N94-23638

A COMPARISON OF TSS AND TRASYS IN FORM FACTOR CALCULATION

Eric Golliher
Thermal and Fluids Analysis Branch
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

INTRODUCTION

As the workstation and personal computer become more popular than a centralized mainframe to perform thermal analysis, the methods for space vehicle thermal analysis will change. Already, many thermal analysis codes are now available for workstations, which were not in existence just five years ago. As these changes occur, some organizations will adopt the new codes and analysis techniques, while others will not. This might lead to misunderstandings between thermal shops in different organizations. If thermal analysts make an effort to understand the major differences between the new and old methods, a smoother transition to a more efficient and more versatile thermal analysis environment will be realized.

DISCUSSION

As dedicated computers becomes more affordable and faster, the method for performing radiation thermal analysis using a "ray-tracing" [1] technique may become the standard. The advantage of some ray-tracing codes lies in their versatility: the ability to account for specular and transmissive surfaces, and to handle boolean geometrical constructions, fence problems, and the box-on-a-plate problem. The disadvantage is that ray-tracing has historically been thought of as less computationally efficient than the traditional method for typical space vehicle thermal design problems, which has been the "unit sphere" or "double summation" method. Since many mainframe computer departments account for costs on a per-CPU hour basis, double summation codes such as TRASYS and VectorSweep currently dominate. However, with the popularity and speed of workstations and personal computers on the rise, many new ray tracing codes and enhancements are taking shape.

Examples of some of the codes with full ray-tracing capability are ESARAD (European Space Agency Research and Technology Center), NEVADA (Turner Associates), OPERA (Boeing Monte Carlo), TMG (Thermal Model Generator from Maya Heat Transfer), and TSS (Thermal Synthesizer System, sponsored by NASA-JSC). Since the workstation costs are typically accounted as a one-time purchase cost, the perceived longer "run-times" with workstations are not a cost issue. Also, since most large satellite/space station thermal problems require many hours of CPU time on a mainframe as well as on a workstation, the "turn-around" time for both is comparable.

This discussion centers on two codes used in form factor radiation thermal analysis for space vehicles: TSS, as an example of a ray-tracing code, and TRASYS, an example of the unit sphere method. A comparison among the different ray-tracing codes is beyond the scope of this paper. Also, a comparison of the orbital heating rate calculations is not the subject of this paper; although it is acknowledged that the issues described here may cause similar problems with the heating rate algorithms as well.

The motivation for this report came about during a recent project in which NASA-LeRC was involved, called SOHO (Solar and Heliospheric Observatory). The satellite is being integrated by Matra-Marconi of France under contract to ESA/ESTEC in Noordwijk, Holland. The launch vehicle, an Atlas IIAS, is being provided by General Dynamics Commercial Launch Services under contract to NASA-LeRC. The launch date is June 1995. Figure 1 shows the ITPLOT[2] TRASYS plot of the SOHO spacecraft.

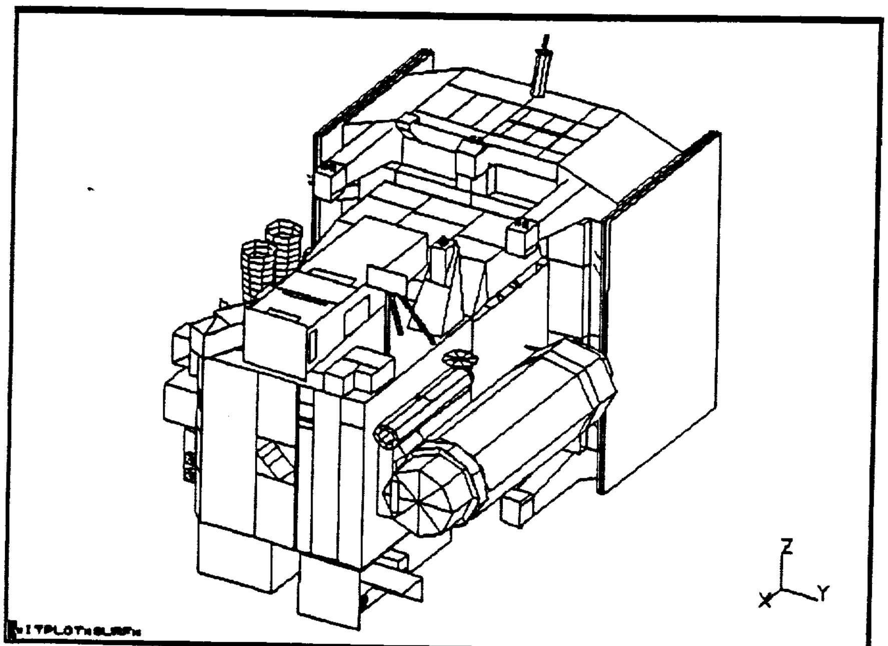

Figure 1 SOHO Spacecraft Geometrical Math Model (GMM)

Recently, ESA/MATRA delivered to NASA/GD simplified thermal models of the satellite which contained approximately 800 surfaces. NASA intended to use TRASYS, a Nusselt sphere-type code, to perform the integrated thermal analysis of the launch phase of the mission. The original GMM was created by Matra in Toulousse, France using ESABASE, a geometry builder for their ray-tracing type radiation thermal analysis code, THERMICA. Therefore, a conversion program had to be written at LeRC to convert ESABASE to TRASYS. The converted GMM was used to generate Hottel-type radiation conductors (RADK's) in TRASYS and then in TSS. After the conversion process, NASA-LeRC plotted the TRASYS surfaces. There appeared to be many surfaces running into each other and many box-on-a-plate problems. It was learned however, that these types of "errors" can, in fact, be handled by ray tracing type codes. Thus, this report can serve as a "lessons learned" from a user's perspective.

It was decided to try using a ray-tracing code at NASA/LeRC to confirm the original TRASYS/SINDA SOHO launch phase temperature predictions[3]. The intent was to compare form factors directly using TSS at NASA/LeRC. However, this became too unwieldy for an 800-node model. A more practical approach was taken. Final temperatures from various SINDA analyses were compared for the corresponding Thermal Math Model (TMM). In each situation, the original SINDA thermal math model contained only TRASYS-generated heating rates and radiation conductors (RADK's). Then, RADK's generated by TSS were switched for those generated by TRASYS, and the temperatures compared.

When using TRASYS, the analyst has a variety of options in the FFDATA and RKDATA statements which control the accuracy of the resultant form factors. Over the years, typical parameters which seem to work well for most space vehicle analysis have been developed through trial and error. Also, rules and techniques for "good" TRASYS model-building have been established[4]. Since it was known that the model in this case contained box-on-plate problems, it was decided to push TRASYS to its limit with some unreasonably tight FFDATA parameters: NELCT=200, FFRATL=-1.0, and FFACS=0.01. The CPU time for RADK's was approximately 8 hours on a VAX 9410.

When using TSS or any other ray-tracing code, the analyst must deal with a completely different set of control parameters and these are generally not well known to engineers familiar with TRASYS. These parameters include Energy Cut Off Factor, Number of Rays per Surface, Numbers of Levels and Objects in the Oct-Tree Accelerator, Random Number Generator Seed Value, Error Parameter, and the Update parameter[5,6]. The Error Parameter applies to an individual surface and is a function of the confidence level.[6] Also, a working knowledge of engineering statistics is helpful in understanding ray-tracing codes.

ANALYSIS AND RESULTS

Figure 2 shows a plot of CPU time on the Apollo DN10000 versus number of rays and error. The default value was used for the other parameters.

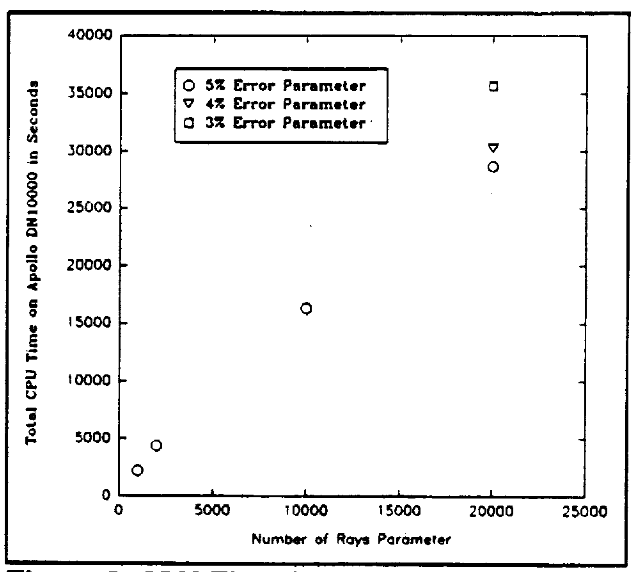

Figure 2 CPU Time Comparison for TSS Parameters

The TSS parameters were successively varied as such: 1000 rays to 20,000 rays with a 5 % Error Parameter, then 20,000 rays with a 4% Error Parameter, and finally 20,000 rays with a 3% Error Parameter. The TSS analyst can effectively use a "restart" file to build upon calculations already performed in previous analyses. For TRASYS, the analyst can increase the accuracy for a particular surface, but the calculations effectively start over for that surface.

As the number of rays per surface increases, the error will decrease. TSS will generate rays from a particular surface until either the Number of Rays or the Maximum Error Parameter is reached. This condition is checked as often as is required by the Update Parameter. For 1000 rays and 5% error, the Number of Rays is the limiting parameter for most surfaces in this problem. For 20,000 rays and 5% error, the Error Parameter is limiting the calculations for some surfaces and fewer than 20,000 rays are generated for many surfaces. As the Number of Rays parameter is increased, it is more likely that the Error Parameter will control the calculations of most surfaces in the problem. The user has the option of forcing practically all surfaces to the same error by setting the Number of Rays equal to a very large number.

If the Error Parameter had the controlled the calculations, the plot would show that the percentage of CPU time increases as the square of the improvement in error. The roughly linear relationship of CPU time to Number of Rays shown in Figure 2 is expected. In this case, the error is different for each surface.

Figure 3 is a close-up view of an instrument which shows a 4-sided boxed protruding through the mounting structure. All surfaces are facing outward. Note that a portion of the active surfaces of the central box view the inactive surfaces of the prism-shaped mounting structure. Also, note that this structure sits directly on a larger rectangle which forms the payload support wall, an example of the box-on-a-plate problem (see Figure 1). This is a typical construction found in the Matra GMM which cannot be handled well by TRASYS, but is acceptable to the Matra ray-tracing package, THERMICA.

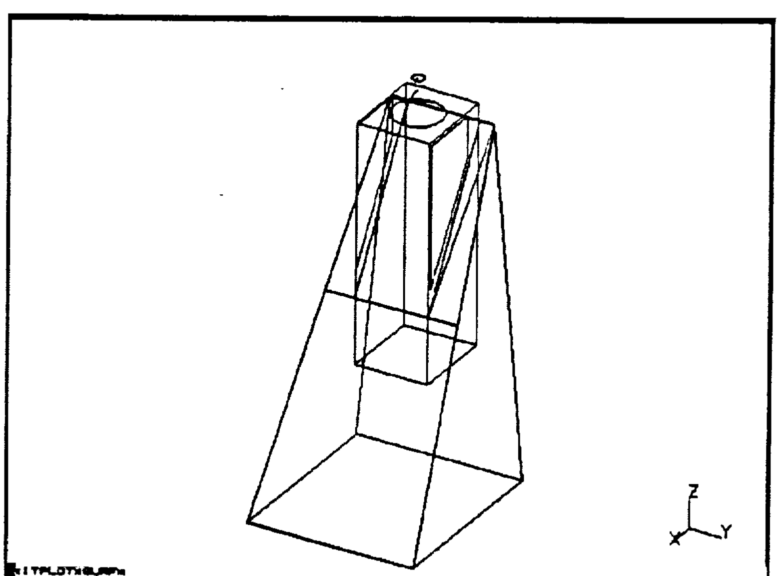

Figure 3 SWAN Instrument GMM

Figures 4 and 5 show the distribution of the difference between temperatures from successive analyses using various TSS and TRASYS RADK's.

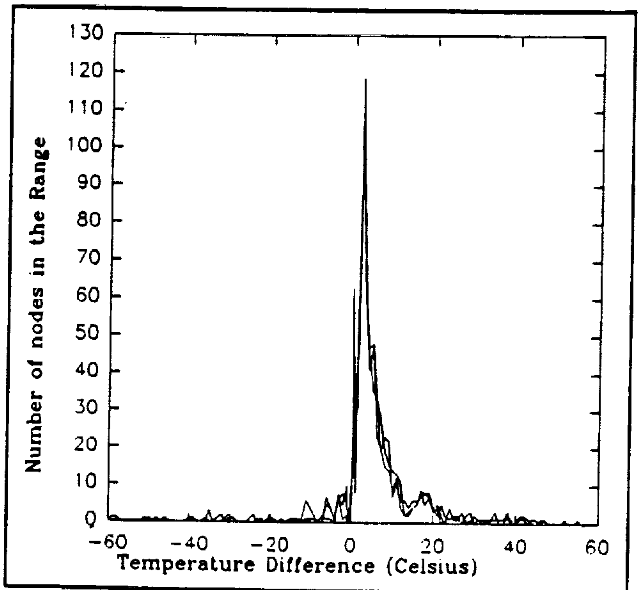

Figure 4 TSS Temperatures Minus TRASYS Temperatures

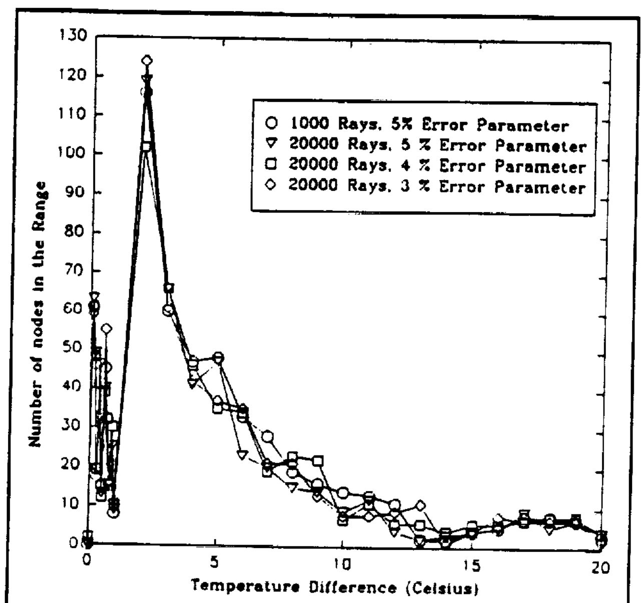

Figure 5 Expanded View of Figure 4

To get this plot, the temperatures produced by TRASYS RADK's were subtracted from those produced by TSS. If the two methods gave exactly the same answers, there would be nothing but a peak at 0 °C. There seems to be a distinct shift in a majority of temperatures by about 2 to 12 °C warmer for most of the TSS values. This is because TSS has greater view factors to the spacecraft surfaces and therefore smaller views to space. The original TRASYS results showed form factors sums for some surfaces to be far from 1.0. Also, for TSS, weighted error results for several small form factors exceeded 5%. Therefore, for this problem and perhaps for many other similar situations, using TRASYS to generate form factors from a GMM originally constructed for use with a ray tracing code may yield generally colder temperatures as well as in some cases, significant errors, both warmer and colder. Figure 5 clearly shows that successive analyses of TSS using more rays and a smaller error parameter do not produce significantly different results. Therefore, 1000 rays with a 5% Error Parameter seems to be adequate for the majority of the surfaces for a model of this size and type. Of course, different spacecraft models may require more rays.

CONCLUSION

Although the converted SOHO model did contain what TRASYS-trained analysts might call "errors," the model was acceptable to a ray-tracing type code. TRASYS did a surprisingly good job on most of the surfaces. However, results of the two codes do differ for a significant number of surfaces. This model should be reworked if a ray tracing code is not used. Also, as a result of the work on the SOHO project, an ESABASE-to-TRASYS FORTRAN conversion program is available at NASA-LeRC.

If a "cookbook" set of parameters to use with various types and sizes of typical space vehicle thermal models could be provided, this would reduce the confusion and ease the transition to TSS or any ray-tracing program. If, however, TSS and other ray tracing codes must be carefully optimized for each particular spacecraft and situation, or if a ray tracing user must thoroughly understand advanced radiation heat transfer and engineering statistics, the popularity of TSS and other ray tracing codes may be much more limited than TRASYS.

References

1. Baumeister, Joseph F., NASA/LeRC, Thermal and Fluids Analysis Branch, "Application of Ray Tracing in Radiation Heat Transfer," lecture presented at the 1993 Thermal and Fluids Analysis Workshop.

2. Analytix Corporation, Timonium Maryland, ITPLOT User's Guide, December 1990.

3. Sagerman, Gary D., SOHO Mission Manager, NASA/LeRC Launch Vehicle Project Office, personal communication, June 1993.

4. Tsuyuki, Glenn T., Jet Propulsion Laboratory, California Institute of Technology, "Advanced TRASYS" shortcourse at the Thermal Fluids Analysis Workshop.

5. Rickman, Steven L., ES34/Thermal Analysis Section, NASA/JSC,"TSS Tutorial," February 1993.

6. Lockheed Missiles and Space Company, Sunnyvale, California, "User Manual, Thermal Synthesizer System," December 18, 1992.