermal model employs an extensive amount of user-developed Fortran code in the solution. One function of this code is to compute the maximum temperature within each material type at every time point, and print that to an output file. This allows simple methods of checking if any material in the model is exceeding its recommended service temperature. Temperatures

for all nodes of interest are output, as are the aeroheating values used for that trajectory. This code has been streamlined to facilitate quicker runs.

Another helpful automation has been to write macros in Microsoft Excel that automate the process of combining the results from 40 runs and constructing a limit line. These macros assemble the time history of density and velocity for all 40 orbit sequences, add the temperature predictions for the maximum material temperatures for all 40 runs, and construct a limit line based on a curve fit through those 40 points. This allows the user to construct an updated limit line in a matter of seconds after a run is complete.

RUN TIME IMPROVEMENTS
At the initiation of this effort to improve the run times, a sequence of 40 runs necessary to produce a limit line for MRO consumed over 20 days. The 40 runs are necessary to assess four heating scenarios for each of ten orbit passes that encompass the range of orbit pass conditions. The four heating scenarios encompass the band of possible behavior, from being at the lowest heating line possible (high in the atmosphere) to the worst case highest heating trajectory (lowest in the atmosphere).

At the beginning of the MRO effort, each run of a single nominal drag pass took roughly six to eight hours (run on Windows on a 2.4 GHz Xeon™ processor). The highest heating scenarios, however, each took over 24 hours to complete. The effect was that for the combination of 40 cases, over 23 CPU days was needed to complete the runs. This run time includes the improvements discussed above, which extract the nodal mapping from the Patran Thermal run and insert it into a separate macro.

The first run time improvement came about due to the upgrade of Patran 2004 to Patran 2005. In this upgrade, new Fortran and C++ compilers (Intel 8.0) were implemented. Running with these compilers and the current Patran version at the time (Patran 2005) brought the individual runs down to the range of two to six hours per run, which brought the time required to produce a limit line down to about nine CPU days. The high heating cases were still the much more time-intensive components of the analysis.

Another change was to reduce the thermal solver accuracy by changing the value of the EPISIT parameter in Patran Thermal. This parameter defines the maximum change in temperature that is used as a convergence criterion for the solver. The original value used was $0.0005°C$; this was increased to $0.005°C$ without affecting the overall thermal results ($<0.4°C$ change at peak). Run time improvement was about 35%.

By a detailed examination of the thermal transient results, it was obvious that the 4x higher run time in the high heating cases was due to a sudden decrease in the time step as the maximum heating portion of the trajectory was approached. By evaluating which nodes were driving this decrease in time step, an error was discovered in the thickness of the outboard panel facesheet -- it was too low. This left those nodes drastically too low in mass, thus forcing a short time step for them to come to convergence under high heating. Correction of this error led to the time for the highest heating cases coming down into the same range as the other cases, and also improved

the solution time on all cases by about 35%. This brought the total time to complete 40 runs and produce a new limit line down to just over two CPU days (over a factor of four improvement).

The model mass was also corrected to align with the as-built mass. The original model was roughly 25% too light. Most of the unaccounted mass was in the solar cell layer, as there were several components in this layer which had not originally been included in the model. Including these brought the peak temperatures down slightly (<2°C on the highest heating runs), but made no change in run time. One interesting feature of Patran is that there is an automatic feature for calculating total model mass. However, the material properties for this model were input in a separate file. In order to have the program automatically make this calculation, it was necessary to construct alternate ("dummy") material properties that had the density embedded within the Patran database itself. These were only used for the mass calculation.

An evaluation was performed of the computer hard drive used for these runs. Most of these runs were done on an external hard drive with a speed of 7200 rpm connected by FireWire. To evaluate if this could be substantially improved by use of a faster drive, an identical run was made on an internal SCSI 15,000 rpm drive. The run time improvement was only 0.6%. This agrees well with what was already assumed, that the execution speed of Patran Thermal is mainly dependent on the processor speed, and not on the speed of accessing the drive.

The execution speed of the solver under a different operating system, Linux, rather than Windows, was also evaluated. Moving the runs to Linux has improved the solution speed by roughly another 20%. An advantage on this system is that the Linux machines are arranged in a cluster, so that a series of jobs can be run on multiple processors simultaneously. This has the potential to reduce the total job time to roughly seven hours of clock time if five processors are used.

Several other modifications have been attempted to improve run time, including changing iteration limits and maximum time step size. The parameters defining the maximum and minimum number of iterations per time step were varied. Several settings were tested, with no appreciable decrease in run time. It may be that these parameters are already optimized, but these parameters will continue to be evaluated. The effect of time step size was also investigated. Patran Thermal does an internal calculation of the time step to be used, depending on the size of the nodes and relative heat load. The maximum time step limit sets the value that Patran Thermal cannot exceed. Runs were performed at both 5 s and 10 s, with a 20% improvement in run time, and no change in results by using 10 s (< 0.1°C over most of the run, and <0.4°C at peak). The results described above are all with a 10 s time step size. Setting this maximum as high as 20 s, however, even for only the portion of the run where the heat load is not greatly varying, gave a change of several degrees in the results, and was thus not acceptable, despite the additional 9% improvement in run time.

Currently, running on a dual processor 2.4 GHz PC, the full limit line prediction takes 2 CPU days, which is one calendar day using both processors in a dedicated mode of operation. Future work will include trials of re-meshing the Patran Thermal model at a lower density to investigate if substantial improvements in run time can be accomplished without sacrificing accuracy. Also,

scripting will be used to allow automated submission of the job to the Linux cluster, which should allow job completion in less than seven hours, even at the current mesh density.

CONCLUSIONS

The method changes made at Langley for aerobraking thermal analysis have been very effective. The separation of models has yielded an analysis process that can respond quickly to a change in scenario, without having to re-run unnecessary analyses. File automation and scripting have improved response time while also decreasing potential for human error. Gains in run time have been achieved of over an order of magnitude. Runs on a multi-node Linux cluster allow for even more gains in the future. With a coarser mesh model, the goal is to be able to run an entire limit line sequence in one to two hours.

ACKNOWLEDGEMENTS

REFERENCES

[1] Dec, John A., and Amundsen, Ruth M., 'A Thermal Analysis Approach for the Mars Odyssey Spacecraft's Solar Array', AIAA 2003 Thermophysics Conference, Orlando, Florida, June 2003.

[2] Amundsen, Ruth M.; Dec, John A.; and George, Ben E.; 'Aeroheating Thermal Model Correlation for Mars Global Surveyor (MGS) Solar Array', *AIAA Journal of Spacecraft and Rockets*, Volume 42, Number 3, May-June 2005, pp 464-473.

[3] MSC/PATRAN User Manual, MacNeal-Schwendler Corporation, Version 2005, December 2004.

[4] Thermal Desktop user manual, Cullimore and Ring Technologies, Inc., Version 4.7, October 2004.