DEVELOPMENT AND IMPLEMENTATION OF EFFICIENCY-IMPROVING ANALYSIS METHODS FOR THE SAGE III ON ISS THERMAL MODEL

Kaitlin Liles, Ruth Amundsen, Warren Davis, and Salvatore Scola
NASA Langley Research Center (LaRC)

Steven Tobin
Northrop Grumman

Shawn McLeod
Analytical Mechanics Associates (AMA)

Sergio Mannu and Corrado Guglielmo
Thales Alenia Space-Italy (TAS-I)

Timothy Moeller
Georgia Institute of Technology/National Institute of Aerospace (GT/NIA)

ABSTRACT

The Stratospheric Aerosol and Gas Experiment III (SAGE III) instrument is the fifth in a series of instruments developed for monitoring aerosols and gaseous constituents in the stratosphere and troposphere. SAGE III will be delivered to the International Space Station (ISS) via the SpaceX Dragon vehicle in 2015. A detailed thermal model of the SAGE III payload has been developed in CRTech Thermal Desktop® (TD). Several novel methods have been implemented to facilitate efficient payload-level thermal analysis, including the use of TD assemblies to move payloads from the Dragon trunk to the Enhanced Operational Transfer Platform (EOTP) to its final home on the Expedite the Processing of Experiments to Space Station (ExPRESS) Logistics Carrier (ELC)-4, implementation of a design of experiments (DoE) methodology to determine the worst-case orbits for SAGE III while on ISS, incorporation of older models in varying unit sets, ability to change units easily (including hard-coded logic blocks), case-based logic to facilitate activating heaters and active elements for varying scenarios within a single model, incorporation of several coordinate frames to easily map to structural models with differing geometries and locations, and streamlined results processing using an Excel-based text file plotter developed in-house at LaRC. This document presents an overview of the SAGE III thermal model and describes the development and implementation of these efficiency-improving analysis methods.

INTRODUCTION

SAGE III is the fifth in a series of instruments developed for monitoring aerosols and gaseous constituents in the stratosphere and troposphere. SAGE III measures solar occultation, as shown in Figure 1 and lunar occultation in a similar fashion. SAGE III also measures the scattering of solar radiation in the Earth's atmosphere (called limb scattering [LS]) as shown in Figure 2. These scientific measurements provide the basis for the analysis of five of the nine critical constituents identified in the U.S. National Plan for Stratospheric Monitoring. These five atmospheric components include the profiles of aerosols, ozone (O_3), nitrogen dioxide (NO_2), water vapor (H_2O), and air density using oxygen (O_2).

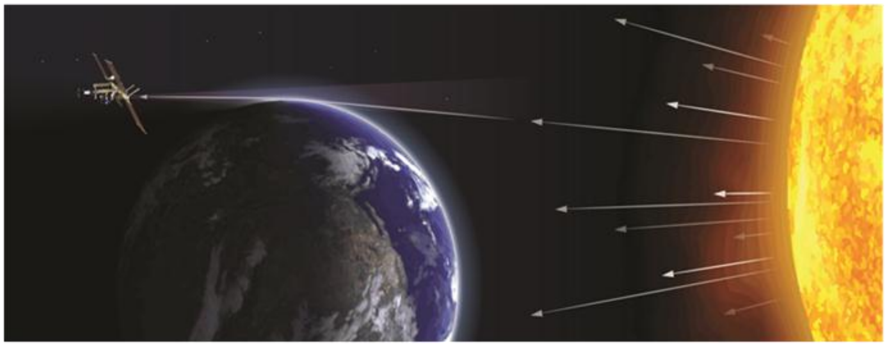


Figure 1. Solar Occultation Measurement

Figure 2. Limb Scattering Technique

The SAGE III payload will be mounted on ISS for an operational lifetime of 3 years. Many types of thermal analysis are required for this payload. These include over 120 orbit configurations of the payload thermal model in the Dragon capsule, runs to validate the payload during transfer and in the ISS mounted configuration, runs to determine the worst case orbital parameters for this payload and this location on ISS, standard runs to evaluate the payload thermal behavior during test and in all operational phases, and mapping of thermal results to a structural model to evaluate thermally-induced stress and deflection. In order to expedite this large amount of thermal analysis, many methods were developed to make this thermal model efficient and effective.

MODEL DEVELOPMENT

The model was developed in TD version 5.5 (http://www.crtech.com/)[1].

Figure 3 and Figure 4 show the SAGE III Instrument Payload (IP) and the Nadir Viewing Platform (NVP), respectively. On ISS the IP is mounted to the NVP to allow a nadir viewing direction for the payload, which is required for the heritage SAGE III Instrument Assembly (IA) to collect science data. The thermal model of SAGE III (IP and NVP mounted together) is shown in Figure 5 and its location on the ISS is shown in Figure 6. SAGEIII will be mounted on the port-facing side of the ELC-4.

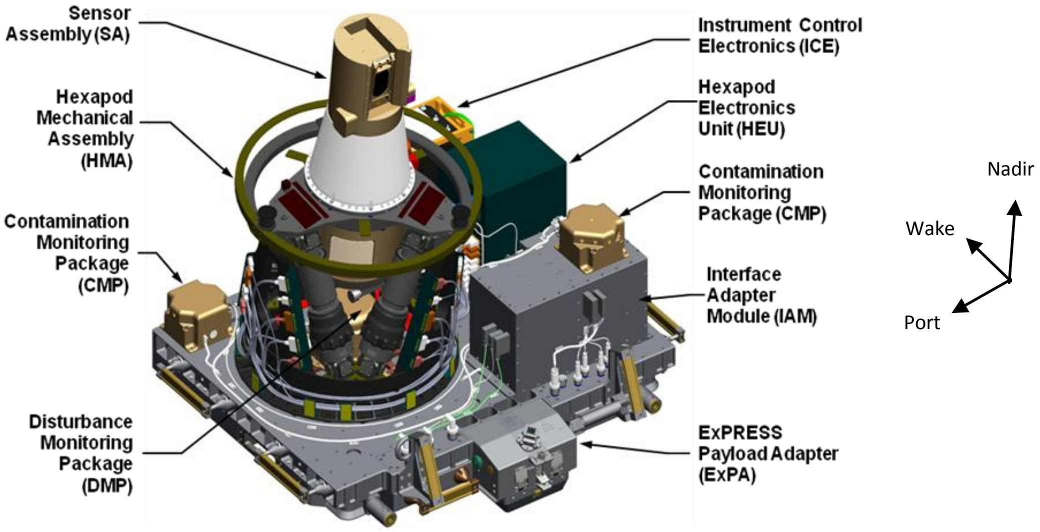


Figure 3. SAGE III Instrument Payload (IP)

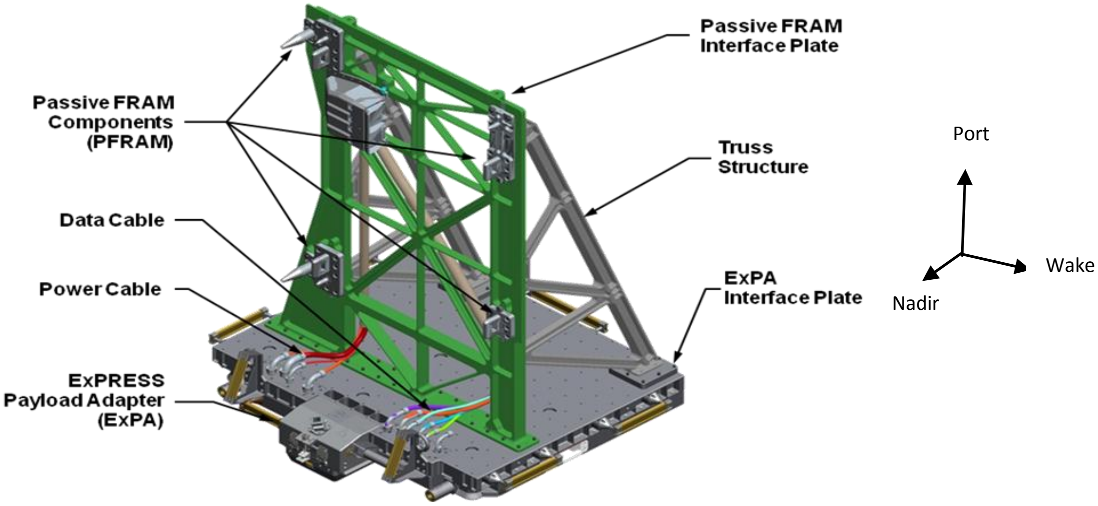


Figure 4. SAGE III Nadir Viewing Platform (NVP)

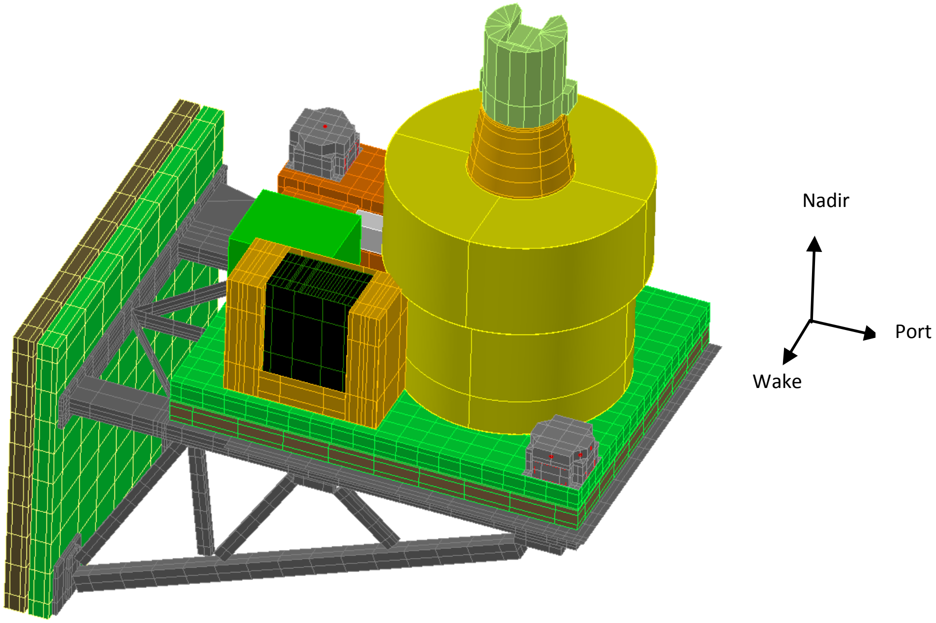


Figure 5. SAGE III Thermal Desktop (TD) model

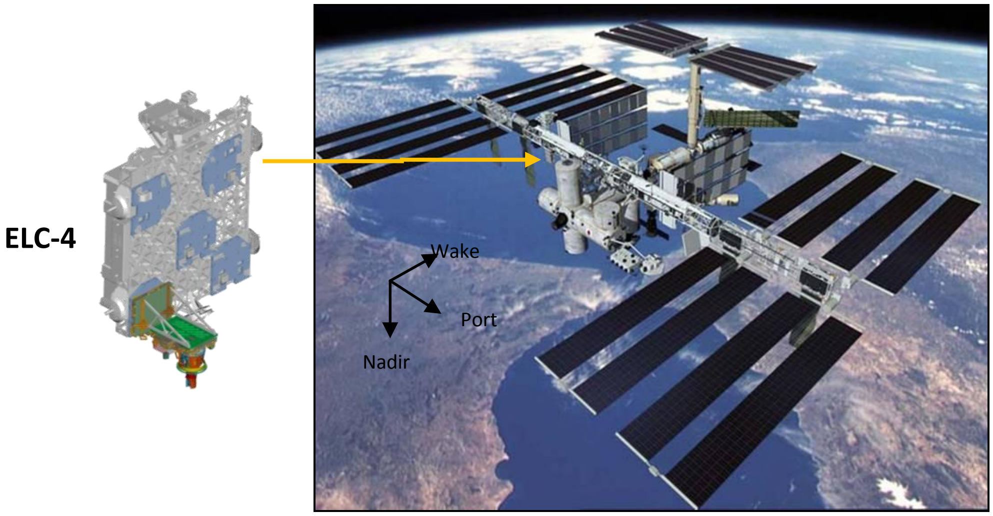


Figure 6. SAGE III Location on ISS

The SAGE III thermal team consists of several engineers, so the model is housed on a share drive which is accessible by all team members. Throughout the development of the model, a versioning approach has been implemented and a detailed change log has been maintained. The versioning approach is as follows: major version changes (changes that impact node numbers) are signified by an increase in the number and minor version changes are signified by an increase in the letter. The current version of the model is v39b and was released on 6/25/2013.

The early versions of the model had simplified representations of each subsystem and detail has been added over time. The detailed Sensor Assembly (SA) model was initially developed at Ball Aerospace and Technologies Corporation (BATC) and has been modified by the LaRC thermal team as needed. The detailed models of the Hexapod Mechanical Assembly (HMA) and Hexapod Electronics Unit (HEU) were developed at Thales Alenia Space-Italy (TAS-I). All other subsystem model development, including development of a detailed NVP model, a simplified DMP model, and board-level models for the IAM, CMP, and ICE subsystems was performed at LaRC. The SAGE III system-level model also includes the ISS model (v6r4) provided by Johnson Space Center (JSC) and the low-fidelity Dragon trunk model (v1r1) provided by SpaceX. Model integration was performed at LaRC. Approximate node counts for various parts of the model are shown in Table 1.

Table 1. Node Counts for the SAGE III System-Level Thermal Model

Model Segment	Number of Nodes
IP	8600
NVP	1800
ISS (including ELC-4 and EOTP)	3950
Dragon	45

USE OF ASSEMBLIES FOR CHANGING CONFIGURATION/LOCATION

The SAGE III payload will have several configurations while in orbit. First will be the free-flight portion in the Dragon capsule, where the NVP and IP are mounted separately within Dragon. Next, each payload will be removed from Dragon and placed on the Enhanced Operational Transfer Platform (EOTP). The EOTP is mounted on the robotic arm, Special Purpose Dexterous Manipulator (SPDM), and together they will move down the ISS backbone to the ELC-4 location, where the payloads will be removed, with the NVP installed on ELC-4 and the IP mounted on the NVP. It is desirable to analyze all payload configurations and cases with a single model, thus eliminating the need to import the payload model into various other models, or synchronize changes to a set of duplicate payload models. However, since the SAGE III payload is in very different configurations in different locations, being mounted separately on Dragon, back-to-back on EOTP, and assembled together on the ISS ELC-4, that presents a challenge.

This suite of analyses was accomplished efficiently with the use of articulators (a TD function that may be used to group sets of surfaces into an assembly or to define motion of a set of surfaces) and incorporating case-based logic for their placement. Registers were created to define the desired location of the SAGE III IP and NVP, named flag_SAGE_MOV and flag_NVP_MOV, respectively. The values for these registers define the location for the payload, as follows: 0 = On ELC4, 1 = In Dragon, 2 = outside near dragon on EOTP, 3 = at w-5 on EOTP, 4 = at w-2 on EOTP, 5 = on EOTP just below ELC-4. The latter four values indicate the different positions of EOTP along the ISS backbone that have been selected as representative for analysis during the transfer. These locations were chosen using a story board of the transfer sequence that was provided by the ISS program.

Within the SAGE III TD model all of the SAGE IP was placed on one articulator, and the entire NVP was placed on another articulator, as shown in Figure 7. The movement and rotation of each of these articulators was accomplished with registers for each axis translation and rotation, e.g., SAGE_IP_trans_x and NVP_DRG_trans_X. These separate articulators are used to control the payload movements when they are placed separately on ELC-4 and Dragon, e.g. with logic like:

```
(Flag_Sage_Mov == 0 ) ? 0 : ((Flag_Sage_Mov == 1)||(Flag_Sage_Mov == 2)||(Flag_Sage_Mov
== 3)||(Flag_Sage_Mov == 4)||(Flag_Sage_Mov == 5)) ? 33.2 : 0
```

which sets the value of SAGE_IP_trans_x to 0 for the ELC-4 case, and 33.2 in all other cases.

In addition, when the MOV flag values are 1, for the Dragon site, a register Dragon_solo is used to indicate that only the Dragon and payload submodels should be built, and not the rest of the ISS. This avoids solving for ISS radiation and temperature when running the SAGE III payload within Dragon.

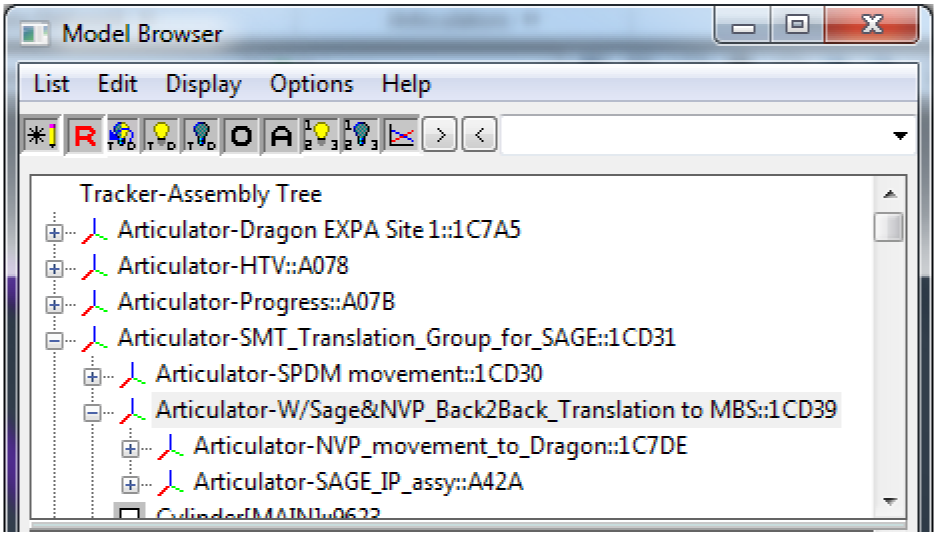
These two payload articulators were placed on a higher level articulator which handles their placement together in their back-to-back orientation on the EOTP whose translations and rotations were controlled by named registers, e.g., EOTP_Grp_Rot_X. Logic was used to define the position of these registers based on the values of the MOV flags, e.g.:

```
((Flag_Sage_mov == 0)&& (Flag_nvp_mov == 0 )) ? 0 : ((Flag_Sage_mov == 1) &&
(Flag_nvp_mov == 1)) ? 0 : ((Flag_Sage_mov == 2) && (Flag_nvp_mov == 2)) ? 0
:((Flag_Sage_mov == 3) && (Flag_nvp_mov == 3)) ? 60 :((Flag_Sage_mov == 4) &&
(Flag_nvp_mov == 4)) ? 60 : ((Flag_Sage_mov == 5) && (Flag_nvp_mov == 5)) ? 60 : 0
```

This example logic sets the value of EOTP_Grp_Rot_X to 0 if the MOV flags are 0-2, and to 60 if the flags are 3-5. Some of the register logic is more complex, such as EOTP_Grp_Tran_Z which sets the Z movement to different values for each of the MOV flag values 3-5:

```
((Flag_Sage_Mov == 0)&& (Flag_nvp_mov == 0 )) ? 0 : ((Flag_Sage_Mov == 1) &&
(Flag_nvp_mov == 1)) ? 0 : ((Flag_Sage_Mov == 2) && (Flag_nvp_mov == 2)) ? 0
:((Flag_Sage_Mov == 3) && (Flag_nvp_mov == 3)) ? -56.4 :((Flag_Sage_Mov == 4) &&
(Flag_nvp_mov == 4)) ? -56.4 : ((Flag_Sage_Mov == 5) && (Flag_nvp_mov == 5)) ? -13 : 0
```

In this case the values are different between MOV flag values 4 and 5, since the EOTP is at a different location on the ISS in that axis. All measurements are in units of feet.

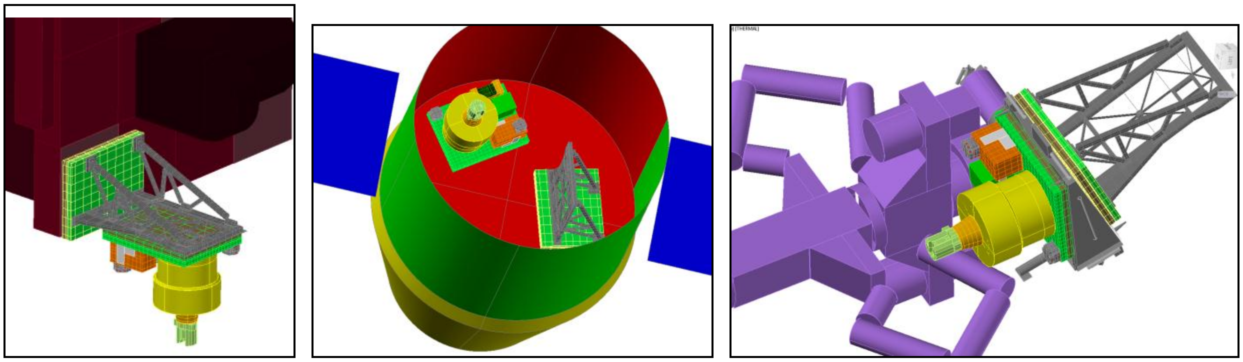


Tracker-Assembly Tree
- Articulator-Dragon EXPA Site 1::1C7A5
- Articulator-HTV::A078
- Articulator-Progress::A07B
- Articulator-SMT_Translation_Group_for_SAGE::1CD31
  - Articulator-SPDM movement::1CD30
  - Articulator-W/Sage&NVP_Back2Back_Translation to MBS::1CD39
    - Articulator-NVP_movement_to_Dragon::1C7DE
    - Articulator-SAGE_IP_assy::A42A

Figure 7. SAGE III IP and NVP articulators

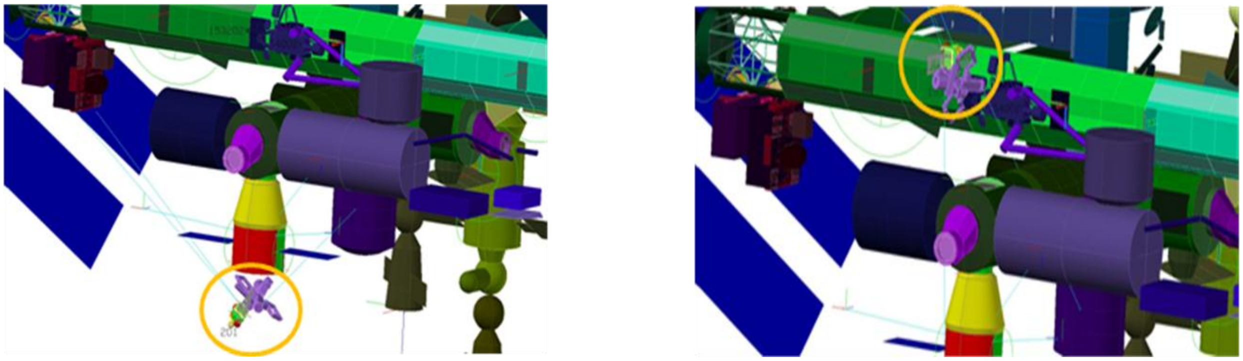
This combined SAGE/NVP articulator was placed with an articulator for the SPDM (‘SPDM movement’) on the highest level articulator for SAGE III (‘SMT_Translation_Group_for_SAGE’). The movements of this articulator are controlled by a third set of registers, e.g., SMT_Group_Tran_X using logic similar to that described above. The SPDM articulator for the robotic arm placement is controlled by a set of registers with names such as SPDM_TRAN_X, which control the placement of the EOTP at the correct location on the ISS for MOV flag location values 2-5, e.g. for the Z movement.

These three levels of articulators, and their logic for translation and rotation, work together to place the SAGE IP and NVP in their correct locations based on the MOV flag values only. The completed translations are shown in Figure 8 and Figure 9. In summary, although these registers and logic take some time to set up and verify, once they are incorporated in the model, it allows all cases and scenarios (Dragon solo flight, all EOTP locations, and mounted at ELC-4) to be run within a single model without the necessity to update several different models or import one model into another for different scenarios. Any payload that is planning to be transported to ISS on a Dragon capsule and be transferred to a permanent mounting location in this manner could take advantage of the logic developed within the SAGE III thermal model as a starting point.

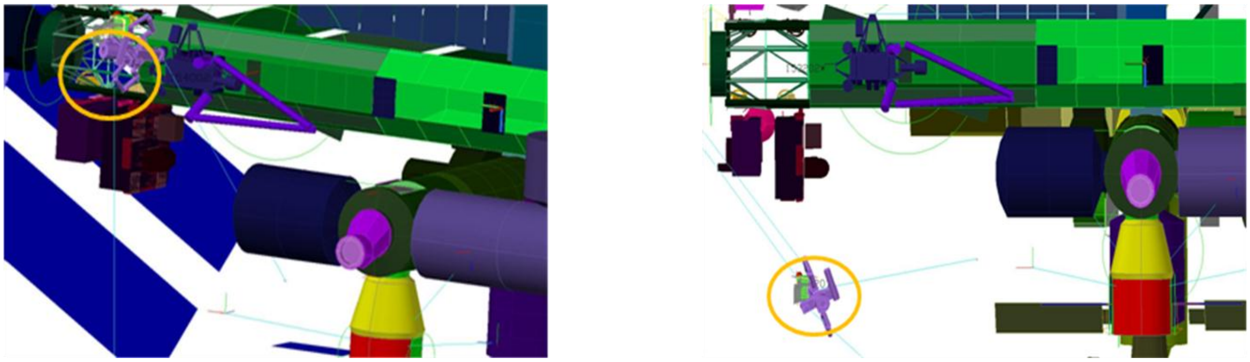


(a) ELC-4 (MOV flag = 0) (b) Dragon (MOV flag = 1) (c) EOTP (MOV flag = 2-5)

Figure 8. Locations of SAGE III IP and NVP for different configurations



(a) MOV flag = 2 (Outside Dragon) (b) MOV flag = 3 (Docked at W-5)



(c) MOV flag = 4 (Docked at W-2) (d) MOV flag = 5 (Prior to ELC-4 attachment)

Figure 9. SAGE III configurations on EOTP

DESIGN OF EXPERIMENTS METHODS FOR WORST CASE ORBIT SELECTION

Two sets of thermal environments were defined by the SAGE III thermal team; one set of environments (referred to as “ISS Extreme”) is used to verify that ISS program requirements are met (i.e. that SAGE III does not damage ISS or its payloads) and one set of environments is used to assure SAGE III mission success. Both sets of environments are based on environmental and orbital parameters provided by the ISS program; however, the SAGE III Mission Success environments are intended to represent realistic (yet still conservative) scenarios whereas the ISS Extreme environments represent off-nominal cases that are not expected in the normal operations of the ISS.

The ISS orbit has many parameters which can vary, including beta angle, yaw, pitch and roll. The ranges for each of these parameters are shown in Table 2. The design of the SAGE III payload must take into account the worst-case combinations of these parameters for the specific location of the payload on the ISS. To determine the worst-case beta angle and attitude combinations for hot and cold cases on ELC-4 and EOTP, Design of Experiments (DoE) methods were used to define sets of parametric runs and the “Find Save Files Min Max” function in TD was used to quickly process the results from each set of parametric runs.

Table 2. ISS Beta Angle and Attitude Ranges

	Beta (°)		**Yaw (°)**		**Pitch (°)**		**Roll (°)**	
	Min	Max	Min	Max	Min	Max	Min	Max
SAGE Mission Success	-75	75	-9	-3	-12	-2	0.5	1
ISS Extreme	-75	75	-15	15	-20	15	-15	15

A DoE table was first generated with all variables normalized between -1 and 1. Since repeated points are not necessary for deterministic models, space-filling designs are standard practice for setting up computer experiments, unless characteristics of the response are known in advance. Based on experience the worst-case hot and cold locations were more likely to occur at extreme angles for beta, yaw, pitch, and roll. Therefore, a 2-level full factorial DoE with the four variables (16 runs) was created to capture all vertices of the 4-dimensional Euclidean design space. A space-filling sphere packing (or maximin) DoE, generated using JMP 10 (http://www.jmp.com/software/jmp10/), was added to fill the interior of the design space. All points within a small Euclidean distance of the points at the vertices were removed to prevent wasted computational time from running duplicate or near-duplicate points. This DoE had the desired property of including all of the vertices of the design space while filling the interior evenly.

The general DoE was then scaled to generate DoEs for each set of cases (SAGE Mission Success and ISS Extreme) based on the actual limits for that set of cases. The two sets of cases were run at three locations—ELC-4 and the hottest and coldest EOTP locations—for a total of six sets of runs. The SAGE Mission Success cases at ELC-4 involved the additional complexity that SAGE III science data is only taken between beta angles of -60° to +60°. To account for this, 16 additional runs were added to that DoE table to bound the narrower beta angle range for the science cases. Points with beta angles outside this range were ignored when determining hot and cold locations for science cases. The final SAGE Mission Success DoE table at ELC-4 had 79 runs, while all others had 64 runs each.

As of the writing of this document (June 2013), the parametric runs for this study are underway. These runs represent a revision of the SAGE III worst-case environment definition. The model has changed significantly since the initial study was conducted, so it is necessary to verify that the worst-case orbits are truly being captured prior to finalizing the pre-flight predictions for SAGE III. Also, additional focus is being placed on optimizing the methodology used to define the DoE matrices.

The results of the previous set of DoE runs, which were set up in a similar but less detailed manner, are shown in Table 3 and Table 4. These parameters are currently being used in the SAGE III system-level model. The number of runs used to determine these parameters varies but was generally around 70. For the EOTP, to which SAGE III is mounted during its transfer from Dragon to ELC-4, it was first necessary to determine the worst-case hot and cold locations. As mentioned in the previous section, four positions of EOTP along the ISS backbone were selected as representative for analysis during the transfer. A set of 120 runs led to the identification of work station 2 (w-2, as shown in Figure 7) as the worst-case hot location and just outside Dragon as the worst-case cold location.

Table 3. Worst-Case Orbital Parameters – ISS Extreme Environments

Parameter	**ELC-4 Hot**	**ELC-4 Cold**	**EOTP Hot**	**EOTP Cold**
Beta Angle	-75°	-75°	-75°	+75°
Yaw	-7°	+15°	-15°	+7°
Pitch	-20°	-20°	-2.5°	-2.5°
Roll	-15°	+15°	5°	-15°

Table 4. Worst-Case Orbital Parameters – SAGE Mission Success Environments

Parameter	**ELC-4 Hot**	**ELC-4 Cold**	**EOTP Hot**	**EOTP Cold**
Beta Angle	60°	-75°	-75°	+10°
Yaw	-3°	-3°	-6°	-6°
Pitch	-2°	-7°	-6°	-12°
Roll	+0.5°	+0.5°	+0.875°	+0.5°

Although the number of runs required by this approach may seem large, it allows for exploration of the entire design space that must be considered by ISS payloads without using a full-factorial approach. Such an approach would result in a larger number of runs without providing confidence that the worst-case orbit has been identified. For example, taking beta at 10° increments and yaw at 5° increments (which may or may not be small enough increments to capture the worst case) would result in 15 * 6 = 90 runs for the ISS extreme environments and would not include the evaluation of the effects of pitch and roll. The typical approach for an ISS payload is to use engineering judgment select a set of orbits represented by the extreme ranges of beta angle and attitude. The benefit of using the DoE approach is that intermediate values for each variable can be evaluated along with the extreme values to identify the orbits that result in the most extreme temperatures for a given payload, while keeping the number of analysis runs within the realm of viability. The computers used for the SAGE III DoE runs were 64-bit machines equipped with 32GB of RAM and 2 Intel Xeon E5-2640 processors at 2.5 GHz (4 cores/2 threads each). A set of 70 runs takes approximately 4.5 days (108 hours) to complete.

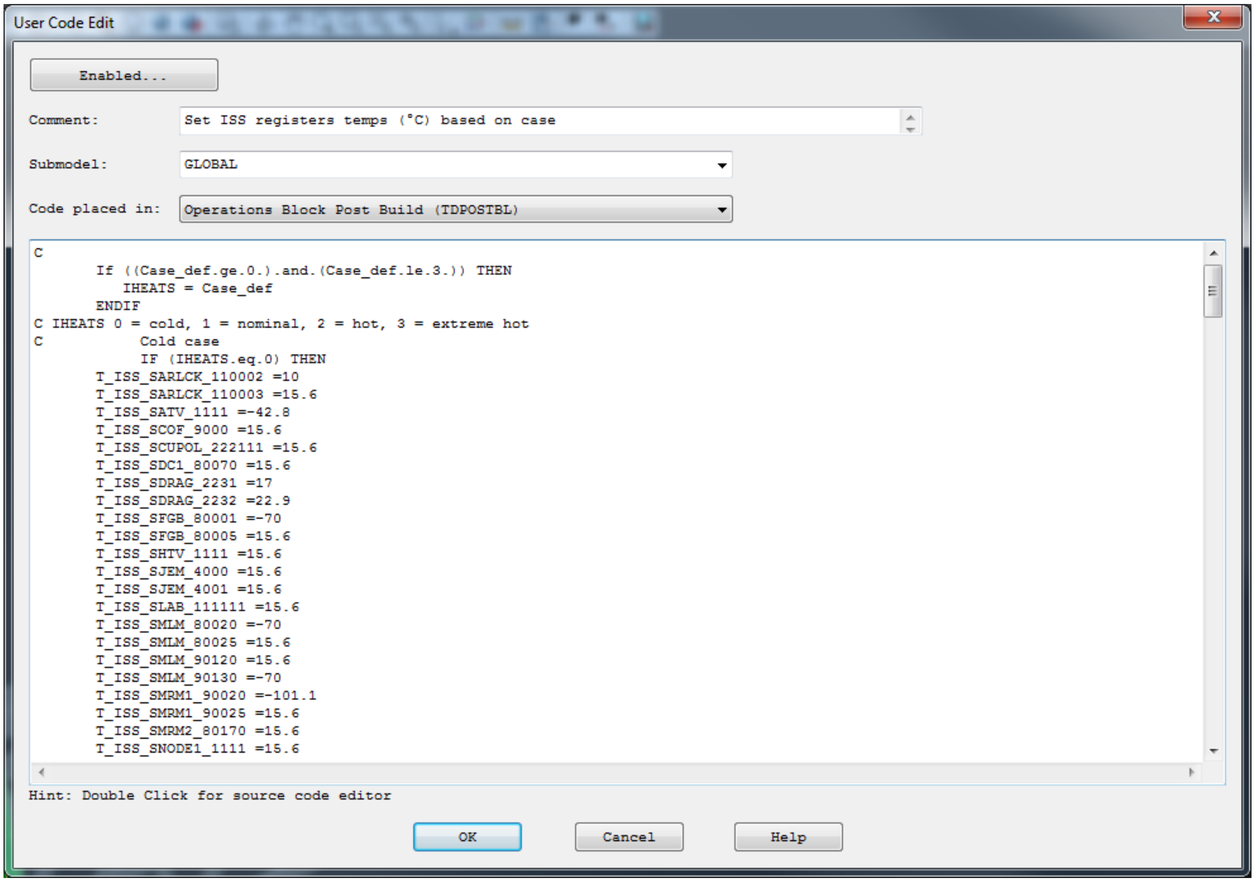
As part of the processing of results for the updated set of environments, an Efficient Global Optimization (EGO) algorithm[2] will be used to further refine the worst-case orbital parameters for SAGE III and potentially reduce the number of runs required for similar problems in the future. The rule of thumb for a DoE matrix when used along with an EGO algorithm is 10 runs per dimension, which would result in approximately 40 runs for the SAGE III environments.

TEXT SUBMODEL INCORPORATION

Several submodels that were necessary for inclusion in the model were provided either only as text, or as a combination of a text SINDA model to define nodes and conductors, and a TD model to define radiative exchange. In order to have one self-contained model, these submodels, including the ISS, HMA, and HEU, were incorporated into the SAGE III TD model using logic blocks in the Logic Manager.

The most extensive of these was the ISS model. Since this model was originally developed in TRASYS, this model existed as surfaces only, with the nodes defined by a text SINDA model. In order to facilitate working with the model, the logic was broken into several different blocks. Altogether, 128 logic blocks were used for the entire ISS model. As will be discussed later, some of there were used to facilitate changing the units of the model. Also, nearly half of them exist only to provide an automated capability to build the correct submodels based on the case, rather than having a manually constructed BUILD block for each case (which is quite cumbersome, labor-intensive, and prone to error). One main logic block was used to set up the ISS nodes, conductors and Variables 2 blocks. This text was abstracted from the ISS model that was provided (v6r4). The only change made was to set the initial temperature of all diffusion and arithmetic nodes in the ISS model to a register, T_ISS_initial, so that this value could start at the correct number depending on case scenario and temperature units (°C versus °F). In addition, the temperatures for boundary nodes in the model were each assigned their own register, so that boundary node temperatures could be easily changed by the case variable, rather than manually editing a text block.

The ISS boundary temperature registers were then set in a single block, shown in Figure 10, with logic to set their temperatures based on the case definition. The register in the ISS model used to define the hot/cold case is IHEATS, which has the values 0 for cold, 1 for nominal, 2 for hot and 3 for extreme hot. Those same values are used for the SAGE III model register to define case, Case_def, which is the only register changed in each case run to define the case, and is then used to set up all other logic and registers correctly.



Comment: Set ISS registers temps (°C) based on case

Submodel: GLOBAL

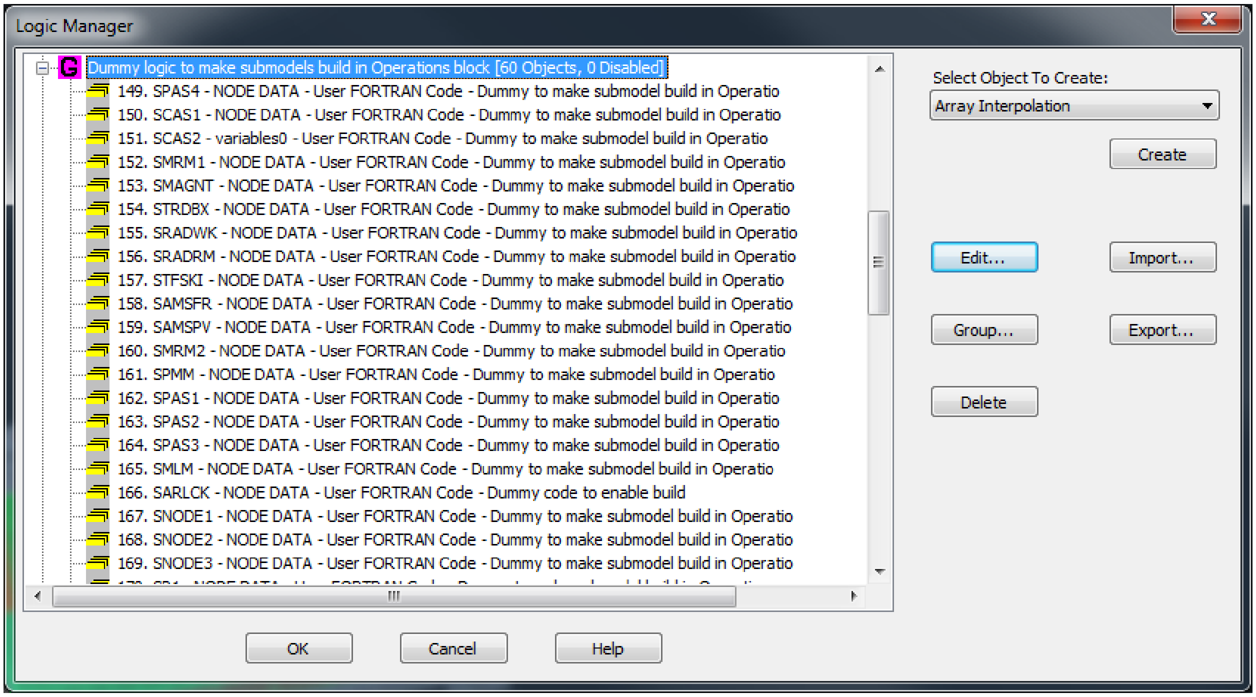
Code placed in: Operations Block Post Build (TDPOSTBL)

```
C
       If ((Case_def.ge.0.).and.(Case_def.le.3.)) THEN
          IHEATS = Case_def
       ENDIF
C IHEATS 0 = cold, 1 = nominal, 2 = hot, 3 = extreme hot
C           Cold case
            IF (IHEATS.eq.0) THEN
       T_ISS_SARLCK_110002 =10
       T_ISS_SARLCK_110003 =15.6
       T_ISS_SATV_1111 =-42.8
       T_ISS_SCOF_9000 =15.6
       T_ISS_SCUPOL_222111 =15.6
       T_ISS_SDC1_80070 =15.6
       T_ISS_SDRAG_2231 =17
       T_ISS_SDRAG_2232 =22.9
       T_ISS_SFGB_80001 =-70
       T_ISS_SFGB_80005 =15.6
       T_ISS_SHTV_1111 =15.6
       T_ISS_SJEM_4000 =15.6
       T_ISS_SJEM_4001 =15.6
       T_ISS_SLAB_111111 =15.6
       T_ISS_SMLM_80020 =-70
       T_ISS_SMLM_80025 =15.6
       T_ISS_SMLM_90120 =15.6
       T_ISS_SMLM_90130 =-70
       T_ISS_SMRM1_90020 =-101.1
       T_ISS_SMRM1_90025 =15.6
       T_ISS_SMRM2_80170 =15.6
       T_ISS_SNODE1_1111 =15.6
```

Hint: Double Click for source code editor

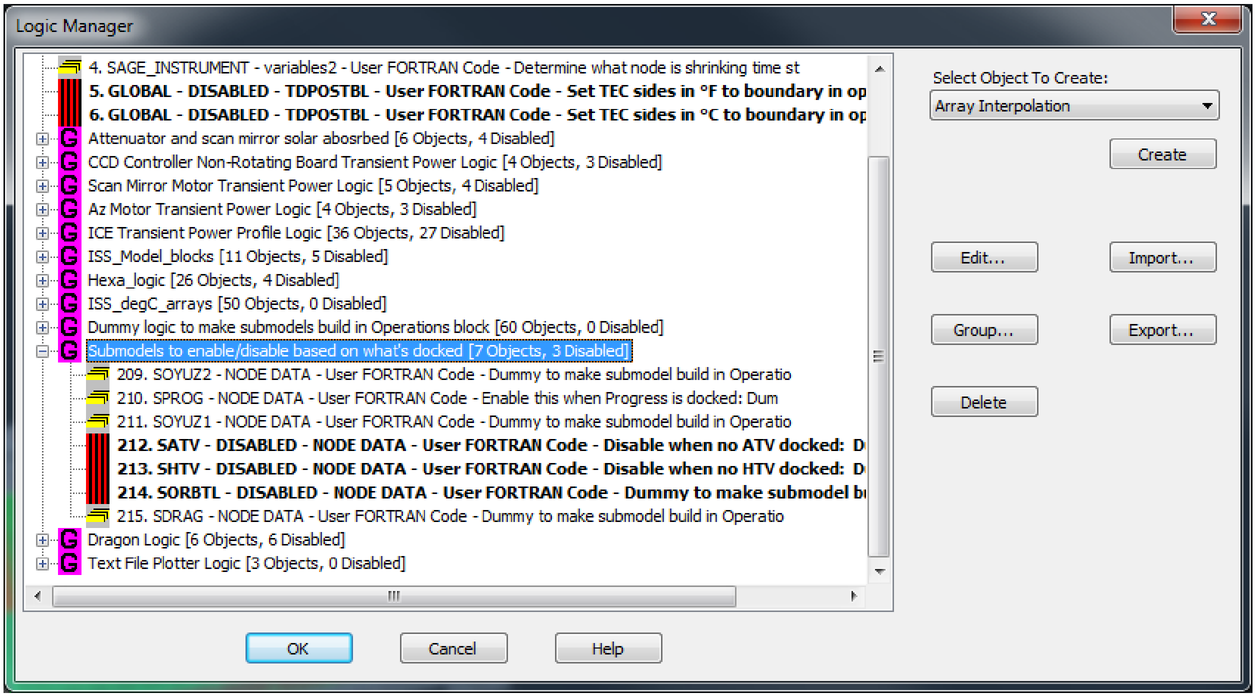
Figure 10. ISS register temperature definition logic

In order to have the correct submodels build automatically, empty logic blocks were incorporated which included a flag in the Enabled block, so that based on whether this was a case run with the ISS present, or run in the Dragon capsule in free-flight, the correct submodels would build (when a logic block is present, even if it is empty, that submodel is automatically built in the case run). Logic blocks were used to define the correct submodels of visiting vehicles (Dragon, Soyuz, etc.) based on what was docked for that case, as shown in Figure 11 and Figure 12.



Dummy logic to make submodels build in Operations block [60 Objects, 0 Disabled]
149. SPAS4 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
150. SCAS1 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
151. SCAS2 - variables0 - User FORTRAN Code - Dummy to make submodel build in Operatio
152. SMRM1 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
153. SMAGNT - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
154. STRDBX - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
155. SRADWK - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
156. SRADRM - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
157. STFSKI - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
158. SAMSFR - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
159. SAMSPV - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
160. SMRM2 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
161. SPMM - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
162. SPAS1 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
163. SPAS2 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
164. SPAS3 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
165. SMLM - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
166. SARLCK - NODE DATA - User FORTRAN Code - Dummy code to enable build
167. SNODE1 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
168. SNODE2 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
169. SNODE3 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio

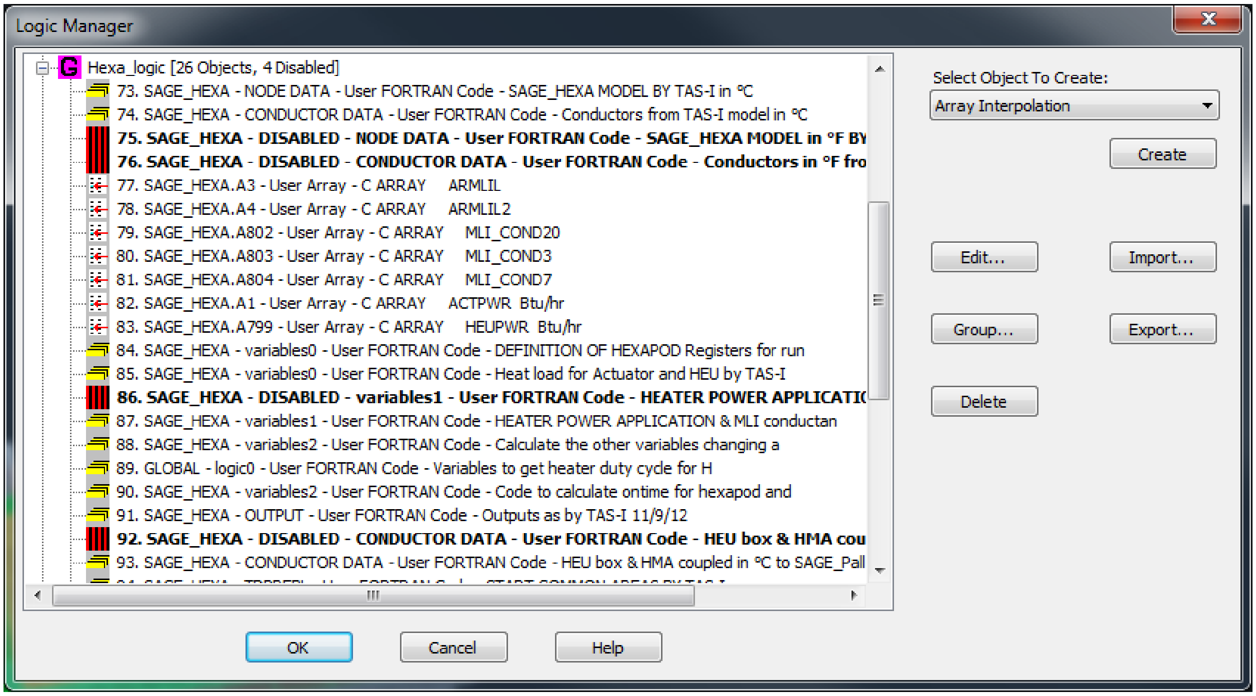
Figure 11. ISS logic blocks to allow build of correct ISS submodels



4. SAGE_INSTRUMENT - variables2 - User FORTRAN Code - Determine what node is shrinking time st
5. GLOBAL - DISABLED - TDPOSTBL - User FORTRAN Code - Set TEC sides in °F to boundary in op
6. GLOBAL - DISABLED - TDPOSTBL - User FORTRAN Code - Set TEC sides in °C to boundary in op
Attenuator and scan mirror solar abosrbed [6 Objects, 4 Disabled]
CCD Controller Non-Rotating Board Transient Power Logic [4 Objects, 3 Disabled]
Scan Mirror Motor Transient Power Logic [5 Objects, 4 Disabled]
Az Motor Transient Power Logic [4 Objects, 3 Disabled]
ICE Transient Power Profile Logic [36 Objects, 27 Disabled]
ISS_Model_blocks [11 Objects, 5 Disabled]
Hexa_logic [26 Objects, 4 Disabled]
ISS_degC_arrays [50 Objects, 0 Disabled]
Dummy logic to make submodels build in Operations block [60 Objects, 0 Disabled]
Submodels to enable/disable based on what's docked [7 Objects, 3 Disabled]
209. SOYUZ2 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
210. SPROG - NODE DATA - User FORTRAN Code - Enable this when Progress is docked: Dum
211. SOYUZ1 - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
212. SATV - DISABLED - NODE DATA - User FORTRAN Code - Disable when no ATV docked: D
213. SHTV - DISABLED - NODE DATA - User FORTRAN Code - Disable when no HTV docked: D
214. SORBTL - DISABLED - NODE DATA - User FORTRAN Code - Dummy to make submodel b
215. SDRAG - NODE DATA - User FORTRAN Code - Dummy to make submodel build in Operatio
Dragon Logic [6 Objects, 6 Disabled]
Text File Plotter Logic [3 Objects, 0 Disabled]

Figure 12. ISS logic blocks to allow build of correct visiting vehicle submodels

In addition to ISS, the other models that were originally provided in text form were the HMA and HEU models. These models were developed by TAS-I, who developed the HMA and HEU hardware. These thermal models were originally in ESARAD and ESATAN, and as such they were provided as separate radiation and thermal models, with the thermal model in SINDA text form. Logic blocks (26 of them) are used to define the nodes, conductors, power, heaters, and operational logic for the HMA and HEU, as shown in Figure 13.



Hexa_logic [26 Objects, 4 Disabled]
73. SAGE_HEXA - NODE DATA - User FORTRAN Code - SAGE_HEXA MODEL BY TAS-I in °C
74. SAGE_HEXA - CONDUCTOR DATA - User FORTRAN Code - Conductors from TAS-I model in °C
75. SAGE_HEXA - DISABLED - NODE DATA - User FORTRAN Code - SAGE_HEXA MODEL in °F BY
76. SAGE_HEXA - DISABLED - CONDUCTOR DATA - User FORTRAN Code - Conductors in °F fro
77. SAGE_HEXA.A3 - User Array - C ARRAY ARMLIL
78. SAGE_HEXA.A4 - User Array - C ARRAY ARMLIL2
79. SAGE_HEXA.A802 - User Array - C ARRAY MLI_COND20
80. SAGE_HEXA.A803 - User Array - C ARRAY MLI_COND3
81. SAGE_HEXA.A804 - User Array - C ARRAY MLI_COND7
82. SAGE_HEXA.A1 - User Array - C ARRAY ACTPWR Btu/hr
83. SAGE_HEXA.A799 - User Array - C ARRAY HEUPWR Btu/hr
84. SAGE_HEXA - variables0 - User FORTRAN Code - DEFINITION OF HEXAPOD Registers for run
85. SAGE_HEXA - variables0 - User FORTRAN Code - Heat load for Actuator and HEU by TAS-I
86. SAGE_HEXA - DISABLED - variables1 - User FORTRAN Code - HEATER POWER APPLICATI
87. SAGE_HEXA - variables1 - User FORTRAN Code - HEATER POWER APPLICATION & MLI conductan
88. SAGE_HEXA - variables2 - User FORTRAN Code - Calculate the other variables changing a
89. GLOBAL - logic0 - User FORTRAN Code - Variables to get heater duty cycle for H
90. SAGE_HEXA - variables2 - User FORTRAN Code - Code to calculate ontime for hexapod and
91. SAGE_HEXA - OUTPUT - User FORTRAN Code - Outputs as by TAS-I 11/9/12
92. SAGE_HEXA - DISABLED - CONDUCTOR DATA - User FORTRAN Code - HEU box & HMA cou
93. SAGE_HEXA - CONDUCTOR DATA - User FORTRAN Code - HEU box & HMA coupled in °C to SAGE_Pall

Select Object To Create:
Array Interpolation

Figure 13. Hexapod logic blocks

Figure 14 shows the complex internal radiative surface structure within the HMA (outer MLI covering is made partially transparent in this image so that internal details can be seen). Because the node numbers of the original text radiative and thermal models were not aligned, node correspondence was used to align the radiative node numbers with the thermal node numbers created in the logic, as shown in Figure 15. This node correspondence allows the thermal results to be viewed on the thermal HMA surfaces present in the model, even though none of them have node and conductors created directly.

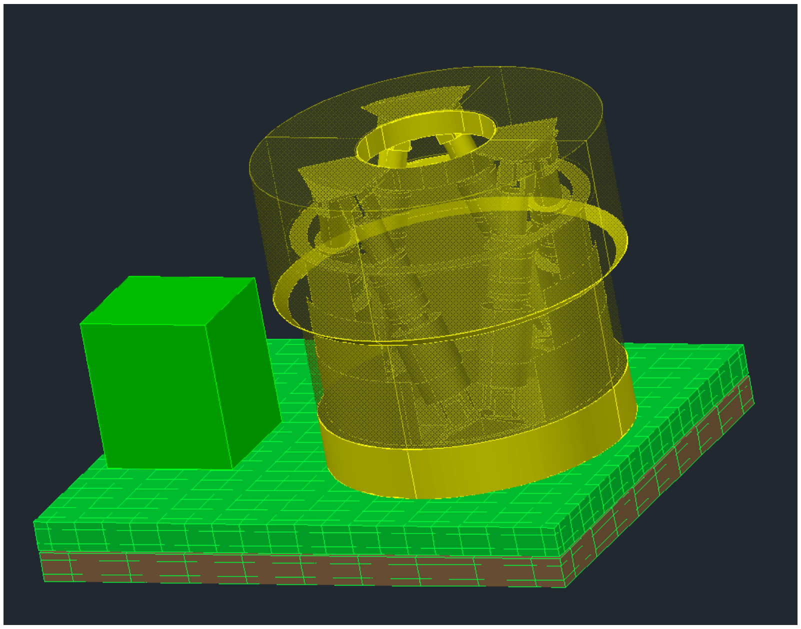
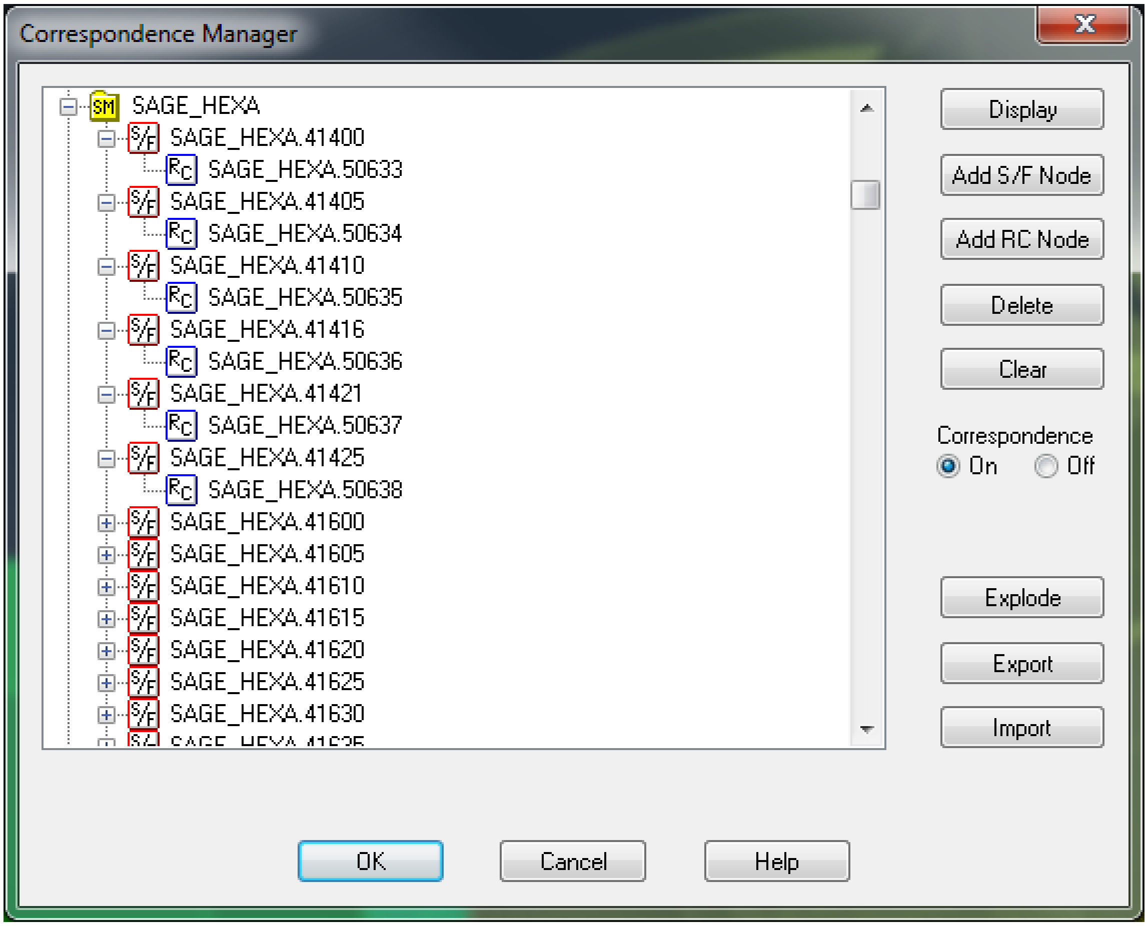


Figure 14. HMA and HEU thermal models



SAGE_HEXA
- SAGE_HEXA.41400
  - SAGE_HEXA.50633
- SAGE_HEXA.41405
  - SAGE_HEXA.50634
- SAGE_HEXA.41410
  - SAGE_HEXA.50635
- SAGE_HEXA.41416
  - SAGE_HEXA.50636
- SAGE_HEXA.41421
  - SAGE_HEXA.50637
- SAGE_HEXA.41425
  - SAGE_HEXA.50638
- SAGE_HEXA.41600
- SAGE_HEXA.41605
- SAGE_HEXA.41610
- SAGE_HEXA.41615
- SAGE_HEXA.41620
- SAGE_HEXA.41625
- SAGE_HEXA.41630

Correspondence ● On ○ Off

Figure 15. Node correspondence example for HMA model

FACILITATION OF UNITS CHANGE

This model was originally developed in metric units (W, s, m, °C). However, because of the large amount of legacy model information to be included, as well as the requirement to deliver to the ISS program in British units, the model was switched to British units (Btu, hr, ft, °F) early in the program. After several months of running and presenting in this set of units, many issues became clear. The personnel responsible for science and testing preferred data to be presented in °C for the sake of consistency with their methods. All power specifications for electronics are in W. Most thermal limit specifications on electronics are in °C. For these reasons, as well as the preference of the thermal team, it was desirable to run the model and present results in °C. There is an option within TD to run the model in °F and simply present results in °C. However, all inputs to the model would still have to be in °F, and all binary save files would have to be maintained in °F so that restarts from those files would be accurate. The thermal team decided that the option of running in °F and presenting in °C would not be acceptable. The team decided that the best option would be to run the model in °C. However, since the payload portion of the model would need to be delivered back to the ISS program in °F, a switch was developed in the model to allow it to be run in either °F or °C. This allows the payload developer to do runs and comparison to limits and testing in °C, and deliver the model to ISS in °F. Performing this same conversion for the length unit was determined to be very labor intensive without providing sufficient benefits to be worthwhile. Powers are in general input to the program in W, and the input settings on the forms are used to change the power to Btu/hr.

The temperature unit switch was developed as follows. A register was created named switch_temp_units which is 0 for °C and 1 for °F Each logic block in the model that uses temperature units (node definition, conductor definition, boundary temperatures, calculation of temperature-dependent MLI conduction by TASI) has two versions. One is activated by the flag in the Enabled block when the units are °C (switch_temp_units=0), and one is activated when units are °F (switch_temp_units=1). The logic in each needed to be correct for the unit set being used. This is complicated in the case of logic blocks that use several different FAC cards throughout the block to change the units of input, because each FAC card must be changed accordingly. But once this work is done, there are only two things that must be done to run the model in a difference unit set: change the register switch_temp_units, and change Thermal>Preferences>Units.

This modification of the model has been tested repeatedly by running the model in both unit sets, and comparing results, and the same results are obtained. The thermal team has found presentation of results much cleaner in °C. When the model was in °F it became necessary to present results in both unit sets (for ease of communication to the team and stakeholders), and having two sets of results tables led to messy reports. The thermal team is very satisfied with the model operation in this way. The only minor irritation is that the results must be viewed in °F before mapping the thermal results on to the structural model (since the structural model requires temperature in °F), but that is rarely done, and very simple to achieve.

CASE-BASED LOGIC

TD logic blocks are used to set parameters that change based on the analysis case, specifically power dissipations for electronics components and heaters. Flags are used to define various scenarios, as shown in Table 5. The use of these flags in logic blocks makes it easy to simulate many different scenarios within a single model. The power dissipations for some of the SAGE III subsystems are defined as arrays based on which types of science data are being taken in a particular orbit.

Table 6 provides a summary of the science events in each analysis case. The use of the flags case_def and flag_limb provides a simple way to define the appropriate power dissipations for each scenario.

Table 5. Flags Used in Case-Based Logic

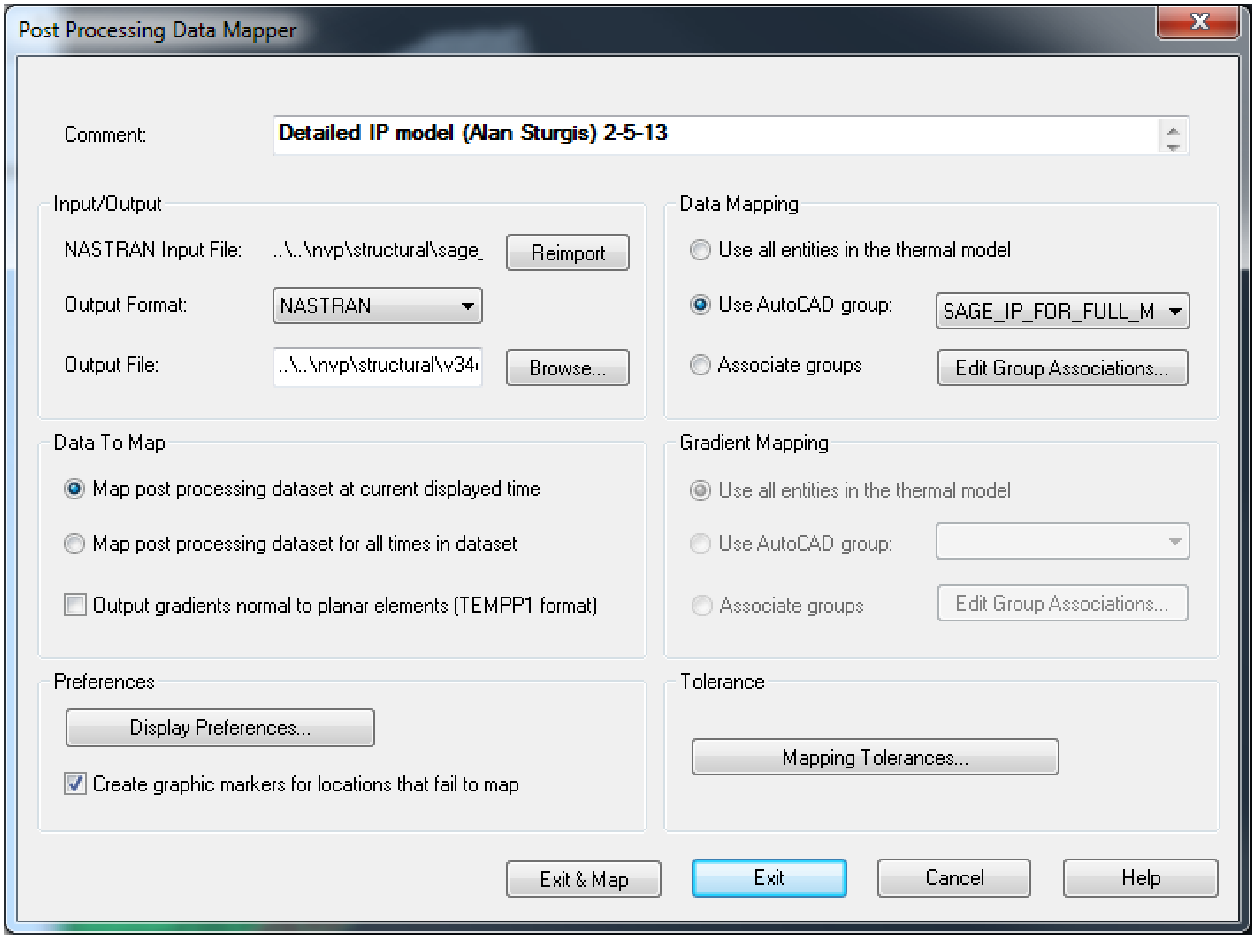
Flag	Values	Function(s)
Case_def	0 = cold, 1 = nominal, 2 =hot, 3 = extreme hot	Sets appropriate ISS boundary temperatures, solar flux absorbed by the SA, and SAGE III component power dissipations
Case_site	0 = ELC-4, 1 = EOTP, 2 = Dragon	Sets location of SAGE III payloads
Flag_survival	1= survival case, 0= all other cases	Turns off operational power
Flag_transfer	1= unpowered case, 0= all other cases	Turns off all operational and heater power to simulate unpowered scenarios
Flag_limb	1= limb-only case, 0= all other cases	Signifies a case where only limb scattering data is taken (vs. solar and lunar occultation measurements)
Flag_voltage	0 = min, 1 = nominal, 2 = max	Sets appropriate survival heater power for each location (ELC-4, EOTP, and Dragon)

Table 6. Breakdown of Science Events per Analysis Case

Case	Solar Occultation Events		Lunar Occultation Events		Limb Scattering Events	
	#	Length (min)	#	Length (min)	#	Length (min)
Extreme hot	2	3.75	2	2	1	5
Nominal hot	2	3.75	1	2	1	5
Cold	2	1.6	0	---	0	---
Limb-only hot	0	---	0	---	1	10
Limb-only cold	0	---	0	---	1	20

MAPPING THERMAL RESULTS TO DIFFERENT STRUCTURAL MODELS

Requirements from ISS include survival of all payload conditions without excessive stress or deflection. One of the payload science requirements is to remain below a set deflection before and during each science event, so that the required pointing accuracy can be achieved. Thus, the structural deflection must be determined using the temperature gradient predictions on all parts; to achieve that, the thermal results must be mapped onto the structural model. This is easily done using the Post Processing Data Mapper within TD. One issue, as mentioned, is that the structural model is set up to utilize temperatures in °F. Thus, before a set of runs are mapped, the TD unit preferences must be changed to °F (for viewing only, not for the solution). Then the post-processing data mapper form, shown in Figure 16, is used. The best mapping is achieved if the mapping is limited to the desired AutoCAD group within the TD model, so that thermal parts are not included which are not part of the structural model. The thermal data can be mapped at either one point in the time set or all times. An example of a detailed structural model superimposed on the thermal model is shown in Figure 17 and Figure 18.



Post Processing Data Mapper

Comment: **Detailed IP model (Alan Sturgis) 2-5-13**

Input/Output

NASTRAN Input File: ..\..\nvp\structural\sage_

Output Format: NASTRAN

Output File: ..\..\nvp\structural\v34

Data Mapping

○ Use all entities in the thermal model

● Use AutoCAD group: SAGE_IP_FOR_FULL_M

○ Associate groups

Data To Map

● Map post processing dataset at current displayed time

○ Map post processing dataset for all times in dataset

☐ Output gradients normal to planar elements (TEMPP1 format)

Gradient Mapping

● Use all entities in the thermal model

○ Use AutoCAD group:

○ Associate groups

Preferences

☑ Create graphic markers for locations that fail to map

Tolerance

Figure 16. Post Processing Data Mapper form

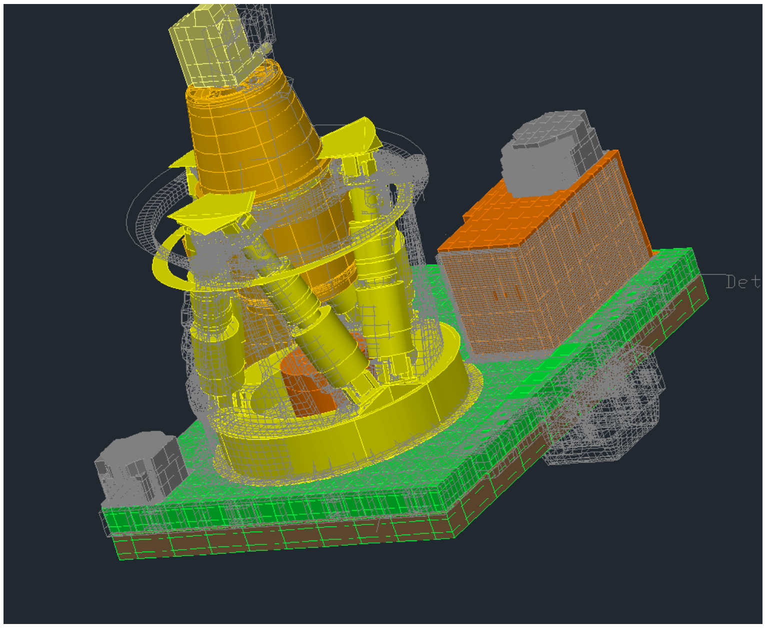


Figure 17. Structural model (in grey) superimposed on thermal model

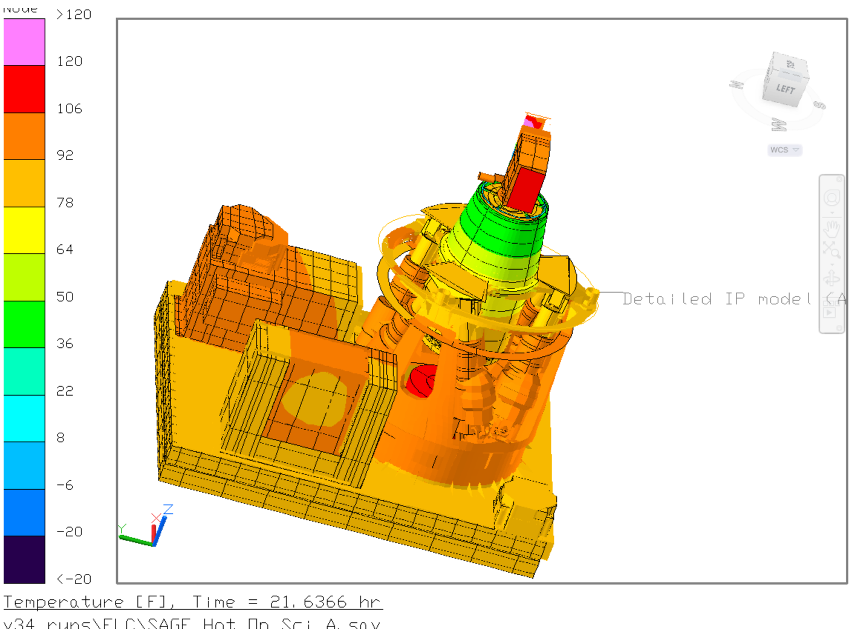


Figure 18. Mapped structural model superimposed on thermal model

There are a few methods which make this mapping easier. First, the structural model file size is often very large (>50GB). Thus, mapping of multiple points along a timeline creates extremely large files. It is normally very difficult to determine from a thermal timeline where the maximum stress or deflection will occur, and so it is difficult to select just a single best point to map for the worst case stress. Thus, registers were created in the model to calculate the temperature deltas in each of three axes across the major parts (NVP and IP). This simplified magnitude of thermal gradient across the parts can then be viewed as a transient in time. By mapping several time points to the structural model, the variables which best predicted the stress level in the parts were determined. Then, after a timeline was run, those registers could be plotted, and the temperatures only mapped to the structural model at the time point where those were at maximum.

Another issue with the large file size of the structural models is that they are unwieldy when maintained within the model, since they more than triple the thermal model size. Also, at least in the case of the SAGE III project, there are several structural models that are used for mapping. Maintaining all of them inside the thermal model uses a large amount of space, as well as slowing the model down in terms of graphics display. However, since they must be resized and moved into location each time they are imported, it is a substantial investment of time to remove them after each mapping and then re-import them. The solution to this was to create a coordinate system for each structural model. When any given structural model is to be imported for mapping, the user coordinate system is set to the appropriate one, and thus the import can be easily accomplished in a single step. The unit conversion from the structural model length units to the thermal model units is accomplished within the import form itself.

STREAMLINED RESULTS PROCESSING

Due to the large number of analysis cases being run by the SAGE III thermal team, it was necessary to determine an efficient way to process the results. An excel-based tool called FilePlottingTools was developed at LaRC by Salvatore Scola (http://fileplottingtools.larc.nasa.gov/). This custom add-in, developed in VB.net, allows users to quickly compare thermal analysis results from many different analysis cases through the use of plots and numerical tables. The add-in can plot and compare delimited text files, which can be produced from logic blocks in most analysis software packages, as well as data acquisition systems from test facilities. Save files from TD can also be imported and specific nodes and registers can be selected so that the user can focus on specific areas of interest. The intuitive interface greatly reduces the time required to develop presentation and report-quality plots from large time-dependent data sets. The “compare min max” feature allows users to compare results from various runs with the click of a single button. After the add-in is installed, a new tab appears in Excel from which all of the FilePlottingTools functions can be accessed, as shown in Figure 19.

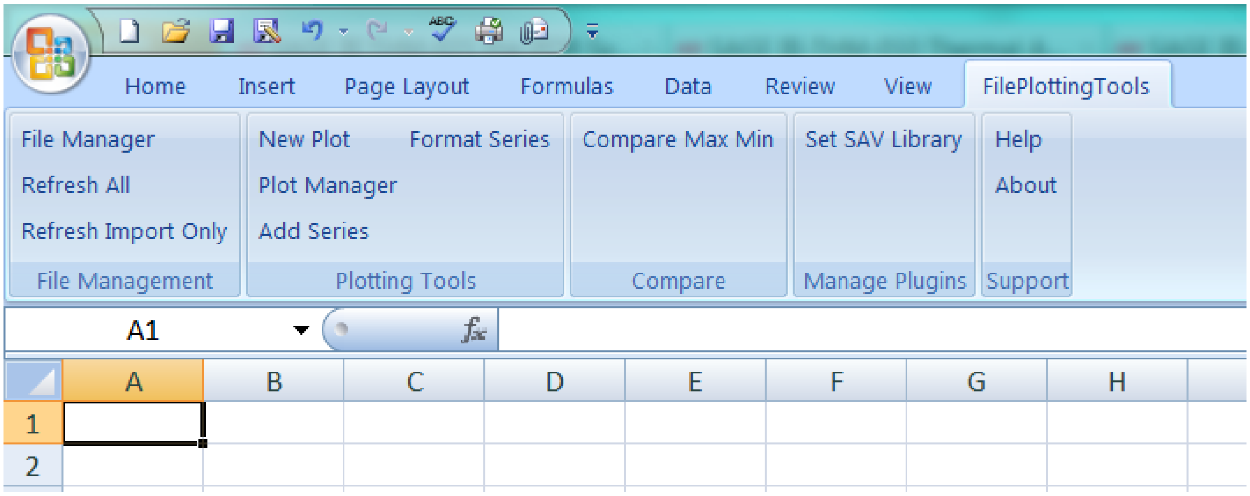
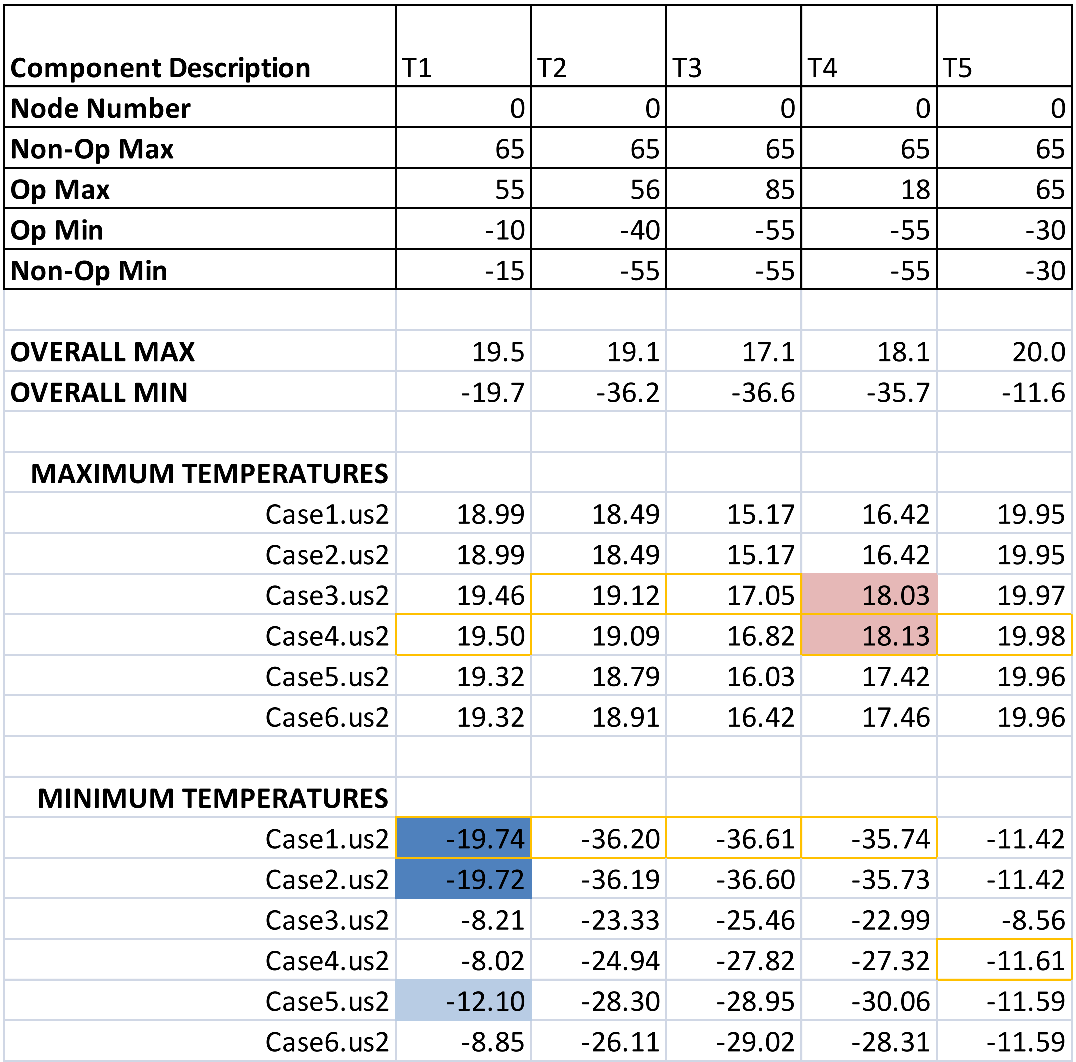


Figure 19. FilePlottingTools Tab in Excel

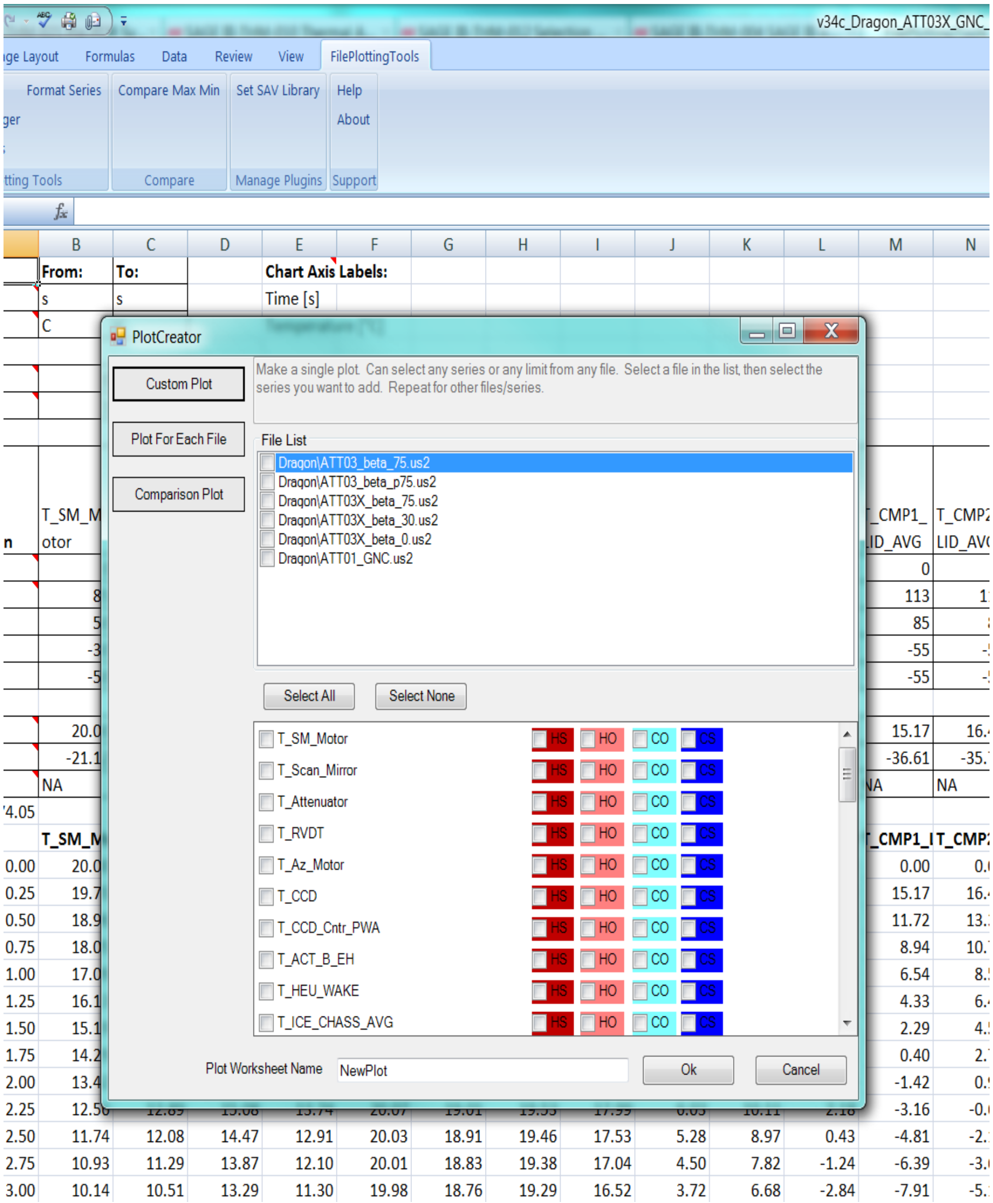
Importing files is accomplished simply by dragging them into the File Manager window from Windows Explorer. In the File Manager window, the user can select to compare files or stitch files. The stitch feature is useful for the SAGE III thermal team when it is necessary to plot temperature data for a power-off scenario (starting from a cold survival case, heater power is turned off for 6 hours). If TD save files are used, component descriptions corresponding to node numbers or register names are automatically loaded included in the column titles. File names are also automatically imported. For each component, the user can enter temperature limits. Conditional formatting is in place on the “Formatted Data” tab that highlights instances where limits are reached. Using the “Compare Max Min” function results in an easy-to-use comparison of maximum and minimum temperatures for each results file. An example is provided in Figure 20. Note that a darker shade of blue indicates that a non-op minimum limit was reached, whereas a lighter blue indicates that an op limit was reached. The minimum and maximum temperatures over all of the cases are outlined in yellow.



Component Description	T1	T2	T3	T4	T5
Node Number	0	0	0	0	0
Non-Op Max	65	65	65	65	65
Op Max	55	56	85	18	65
Op Min	-10	-40	-55	-55	-30
Non-Op Min	-15	-55	-55	-55	-30
OVERALL MAX	19.5	19.1	17.1	18.1	20.0
OVERALL MIN	-19.7	-36.2	-36.6	-35.7	-11.6
MAXIMUM TEMPERATURES					
Case1.us2	18.99	18.49	15.17	16.42	19.95
Case2.us2	18.99	18.49	15.17	16.42	19.95
Case3.us2	19.46	19.12	17.05	18.03	19.97
Case4.us2	19.50	19.09	16.82	18.13	19.98
Case5.us2	19.32	18.79	16.03	17.42	19.96
Case6.us2	19.32	18.91	16.42	17.46	19.96
MINIMUM TEMPERATURES					
Case1.us2	-19.74	-36.20	-36.61	-35.74	-11.42
Case2.us2	-19.72	-36.19	-36.60	-35.73	-11.42
Case3.us2	-8.21	-23.33	-25.46	-22.99	-8.56
Case4.us2	-8.02	-24.94	-27.82	-27.32	-11.61
Case5.us2	-12.10	-28.30	-28.95	-30.06	-11.59
Case6.us2	-8.85	-26.11	-29.02	-28.31	-11.59

Figure 20. Example of Results Comparison in FilePlottingTools

Figure 21 shows a screenshot of the Plot Creator window. Within this window, the user can select to create a plot for each file that would include the same temperatures, create a comparison plot with data from more than one file on a single plot, or create a custom plot with the ability to select files and temperatures as desired. The temperature limits can be included on the plot if desired by selecting the corresponding check boxes. Plot formats can be adjusted very easily, but is generally report-ready without making any adjustments. Figure 22 provides an example plot.



Make a single plot. Can select any series or any limit from any file. Select a file in the list, then select the series you want to add. Repeat for other files/series.

File List

- ☐ Dragon\ATT03_beta_75.us2
- ☐ Dragon\ATT03_beta_p75.us2
- ☐ Dragon\ATT03X_beta_75.us2
- ☐ Dragon\ATT03X_beta_30.us2
- ☐ Dragon\ATT03X_beta_0.us2
- ☐ Dragon\ATT01_GNC.us2

- ☐ T_SM_Motor
- ☐ T_Scan_Mirror
- ☐ T_Attenuator
- ☐ T_RVDT
- ☐ T_Az_Motor
- ☐ T_CCD
- ☐ T_CCD_Cntr_PWA
- ☐ T_ACT_B_EH
- ☐ T_HEU_WAKE
- ☐ T_ICE_CHASS_AVG

Plot Worksheet Name: NewPlot

From: s
To: s
Chart Axis Labels: Time [s]

Figure 21. Plot Creator Window in FilePlottingTools

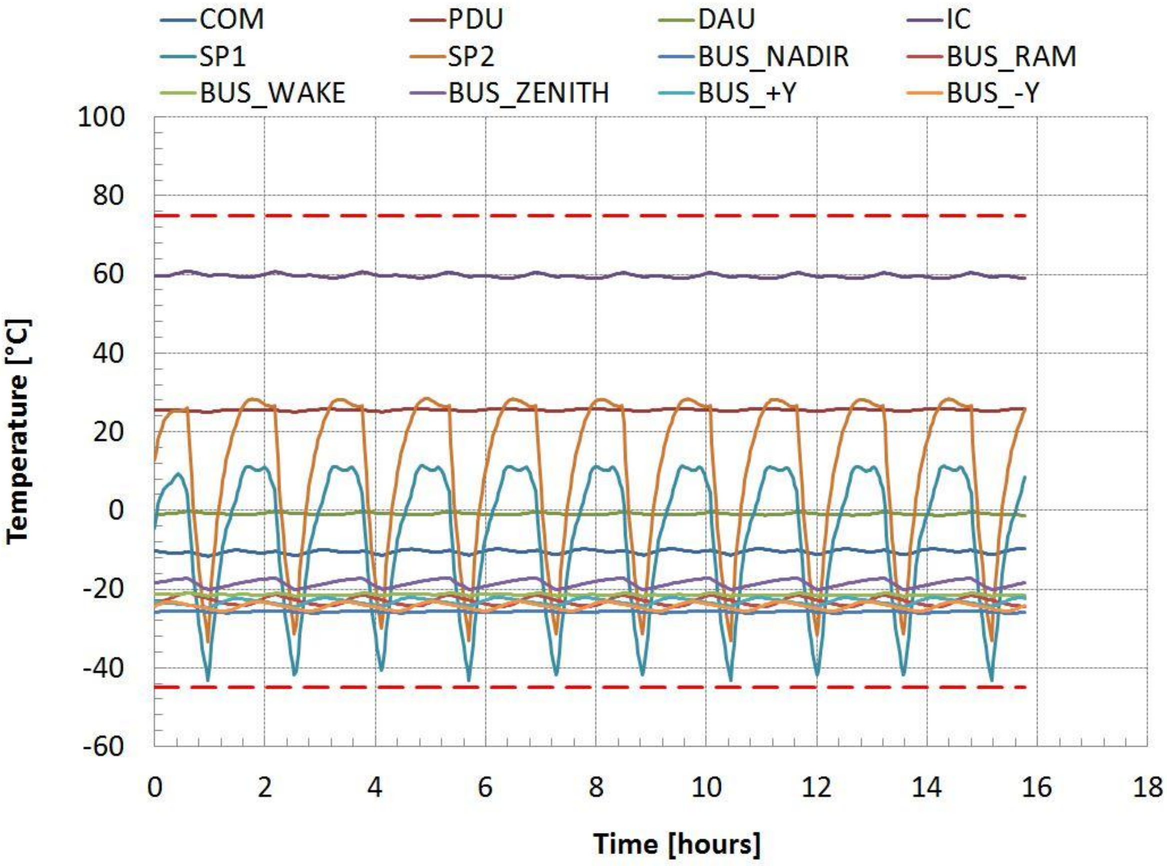


Figure 22. Sample Temperature Plot from FilePlottingTools

FilePlottingTools has been extremely useful for the SAGE III team and has made results processing much more efficient. Using this tool allows results to be quickly viewed in many different formats and easily make comparisons that would have previously required considerable manual data manipulation. An additional feature that is particularly useful is that FilePlottingTools automatically calculates and displays the time-to-limit (TTL) for each component. This is critical for SAGE III because TTL is the preferred way of reporting results from the analysis cases in the Dragon trunk.

SUMMARY

The use of the methods described in this document significantly improved the efficiency with which the SAGE III system-level thermal analysis can be performed. While each of these methods required a considerable up-front time investment, the increase in efficiency in the long-run has been invaluable and as such, the authors recommend implementing some or all of these methods when developing thermal models. Some of the methods are only applicable in special situations, such as when it is necessary to incorporate text-based models with a graphics-based TD model or when the ability to switch between unit sets is required; however, most of them are useful for all thermal models, including the use of case-based logic to facilitate changing power levels for various scenarios, the incorporation of several coordinate frames to facilitate easy mapping of temperature data to structural models, and the streamlined results processing capabilities made possible by using FilePlottingTools. Two of the methods described in this document, the use of TD assemblies to move payloads to various locations and the implementation of a DoE methodology to define worst-case orbits, are particularly useful for ISS payload developers.

REFERENCES

1. Timothy D. Panczak, Steven G. Ring, Mark J. Welch, David Johnson, Brent A. Cullimore, Douglas P. Bell. 2012. CRTech Thermal Desktop® User's Manual, v5.5.

2. Donald R. Jones, Matthias Schonlau, and William J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. J. of Global Optimization 13, 4 (December 1998), 455-492.