

AEROHEATING MAPPING TO THERMAL MODEL FOR AUTONOMOUS AEROBRAKING CAPABILITY

Ruth M. Amundsen

NASA Langley Research Center, Hampton Virginia

ABSTRACT

Thermal modeling has been performed to evaluate the potential for autonomous aerobraking of a spacecraft in the atmosphere of a planet. As part of this modeling, the aeroheating flux during aerobraking must be applied to the spacecraft solar arrays to evaluate their thermal response. On the Mars Reconnaissance Orbiter (MRO) mission, this was done via two separate thermal models and an extensive suite of mapping scripts. That method has been revised, and the thermal analysis of an aerobraking pass can now be accomplished via a single thermal model, using a new capability in the Thermal Desktop software. This capability, Boundary Condition Mapper, has the ability to input heating flux files that vary with time, position on the solar array, and with the skin temperature. A recently added feature to the Boundary Condition Mapper is that this module can also utilize files that describe the variation of aeroheating over the surface with atmospheric density (rather than time); this is the format of the MRO aeroheating files. This capability has allowed a huge streamlining of the MRO thermal process, simplifying the procedure for importing new aeroheating files and trajectory information. The new process, as well as the quantified time savings, is described.

INTRODUCTION

The Mars Reconnaissance Orbiter (MRO) was a spacecraft that launched in August 12, 2005 and began aerobraking operations in the Martian atmosphere in March 2006. In order to save propellant, MRO used aerobraking to modify the initial orbit at Mars. The spacecraft passed through the atmosphere briefly on each orbit; during each pass the spacecraft was slowed by atmospheric drag, thus lowering the orbit apoapsis. The largest area on the spacecraft, most affected by aeroheating, was the solar arrays. A thermal analysis of the solar arrays was conducted at NASA Langley Research Center to simulate their thermal performance throughout the entire ~6-month period of aerobraking. The original thermal analysis done in 2005 and 2006 utilized two thermal models: a Thermal Desktop[®] (TD)¹ model for orbit simulation and radiation calculations, and a Patran Thermal^{®2} model to accomplish the thermal solution. Use of two thermal models at that time was the most feasible solution, since a method existed from a previous program for applying the aeroheating in Patran, and the orbital solution could be most easily accomplished in Thermal Desktop (no orbital capability existed in Patran). This methodology entailed a great deal of mapping and scripts: mapping the radiation conductors to space from the Thermal Desktop model to the Patran model, as well as mapping the solar and planetary fluxes from the Thermal Desktop model to the Patran model. In addition, any changes made to the solar array necessitated changes to both models. An extensive amount of software was written to accomplish the interpolation of the aeroheating files onto the Patran model, and

much of the software needed to be modified if any changes were made to the meshing of the model. Recently, a new capability was added in Thermal Desktop which streamlined this process and allowed the use of a single model.

The MRO model is used here as the example model in a new body of work to determine the capability of a spacecraft to perform autonomous aerobraking. Thus, instead of a large team of personnel analyzing the spacecraft's orientation and trajectory and determining the proper maneuver to be performed to optimize the aerobraking pass, the spacecraft would carry that capability on-board, and would in normal situations determine its own maneuvers. In order to simplify the thermal process for this work, the use of a single thermal model was evaluated, using the MRO model as the starting point.

AEROBRAKING HEAT LOADS

The heat loads in an aerobraking pass are a function of the atmospheric density, the velocity of the spacecraft, and the heat transfer coefficient, C_H . The incident heat flux is defined using the equation:

$$Q_{incident} = \frac{1}{2} \cdot \rho \cdot V^3 \cdot C_{H_{incident}} \quad (1)$$

where ρ is the atmospheric density at a given point in the trajectory, V is the spacecraft velocity at that point, and C_H is the heat transfer coefficient for incident aeroheating. This relationship is described in more detail in one of the original papers on the MRO aerobraking thermal analysis³. The identical equation holds for $Q_{reflected}$, the heating that is reflected from the array. The heating reflected back into the flow is dependent on the skin temperature of the array. The $C_{H_{reflected}}$ for MRO was calculated at a constant 300K, and thus the true $Q_{reflected}$ is calculated by the equation:

$$Q_{reflected} = Q_{reflected(@300K)} \cdot \frac{T_{wall}}{300} \quad (2)$$

Total aeroheating, Q_{net} , is then the sum of $Q_{incident}$ and $Q_{reflected}$ ($Q_{reflected}$ being negative).

The C_H is normally defined over the expanse of the array, for several different atmospheric densities. C_H is dependent on both the density and the position on the array, as shown by the discrete maps for each density in Figure 1.

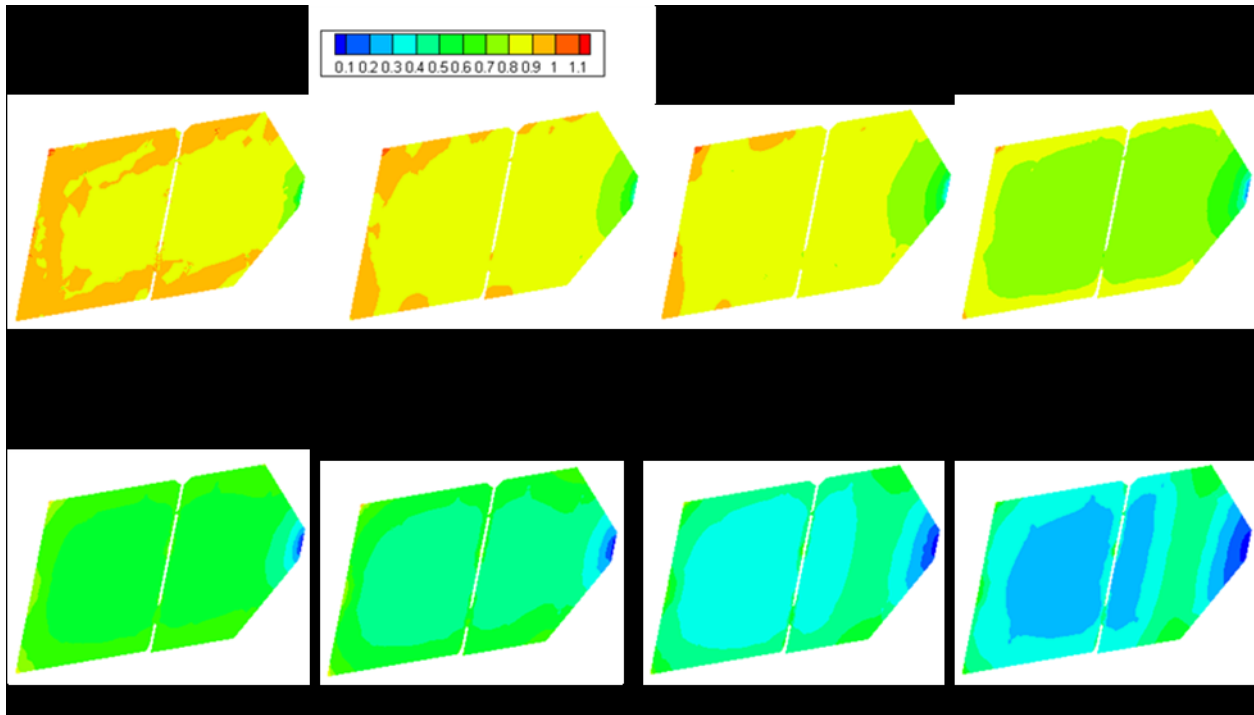


Figure 1. Heat transfer coefficient C_H over the solar array for several atmospheric densities.

To determine the aeroheating over an aerobraking pass, a trajectory file is used which has a very detailed timeline (i.e., an entry roughly once per second) of the atmospheric density and spacecraft velocity during the pass. At each time point, and each point on the solar array, the correct C_H is determined and multiplied by the density and velocity as in equation 1 to determine the heating. In the original modeling effort, this mapping was done via user-developed software, and if the array was re-meshed, the mapping and code required manual modification. This methodology was developed and correlated over three successive Mars missions: Mars Global Surveyor (MGS)⁴, Mars Odyssey⁵, and MRO⁶.

A new capability has been developed in Thermal Desktop since that time, which allows much simpler input of the heating files. The new capability is called Boundary Condition Mapper. In many aeroheating situations, the input file is a file that defines heat flux as a function of time, for many different positions on the geometry, at several skin temperatures. This capability was designed to accept input heat fluxes from a detailed Computational Fluid Dynamics (CFD) mesh. However, for this project, Cullimore & Ring, the developers of Thermal Desktop, added a feature to the Boundary Condition Mapper (BCM) module which allows the input files to be defined as a function of atmospheric density (or any other parameter) rather than time. To utilize density-based files, the user simply adds two lines of logic to the BCM input file, as follows:

```
ADD_MULT AERO_MULTIPLIER
CHANGE_TIMEN AERO_DENSITY
```

where *aero_multiplier* and *aero_density* are register names chosen by the user. These are flag lines, which tell the code to add a multiplier to the interpolated values, and to use a value other than time for interpolation. The registers *aero_multiplier* and *aero_density* represent, respectively, the multiplier to be added to the interpolated values from the input file, and the

variable to be used for the dependent variable (rather than time) for interpolation. As an example, the initial lines of the input file for this model are shown in Figure 2.

```
1 DATA: TEMPERATURE DEPENDENT HEAT FLUX
2 UNITS LENGTH m
3 UNITS TEMPERATURE K
4 UNITS TIME SECONDS
5 UNITS DATA W/m2
6 ADD_MULT AERO_MULTIPLIER
7 CHANGE_TIMEN AERO_DENSITY
8 PRELOGICC PRE LOGIC TO CALCULATE DENSITY & MULTIPLIER for CH Map
9 PRELOGICC multiplier is 1/2 rho v3
10 PRELOGICC density is in kg/km3, velocity is in km/s
11 PRELOGICC multiplied out (N=kg-m/s^2), Ch being unitless, ends up as W/m2
12 PRELOGICC Ch refl is negative, so added it will decrease Ch incident
13 PRELOGIC CALL D1D1DA( TIMEN, M55J_HOT.A1, M55J_HOT.A2, AERO_DENSITY )
14 PRELOGIC CALL D1D1DA( TIMEN, M55J_HOT.A1, M55J_HOT.A3, VELOCITY )
15 PRELOGIC AERO_MULTIPLIER = .5 * AERO_DENSITY * VELOCITY * VELOCITY *
VELOCITY
16 TEMPERATURES 1
17 300.
18 NODE 1 1.610000014E+000 0.000000000E+000 -1.560000032E-001
19 NODE 2 1.628000021E+000 0.000000000E+000 -2.249999940E-001
```

Figure 2. Initial lines for density-dependent BCM input file.

The importance of being able to use the density files directly has to do with both file size and ease of use. First, there are nine files of C_H mapping (nine different densities). Each one is 80 MB in size. If each of those files were to be translated to heating, and a heating file as a function of position created for each of the roughly 1000 time points in a normal trajectory file, the resulting size for each pass would be 80 GB. Since there are roughly 400 passes, each with a different trajectory, the final data size for the input heating would be 32 TB. Thus, it is much simpler to maintain the files as density-based, and use the same set of C_H files for each pass (total data file size 720 MB). Second, in terms of usability, the BCM is applied directly to the Thermal Desktop geometry. There is no need for user-written code to accomplish the mapping. Thus, if the Thermal Desktop surface is remeshed, the new mapping will be handled automatically by the Thermal Desktop software. The only user-written code is the three lines (lines 13-15 in Figure 2) that define how heating will be calculated. These lines perform the interpolation on density and velocity from the input trajectory file, and calculate the aeroheating using equation 1.

TRAJECTORY FILE INPUT

The trajectory file for each pass is a text file giving atmospheric density and spacecraft velocity at each time point (roughly each second) as the spacecraft passes through the planetary atmosphere. This file is read into the Thermal Desktop model using a block of code in The Logic Manager, as shown in Figure 3. Line letters are given only to clarify this description. Because each pass can have a trajectory file with a different number of points, the code must be able to successfully read in arbitrary length files. Line A defines the name of the file to be used, using a character array pre-defined in the model. Line E skips the first line of the file because that is always a header

line with column labels. Lines F through M form a loop which reads through the file, where line K loads each value of time, density and velocity into the appropriate arrays. Lines P, R and S set the beginning and end times of the thermal solution to be the first and last time points from the trajectory file, respectively. Lines T through V set the size of each array based on the trajectory file read in for this case.

```

A      CALL USRFIL2(num_traj_file,AERO_FLUX.UCA(1),'OLD',iostat)
B      IF (iostat.eq.1) GOTO 2
C      eitest=0
D C    Read to skip header.
E      read(num_traj_file,*)
F 10   itest=itest+1
G C
H C    Read time, density, velocity
I C      Time is in seconds (0=periapsis), density is in kg/km3, velocity is in km/s
J C      These will be used to build the aero_multiplier that will multiply Ch to create Qflux
K read (num_traj_file,*,end=20) M55J_HOT.A(1+itest),M55J_HOT.A(2+itest),M55J_HOT.A(3+itest)
L      goto 10
M 20   continue
N      close (num_traj_file)
O      ITRAJ=itest-1
P      t_begin_pass=M55J_HOT.A(1+1)
Q C Set start and end times based on the input file trajectory file
R      TIMEO = t_begin_pass
S      TIMEND = M55J_HOT.A(1+ITRAJ)
T      M55J_HOT.NA1 = ITRAJ
U      M55J_HOT.NA2 = ITRAJ
V      M55J_HOT.NA3 = ITRAJ
W 2    print *,'error opening DSMC files!'

```

Figure 3. Code block for reading in text trajectory file.

IMPLEMENTATION IN THERMAL DESKTOP

The aeroheating is applied using the BCM command on the Thermal > FD/FEM Network menu. This command brings up a form to specify the formatted input file, as shown in Figure 4. Inputs on this form specify a register to be used to enable and disable the application of this heating, the submodel name for the created arrays and logic, tolerances for how far the input mesh can be from the TD nodes, and a specification for what surfaces to apply the heating to, either by selecting directly, or applying to an entire AutoCAD group. When a case is run, two files will be created containing the logic to apply this heat load, .hrl and .hra files. The .hrl file will contain all the initial logic as in Figure 2, as well as the commands for interpolating and applying the heating to each node. The .hra file will contain the aeroheating arrays necessary to accomplish the interpolation. Once the .hrl and .hra files have been created, the BCM can be disabled if desired, and those files called with INSERT statements in the case set manager (in the 'OTHER' section). This avoids the time in re-mapping the input mesh to the nodes for each run. If either the input mesh or the TD mesh change, the BCM can be re-activated to perform the mapping as desired. Another advantage to calling the .hrl and .hra files separately is that they can be edited if necessary. For the $CH_{\text{reflected}}$ component, the heating was multiplied by the factor in equation 2, which depends on the skin temperature. Thus, after it was created, the lines in the .hrl file for

$CH_{\text{reflected}}$ were edited to add the dependence on skin temperature. The lines necessary to apply aeroheating for a single node are shown as a sample in Figure 5, with the line to account for skin temperature added for each node shown in bold. The lines are added by an automated search-and-replace, not manual editing.

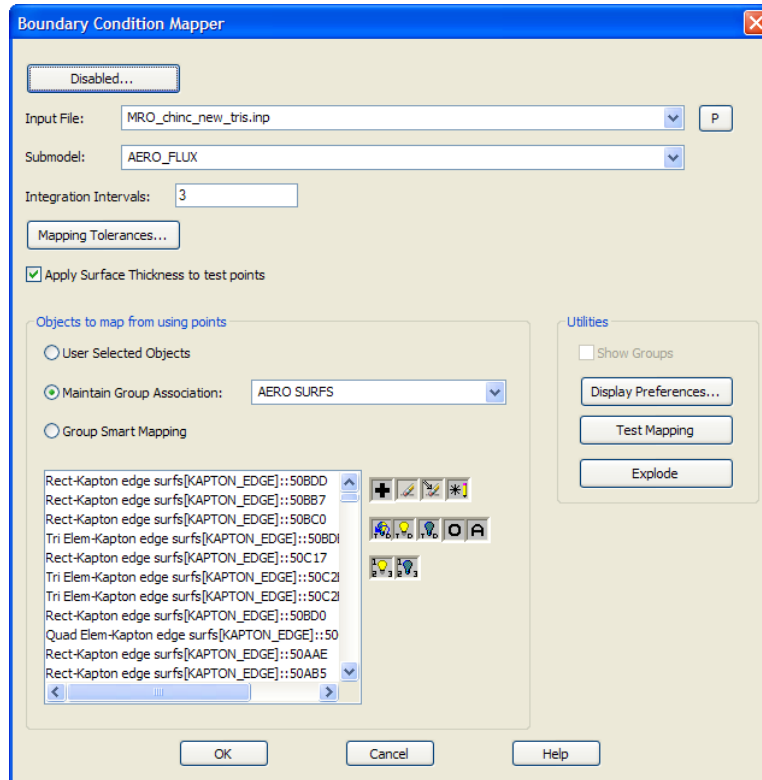


Figure 4. BCM input form.

```
CALL D2DEG1(KAPTON_EDGE.T1,AERO_DENSITY,A1,T__TD)
T__TD = T__TD * (KAPTON_EDGE.T1 + 273.15)/300.0
KAPTON_EDGE.Q1 = KAPTON_EDGE.Q1 + T__TD * 0.00151775 * AERO_MULTIPLIER
```

Figure 5. Modified lines for temperature dependence of $CH_{\text{reflected}}$ (bold line added).

The C_H files were provided in TecPlot format. A software program, Map2CFD, had already been developed at NASA Langley to translate files from TecPlot format to the required BCM input file format. After the files were run through Map2CFD to produce a BCM-formatted input file, the initial lines described earlier were added, and the editing to incorporate skin temperature dependence was done.

The Thermal Desktop model incorporated all thermal mass on the solar array. The radiative effect of the spacecraft was included, but the temperature of the spacecraft was assumed rather than calculated, the same as was done in the original modeling process. The array itself is described in detail in other papers^{3, 7} but a brief description is given here and shown in Figure 6. The array consisted of an aluminum honeycomb core with facesheets of extremely thin graphite composite. The solar cells were mounted on one of the facesheets. During aerobraking, the array was set with one graphite facesheet (called the ‘hot side’) into the prevailing wind, and the

solar cells on the back face, away from the wind. The names and locations of the four flight sensors on the array are shown.

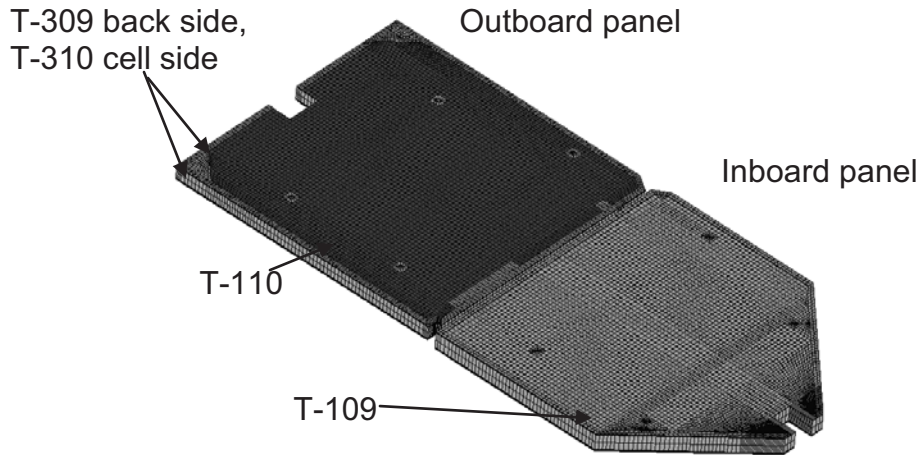


Figure 6. Configuration of MRO solar array and sensors (thickness of panel exaggerated).

RESULTS

CORRELATION TO MRO IN-CRUISE EVENTS

On MRO, there were several planned changes in orientation performed during the cruise to Mars. These were known as ‘calibration events’ as they were used to calibrate and check the guidance and pointing of the spacecraft. These events were extremely useful to the thermal team in correlating the thermal model of the solar array prior to reaching Mars orbit. The thermal model correlation using the previous method of two separate thermal models is documented in other publications.⁷ This correlation was performed with the current single-model approach as a verification of the model and approach. The in-cruise correlation addresses only the model mass, conduction and radiation; since no aeroheating is present, that portion of the analysis is not verified. Correlation was at least as good as in the earlier (dual-model) work, even though the current correlation was done by one person over two weeks, and the previous work was done by three thermal analysts over a period of several months.

Plots of the correlations to two different in-cruise events are shown in Figure 7 and Figure 8. The Trajectory Correction Maneuver 1 (TCM1) maneuver was an actual trajectory modification, and had only one prolonged change in angle. The Stanford calibration event had a sequence of abrupt changes in orientation. These two are shown as samples; all five of the in-cruise events were run. In each case, the only input to the thermal model was the actual flight data on orientation with respect to the sun and Earth; the output is sensor temperatures which are then compared to the flight data. As shown, the model agreement to the flight data is extremely good.

The results from the correlation of all five in-cruise events is shown in Table 1, as compared to the original correlation. The root-mean-square (RMS) error is computed by combining the difference between the model and the sensor data at each time point as an RMS value, and then taking the RMS of all four sensors combined. The peak error was the most critical, since the

peak temperature of the array was most important in flight. The correlation is very good (3°C), and is slightly better than the original modeling effort, despite the lower level of effort.

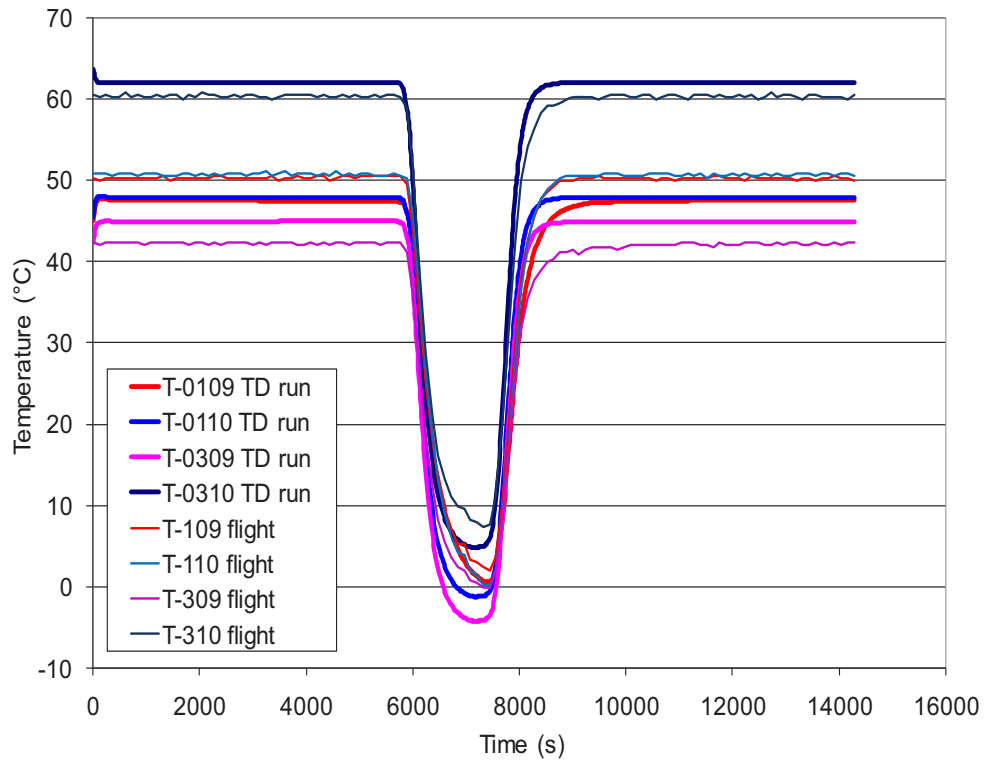


Figure 7. Correlation to Trajectory Correction Maneuver 1 (TCM1) flight temperatures.

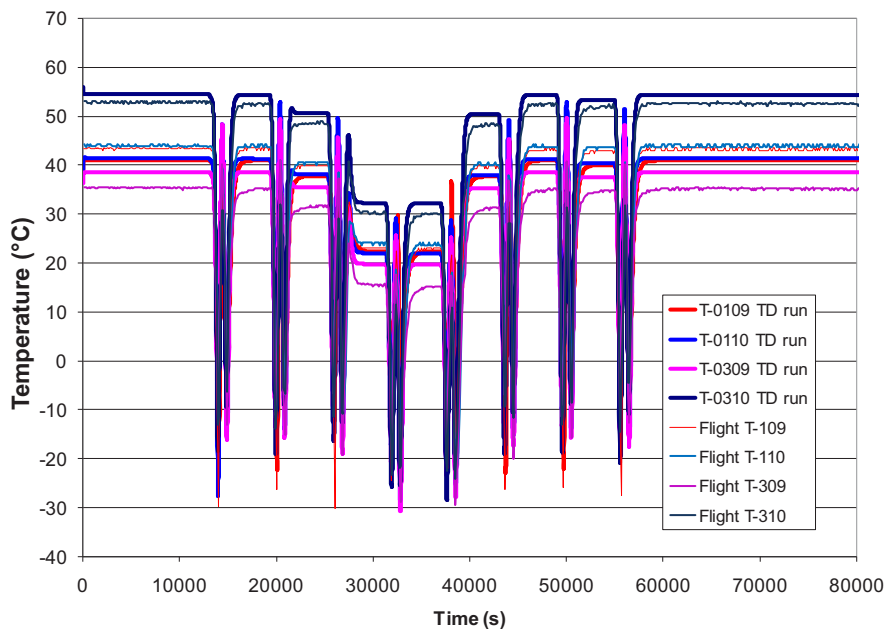


Figure 8. Correlation to Stanford calibration maneuver flight temperatures.

Table 1. Correlation to In-Cruise Events

In-Cruise Event	4-sensor average RMS error (°C)	Error in peak predict (°C)	Original model RMS error (°C)
TCM1	2.5	2.9	2.2
TCM2	2.8	4.2	4
Thruster calibration 1	2.7	3.4	1.3
Thruster calibration 2	5.3	2.4	4.7
Stanford calibration	3.7	1.7	5.4
Average	3.4	2.9	3.5

LIMIT LINE DEVELOPMENT

One use of the thermal model for MRO was to develop “limit lines”: lines that defined when the heating in a particular pass would be high enough to drive the solar array over its maximum temperature limit. This was done by running a series of four different heating levels on each of 10 different passes. In the original modeling effort, this was done through a series of scripts. The model first had to be run around the entire orbit, to generate the initial temperatures that would exist before the pass. Then for each pass, four different heating levels were run, and the limiting heating value was interpolated from those four. In the original modeling effort, the initial pass runs took over 4 days of computer time to complete, and the series of 40 limit line runs took over 20 hours (after an extensive effort to decrease run times). These runs were performed on a multi-node Linux cluster to limit the clock time necessary (since cases could be run in parallel). Then all the results were drawn back to a single location to interpolate the limit lines. In this latest modeling effort, not only was all the effort of scripting and mapping between models avoided, but all initial pass runs could be completed in less than 7 hours, and all limit line runs were completed in 1.5 hours. Each of these could be accomplished from within the Thermal Desktop model, with a single button click. Total run time was 8.5 hours versus the original time of over 116 hours, which is better by a factor of 14.

The method for accomplishing all four limit line cases per pass in a single case set was to embed the instructions for running all four cases in the operations block of the case set. These lines simply reset the trajectory file name and the save file name and performed the next run (lines for the first two runs are shown in Figure 9 as a sample). Then all cases for the initial temperatures and limit lines could be run by clicking “Run Selected Cases” as in Figure 10.

```

C Do _1 run *****
CALL LOADT
CALL TDPREBL $ Logic in Subroutine Data
BUILD ALL
CALL DPCS $ Enables NODMAP in model browser
CALL USRFIL(nuser3,'p005_limit_line_1.us3')
AERO_FLUX.UCA(1)='C:\my_docs\AAB\limit_line_runs\traj_files\p005_1.txt'
CALL TDPOSTBL $ Logic in Subroutine Data
CALL FWDBCK
CALL TDHTOT $ Output Heater Summary
CALL TDPOSTSL $ logic in Subroutine Data
C
C Now set up _2 run *****
C Go back to initial temps, set user and save file names, set trajectory file
C and run
CALL LOADT
CALL USRFIL(nuser1,'p005_limit_line_2.us1')
CALL USRFIL(nuser2,'p005_limit_line_2.us2')
CALL USRFIL(nuser3,'p005_limit_line_2.us3')
CALL CHGSAVE('p005_limit_line_2.sav')
CALL TDPREBL $ Logic in Subroutine Data
BUILD ALL
CALL DPCS $ Enables NODMAP in model browser
AERO_FLUX.UCA(1)='C:\my_docs\AAB\limit_line_runs\traj_files\p005_2.txt'
CALL TDPOSTBL $ Logic in Subroutine Data
CALL FWDBCK
CALL TDHTOT $ Output Heater Summary
CALL TDPOSTSL $ logic in Subroutine Data

```

Figure 9. Limit line case setup.

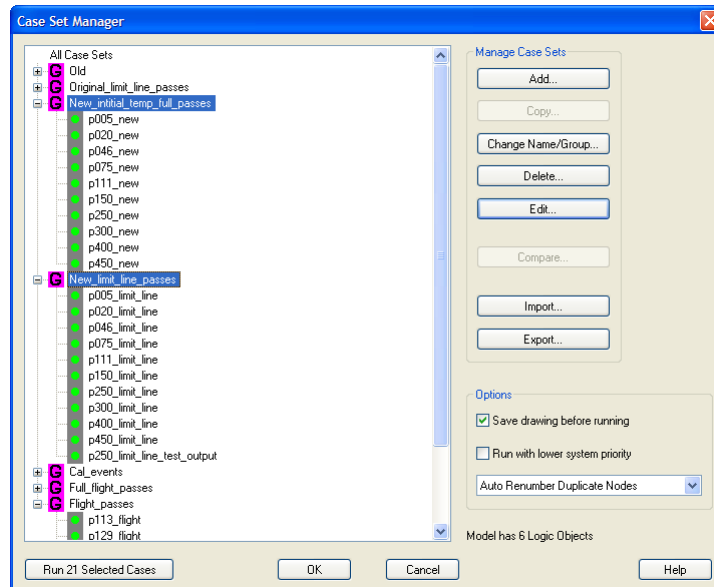


Figure 10. Screen for running multiple cases.

The aerobraking heat flux applied during one of the limit line runs, and the resulting temperatures on the hot side facesheet of the array, are shown for a sample pass in Figure 11 and Figure 12.

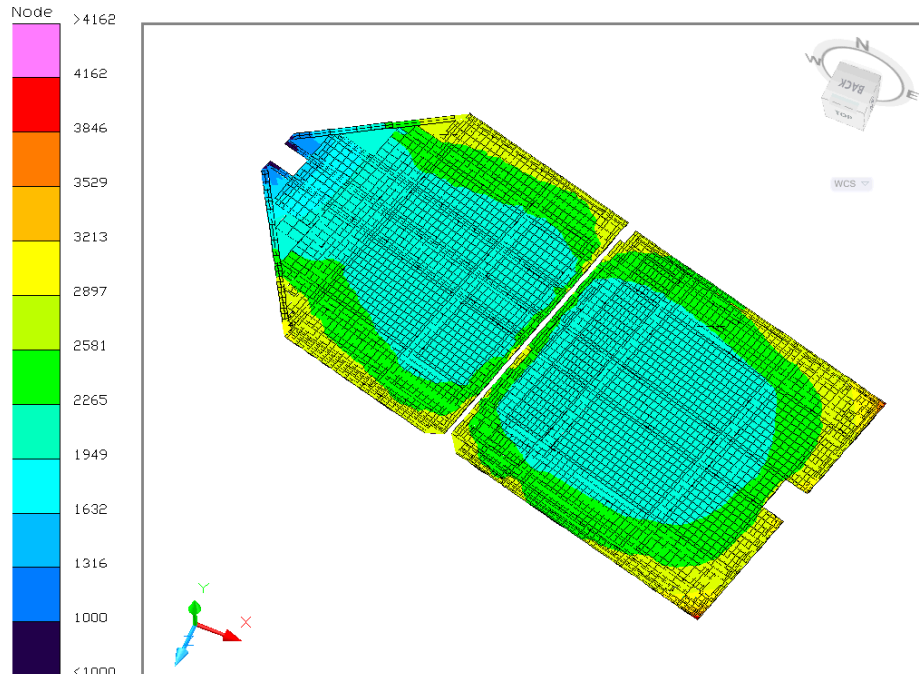


Figure 11. Maximum heat flux applied (W/m^2) for pass 250, heating level 3.

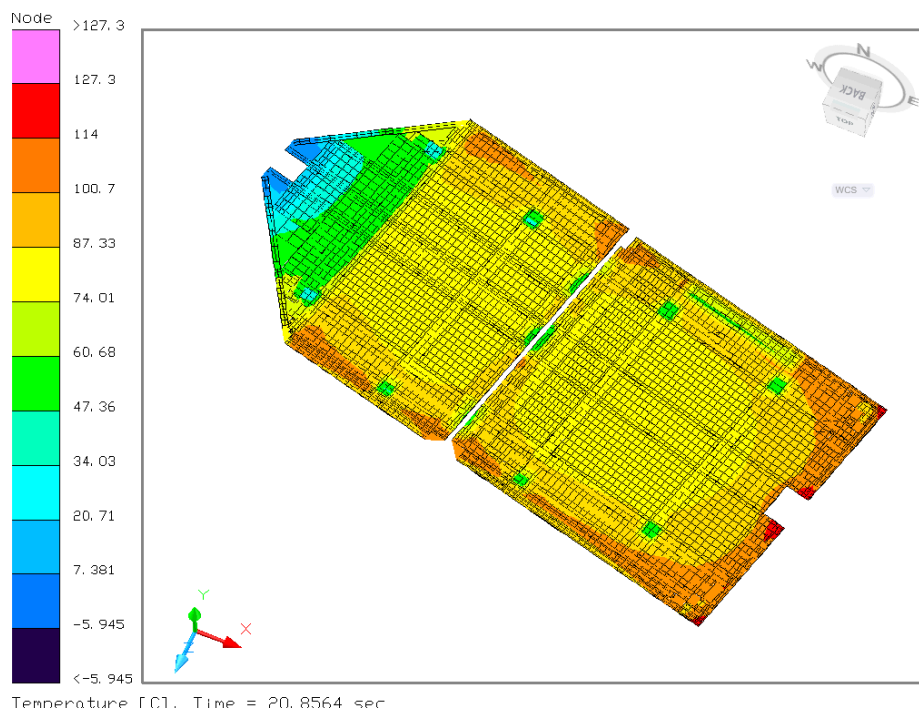


Figure 12. Maximum temperatures for limit line pass 250, heating level 3, at 20 s ($^{\circ}\text{C}$).

CORRELATION TO MRO FLIGHT AEROBRAKING PASSES

The model was also used to analyze the flight passes that occurred during the MRO mission. Run time for a single flight pass, using this new method, was only six minutes (five times faster

than the old method). The correlation to flight was very good, despite the fact that very little adjustment of the model was done. In the original modeling effort, roughly six months of time was spent in correlation, while only a few weeks was used for this new approach. The main actions taken during correlation were to add an accommodation coefficient to the aeroheating, adjust the T-109 thermal sensor modeled position, modify the specularity of the spacecraft MLI, and adjust the contact of the array honeycomb core to the facesheets. Six flight passes, encompassing a wide range of orbit periods and aeroheating rates, were selected for correlation. An example of the temperatures results from the hottest pass, p262, is shown in Figure 13. The correlation to the flight data is shown in Figure 14, where the markers are flight data, and the smooth lines are the model predictions. The correlation is excellent except for sensor T-109, which starts the transient warmer than the flight data. It appears that sensor T-109 was in a location in the model where a reflection from the spacecraft maintained a warmer temperature than was seen in flight. Since the spacecraft was not modeled in great detail, this would be difficult to correct, but is not a substantial error. The final correlation to these six passes is shown in Table 2. These values are equivalent to what was achieved after months of work in the original modeling effort: 0.7°C as the average error on the peak sensor value, and 6.3°C as an RMS error computer over the whole time period of the pass. In the original modeling effort, these values were 1°C for peak error, and 5.2°C for RMS error. The peak error is the most important, since that is used to compute the survival of the array and thus the limit lines for maneuver planning.

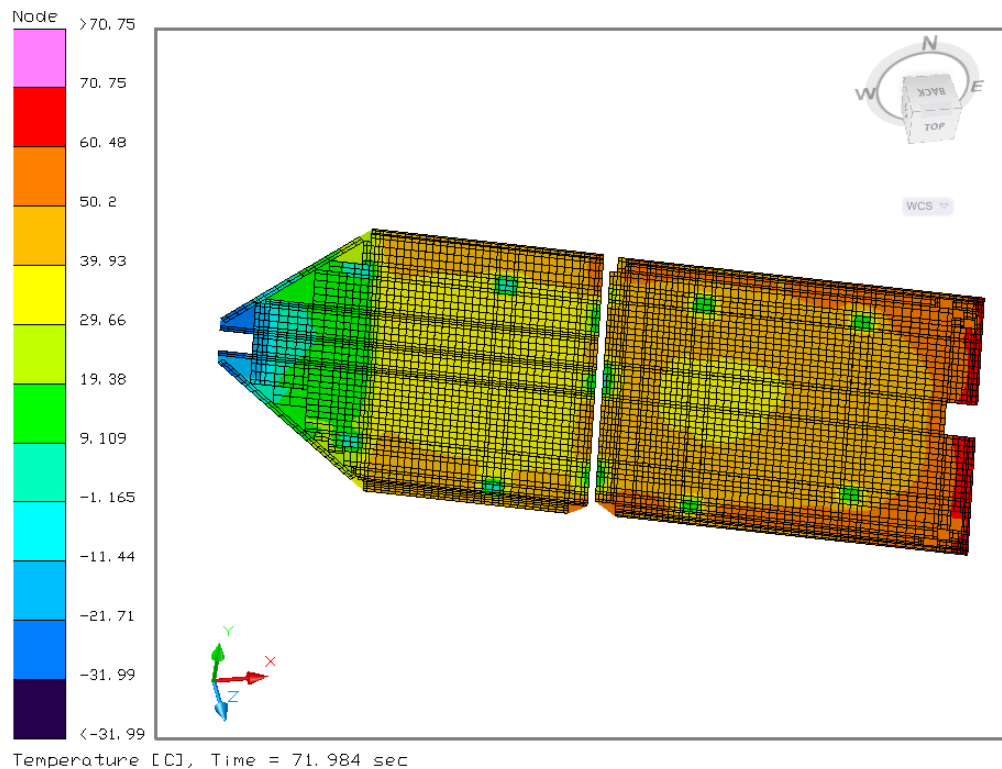


Figure 13. Maximum temperature during flight pass 262 (°C).

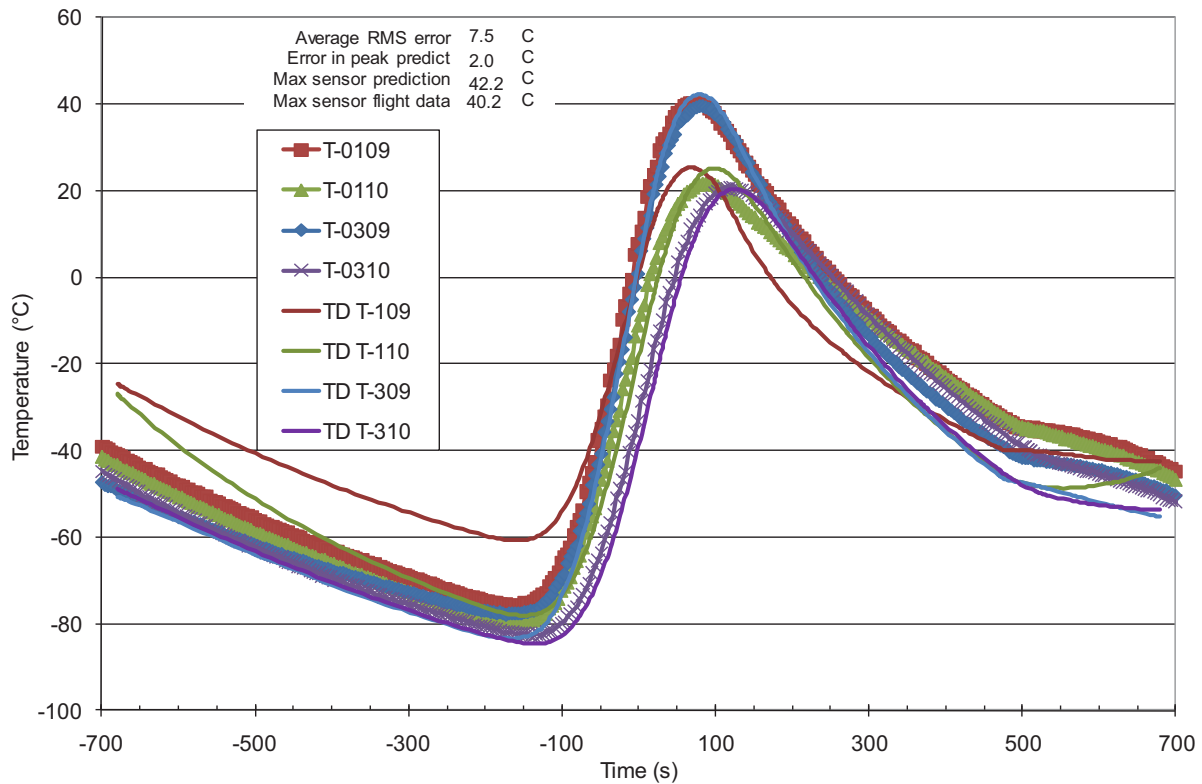


Figure 14. Correlation to flight for pass 262.

Table 2. Final Correlation to Flight MRO Aerobraking Pass Data

Pass	Peak errors (°C)					Overall peak error	RMS errors (°C)				
	T-109	T-110	T-309	T-310 (cell side)	Delta		T-109	T-110	T-309	T-310 (cell side)	Avg
p113	-5.6	5.8	4.4	3.0	0.5	3.5	7.5	5.0	4.4	4.7	5.4
p129	-1.2	1.9	-2.2	-0.5	-1.0	-1.5	4.3	5.7	5.7	5.8	5.4
p217	-7.9	7.5	4.6	2.8	0.6	4.3	9.9	4.8	3.6	3.6	5.5
p262	-14.9	3.5	2.7	-0.4	2.4	2.0	12.1	8.5	4.0	5.3	7.5
p291	-7.9	-3.5	-2.0	-10.2	4.7	-2.0	6.9	6.7	7.7	7.6	7.2
p334	-1.3	7.4	-4.2	-3.0	-0.6	-2.0	4.6	4.7	8.7	8.6	6.6
Overall avg						0.7					6.3

CONCLUSIONS

The methodology for thermal analysis of an aerobraking pass has been substantially improved. The use of a single model for the thermal analysis was validated as feasible. The advantages are a huge savings of time in model development and updates, since only one model need be developed and kept updated. All work with mapping scripts, mapping radiation results, and manual work in re-coding for a re-meshed model is eliminated. All optical properties are in the model, which makes modification easier, rather than the previous method of maintaining certain of the optical property values actually in the user-developed code. The Thermal Desktop Boundary Condition Mapper use of density-based C_H files was validated. The density-based files can be used directly, without modification, and with no manual work of re-mapping if the model is re-meshed. All limitations on model numbering are eliminated, since there is no user-developed code that references the node numbers of specific nodes. Run times were improved by more than an order of magnitude over the old method. The quality of model correlation to flight data in the new method matched or exceeded the old method, although much less time was spent on it. Overall, the new method is more streamlined, involves much less manual work, is less prone to error, and allows much more rapid analysis and response to changes.

ACKNOWLEDGEMENTS

The assistance of Cullimore & Ring Technologies in adding the capability for the Boundary Condition Mapper to accept density-based files is gratefully acknowledged. This work would have been much more time-consuming without that addition. The work of John Dec and Joe Gasbarre in the original effort as well as in helping to develop this new method is also acknowledged with thanks. The assistance of Joe Del Corso in using Map2CFD was invaluable.

NOMENCLATURE, ACRONYMS, ABBREVIATIONS

BCM	Boundary Condition Mapper
C_H	Heat transfer coefficient
CFD	Computational Fluid Dynamics
FD/FEM	Finite Difference / Finite Element Mesh
MGS	Mars Global Surveyor
MRO	Mars Reconnaissance Orbiter
RMS	Root-Mean-Square
TCM	Trajectory Correction Maneuver
TD	Thermal Desktop [®]

REFERENCES

- ¹ Thermal Desktop User Manual, Cullimore and Ring Technologies, Inc., Version 5.3, January 2010.
- ² MSC/PATRAN User Manual, MacNeal-Schwendler Corporation, Version 2010, February 2010.

³ Dec, John A., Gasbarre, Joseph F., and Amundsen, Ruth M., "Thermal Modeling of the Mars Reconnaissance Orbiter's Solar Panel and Instruments During Aerobraking," 07ICES-64, 37th International Conference On Environmental Systems, Chicago, Illinois, July 2007.

⁴ 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.

⁵ 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.

⁶ Amundsen, Ruth M., Gasbarre, Joseph F., Dec, John A., "Thermal Analysis Methods for Aerobraking Heating," 16th Thermal & Fluid Analysis Workshop (TFAWS 05), Orlando, Florida, August 8 - August 12, 2005.

⁷ Amundsen, R. M., Dec, J. A., Gasbarre, J. F., "Thermal Model Correlation of the Mars Reconnaissance Orbiter", 07ICES-17, 37th International Conference On Environmental Systems, Chicago, Illinois, July 2007.