2008

A thermal analysis and design tool for small spacecraft

Cassandra Belle VanOutryve
San Jose State University

```
end
xlabel('Time (s)')
ylabel('Temperature (k)')
legend(leg)
hold off

%Reshape temp array for output in 2D matrix form
S=size(temp);
temp=reshape(temp,S(1),S(3));

end
```

A THERMAL ANALYSIS AND DESIGN TOOL FOR SMALL SPACECRAFT

A Thesis

Presented to

The Faculty of the Department of Mechanical and Aerospace Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Cassandra Belle VanOutryve

December 2008

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

SAN JOSÉ STATE UNIVERSITY

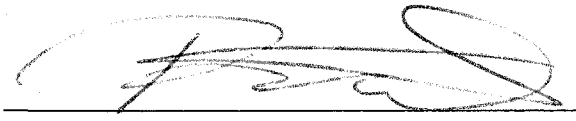
The Undersigned Thesis Committee Approves the Thesis Titled

A THERMAL ANALYSIS AND DESIGN TOOL FOR SMALL SPACECRAFT

by

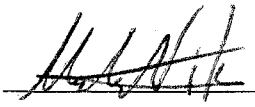
Cassandra Belle VanOutryve

APPROVED FOR THE DEPARTMENT OF
MECHANICAL AND AEROSPACE ENGINEERING



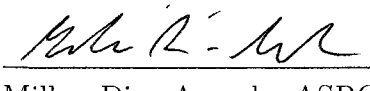
11/4/08

Dr. Periklis Papadopoulos, Dept. of Mechanical & Aerospace Engineering Date



11-06-08

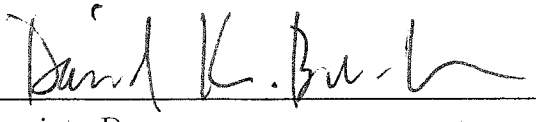
Dr. Nikos Mourtos, Dept. of Mechanical & Aerospace Engineering Date



11/10/08

Millan Diaz-Aguado, ASRC Research and Technology Solutions, Inc. Date

APPROVED FOR THE UNIVERSITY



11/17/08

Associate Dean Date

ABSTRACT

A THERMAL ANALYSIS AND DESIGN TOOL FOR SMALL SPACECRAFT

by Cassandra Belle VanOutryve

SatTherm is a simple thermal analysis tool for small spacecraft that has been developed in collaboration with the Mission Design Center at NASA Ames and San Jose State University. A comparison of the ease of use and accuracy of results is made between SatTherm, and the commercially available Thermal Desktop software by Cullimore & Ring Technologies. Several benchmark cases are presented, including a model of the small spacecraft PharmaSat. The non-steady temperatures predicted by the SatTherm model agree with those predicted by the Thermal Desktop model within 4 $^{\circ}C$ or less. This demonstrates that SatTherm can be a useful tool for the early design stage of a small spacecraft.

ACKNOWLEDGEMENTS

First and foremost I would like to thank Millan Diaz-Aguado for his willingness to teach me all that he knows about thermal analysis, answer my unending questions, and help debug stubborn code. His overall support of this project was crucial to its success. I would also like to thank Dr. Papadopoulos and Dr. Mourtos for being on my thesis committee. Special thanks goes to Belgacem Jaroux and the NASA Ames Mission Design Center for use of their facilities and providing an excellent learning environment. I am also very grateful to Education Associates Program for their funding and support. Finally, I would like to thank my parents, Dirk and Andrea VanOutryve, and my future husband, Tom Allison, for their untiring love and support that has been so important to me through the years, and will continue to be for years to come.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

Proper temperature control within a spacecraft is critical to the success of a mission. Every component of the the payload and spacecraft bus has required temperature limitations. These requirements may include an operation and a survival temperature range, which, if exceeded, may result in reduced performance and/or permanent damage to the component. Electrical devices will not work properly or may have a shortened life span if they overheat. Battery efficiency decreases if the temperature is off nominal or if there is a significant temperature difference between battery cells. Liquid hydrazine will freeze in the fuel lines if it gets too cold, making it impossible to get fuel to the thrusters. Large temperature gradients can also deform the spacecraft structure, possibly leading to significant pointing errors. These are just a few of the mission-killing problems that may occur if temperatures are left uncontrolled (Gilmore, Hardt, Prager, Grob, & Ousley, 2006).

The Thermal Control System of a spacecraft is responsible for maintaining temperatures within the required limitations. Throughout the various stages of the spacecraft design process, a thermal analysis, or prediction of key component temperatures, should be performed. There are a variety of commercially available software programs, such as Thermal Desktop by Cullimore & Ring Technology and Thermica by Network Analysis Inc., that can be used to create detailed thermal models and predict the temperature of a spacecraft in orbit. While the complexity of a model may increase its ability to accurately predict temperatures at precise locations within a spacecraft, it can also become a hindrance. Detailed spacecraft models are difficult

and time consuming to build without human error. Although software programs, such as Thermal Desktop and Thermica, are very powerful for a wide range of situations, they require a significant investment of time to learn how to use.

The Mission Design Center is a branch within the Small Spacecraft Group at the NASA Ames Research Center. One goal of the Mission Design Center is to streamline the small spacecraft design process, and it is in the process of creating a set of tools to do so. In collaboration with the Mission Design Center and San Jose State University, a tool has been developed to simplify the process of small spacecraft thermal analysis and thermal control system design. This thermal analysis program is called SatTherm. It has been designed to be an easy-to-use tool to produce a simple thermal model of a small spacecraft. It can be used to quickly approximate the average temperature of spacecraft components, which is especially useful in the early design stages of a spacecraft mission.

This report details the scientific background needed to understand thermal analysis in general and specifically how the SatTherm program predicts spacecraft temperatures (Ch. 2). It also gives a user's manual-style description of the structure of SatTherm and a comprehensive explanation of each input required to build a thermal model in SatTherm (Ch. 3). Finally, a validation of SatTherm is presented in the form of several benchmarking cases, comparing results found through the use of SatTherm to those found through the used of Thermal Desktop (Ch. 4).

CHAPTER 2

THERMAL ANALYSIS

2.1 Spacecraft Heat Transfer

A combination of the three modes of heat transfer (conduction, convection, and radiation) can be used to describe the flow of thermal energy into, out of, and within a spacecraft.

2.1.1 Conduction

Conduction is the transfer of energy through a material as a result of interactions between particles in that material (Cengel, 1998). Conduction heat flow $[W]$ between two points within a material, i and j, can be described mathematically as

$$\dot{Q}_{ij} = \frac{kA}{L}(T_i - T_j) \tag{2.1}$$

where k is the thermal conductivity $[\frac{W}{m \cdot K}]$ of the material, A is the cross-sectional area through which the heat is flowing, L is the distance between the two points, and T is the temperature of each point in Kelvin. An analogy can be made between heat flow and electrical circuits (Chapman, 1987). In this analogy, heat flow, $\dot{Q}$, corresponds to electrical current, and a temperature difference, $T_i - T_j$, corresponds to a potential difference. A thermal resistance for heat flow by conduction can be defined as

$$R_{ij} = \frac{L}{kA} \tag{2.2}$$

and Eqn. 2.1 can be rewritten as

$$\dot{Q}_{ij} = \frac{T_i - T_j}{R_{ij}} \tag{2.3}$$

displaying its similarity with Ohm's Law for electrical circuits.

Conduction not only transfers heat within an object, but also between objects that have touching surfaces. Conduction between surfaces is characterized by a term called contact conductance, C_{cond}, which has the units $[\frac{W}{K}]$. When calculating the contact heat flow, the contact conductance, C_{cond}, can be put in place of the term $\frac{kA}{L}$ in Eqn. 2.1. The value of the contact conductance strongly depends on the material of the objects, the size of the area that is in contact, the waviness and roughness of the surfaces, the strength of the pressure holding the surfaces together, and the uniformity of that pressure. Estimating the value of the contact conductance can become complicated and is discussed in much detail by Gluck and Baturkin (2002).

2.1.2 Convection

Convection is the transfer of energy between a solid and a neighboring gas or liquid that is in motion. It includes both the effects of adjoining particle interaction and the physical movement of particles (Cengel, 1998). The convective heat flow between two points, i and j, is

$$\dot{Q}_{ij} = hA(T_i - T_j) \tag{2.4}$$

where h is the convective heat transfer coefficient $[\frac{W}{m^2 \cdot K}]$ between the two points. Continuing with the electrical circuit analogy, the thermal resistance for convection is defined as

$$R_{ij} = \frac{1}{hA} \tag{2.5}$$

and Eqn. 2.4 takes the same form as Eqn. 2.3.

For small spacecraft applications, convection is not frequently seen because most components are in vacuum. However, convection may be needed to calculate heating within fluid fuel tanks or pressurized biological experiments on a spacecraft.

2.1.3 Radiation

Radiation is the transfer of energy through the emission and absorption of photons.

2.1.3.1 Properties of a Radiating Surface

The amount of energy emitted from a surface at a specific wavelength is a function of the temperature and depends on the type of material and the form of the surface. The starting point for describing how objects radiate is the ideal blackbody. It absorbs all incident light and re-emits it according to Plank's Distribution Law, which is purely a function of wavelength and termperature. The amount of energy per unit time per unit area, integrated over all wavelengths, is called the total black body emissive power, E_b, which is expressed as

$$E_b = \sigma T^4 \tag{2.6}$$

where σ is the Stefan-Boltzmann constant, $5.669 \times 10^{-8}\ \frac{W}{m^2 K^4}$ and T is temperature. Eqn 2.6 is called the Stefan-Boltzmann Law (Cengel, 1998). Of course, no real surface is a perfect blackbody. Real surfaces are characterized by their emissivity, ϵ, such that emissive power, E, is

$$E = \epsilon \sigma T^4 \tag{2.7}$$

In practice, it is observed that the emissivity of a material is also a function of temperature, wavelength, and direction, but can be approximated as diffuse (its properties are independent of direction) and grey (its properties are independent of wavelength). For example, one may often find the emissivity of a material listed for both the visible and the infrared regime of light, and it can be considered constant within each of these regimes.

Other important properties of a real surface are its absorptivity, α, reflectivity, ρ, and transmissivity, τ, which are the fraction of the incident light on a surface that is absorbed, reflected, and transmitted, respectively. These three are related such that

$$\alpha + \rho + \tau = 1 \tag{2.8}$$

If the material is opaque, τ=0 and

$$\alpha + \rho = 1 \tag{2.9}$$

When a surface is in thermal equilibrium, Kirchoff's Law implies that the emissivity equals the absorptivity (Cengel, 1998).

$$\alpha = \epsilon \tag{2.10}$$

2.1.3.2 Radiation View Factors

When considering the transfer of radiative energy between two different surfaces, one must first consider the orientation of the two surfaces relative to each other. The view factor (also known as the configuration factor or shape factor) is used to characterize this orientation. The view factor, F_{i-j}, is defined as the fraction of radiation emitted by surface i that is intercepted by surface j (Gilmore & Collins, 2002). Similarly, F_{j-i} would be the fraction of radiation emitted by j that is intercepted by i. An important symmetry relationship exists between these two fractions:

$$A_i F_{i-j} = A_j F_{j-i} \tag{2.11}$$

where A is the area of the surface. Therefore, the view factor depends on the geometric configuration of the two surfaces with respect to each other. Fig. 2.1 shows two differential areas dA_i and dA_j. The vectors normal to each surface are n_i and n_j. S

is the shortest line between the surfaces, and θ_i and θ_j are the angles between the normal vectors and the line S.

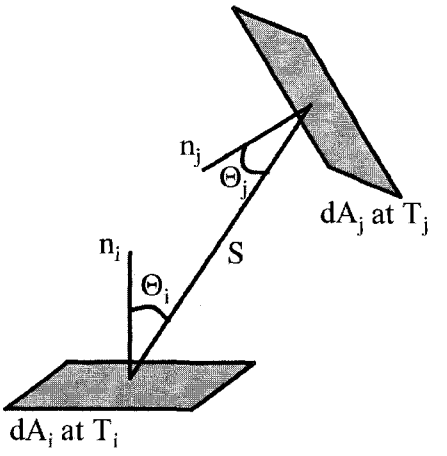


Figure 2.1: Geometry for calculating the view factor between two differential areas

Siegel and Howell (1996) show that the view factor between two differential areas is

$$F_{di-dj} = \frac{cos\theta_i cos\theta_j}{\pi S^2} dA_i dA_j \tag{2.12}$$

The exact solution for the view factor between two finite areas, A_i and A_j, is found by integrating Eq. 2.12 over the two areas

$$F_{i-j} = \frac{1}{A_i} \int_{A_i} \int_{Aj} \frac{cos\theta_i cos\theta_j}{\pi S^2} dA_i dA_j \tag{2.13}$$

While this double integral becomes quite complicated for complex geometry, the shape factor for many simple geometric configurations have been approximated and catalogued. Appendix A provides the view factors for several configurations that are relevant to thermal analysis of small spacecraft.

2.1.3.3 Radiation Exchange Between Surfaces

Once the view factor has been determined, the net radiated heat flow between two surfaces can be calculated. We will assume the surfaces are diffuse, grey, and opaque. The total radiation energy leaving a surface includes both emitted and reflected radiation and is called the radiosity, J, of the surface. If an analogy is made to Ohm's Law for electric circuits then the net heat flow, $\dot{Q}_i$, through a surface i can be written as

$$\dot{Q}_i = \frac{E_{bi} - J_i}{R_i} \tag{2.14}$$

where

$$R_i = \frac{1 - \epsilon_i}{A_i \epsilon_i} \tag{2.15}$$

is called the surface resistance (Cengel, 1998), and A is the surface area. $\dot{Q}$ is analogous to electrical current, and the term $E_{bi} - J_i$ corresponds to a potential difference. Note that for a blackbody, the surface resistance, $R_i = 0$, because $\epsilon_i = 1$ and $E_{bi} = J_i$.

Continuing with the electrical analogy, the net heat transfer between the two surfaces, $\dot{Q}_{i \to j}$, is

$$\dot{Q}_{i \to j} = \frac{J_i - J_j}{R_{i \to j}} \tag{2.16}$$

where

$$R_{i \to j} = \frac{1}{A_i F_{i-j}} \tag{2.17}$$

can be thought of as the space resistance. The radiation network between the two surfaces can be thought of as an electrical circuit, consisting of the space resistor, and the surface resistors for each surface. The radiative heat flow through this network behaves like current through an electrical circuit. Fig. 2.2 illustrates such a circuit. As with an electrical circuit, the total resistance of the network is simply the sum of the three individual resistances, $R_{total} = R_i + R_{i \to j} + R_j$. Note that, for radiation, the

potential is quartic in temperature, T, as opposed to linear (as it is for conduction and convection). Cengel (1998) shows that the heat flow from surface i to surface j is

$$\dot{Q}_{ij} = \frac{\sigma(T_i^4 - T_j^4)}{\frac{1-\epsilon_i}{A_i\epsilon_i} + \frac{1}{A_i F_{i-j}} + \frac{1-\epsilon_j}{A_j\epsilon_j}} \tag{2.18}$$

This is assuming that the two surfaces form an enclosure, meaning that none of the radiation is lost to other surfaces or to space.

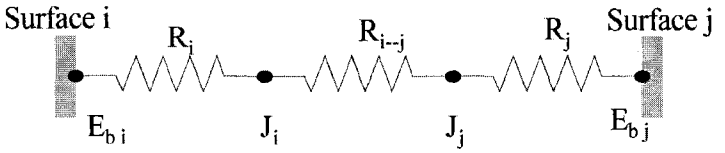


Figure 2.2: The radiation network between two non-blackbody surfaces, consists of a space resistance, $R_{i\to j}$, and a surface resistance at each of the surfaces, R_i and R_j.

However, because spacecraft usually have more than two surfaces, one must consider the radiation heat flow between multiple surfaces. To calculate the net heat flow into or out of one surface, i, it is tempting to just single out each of the other surfaces, one at a time, calculating the radiation heat flow between surface i and that surface, and then sum this amount for all of the surfaces to obtain a total heat flow. This method is incorrect, though, because the heat flowing from surface j, to surface i not only includes the radiation emitted from surface j, but may also include the radiation emitted from several other surfaces and reflected off of surface j. The radiation network for multiple surfaces can be expanded from that for only two surfaces, as shown in Fig. 2.2. As the number of surfaces grows, so does the complexity of the radiation network.

For simplicity, one can first consider a three surface enclosure, and then expand this to a generic situation of n surfaces. Fig. 2.3 displays the appropriate radiation network for a three surface enclosure. There is a surface resistance for each of the three

surfaces, R_1, R_2, and R_3 (defined by Eqn. 2.15), and a space resistance between each, $R_{1\to2}$, $R_{1\to3}$. and $R_{2\to3}$ (defined by Eqn. 2.17). One can see that there are several paths that the radiation can take. For example, light from surface 3 can reach surface 1 directly, or by reflection off of surface 2.

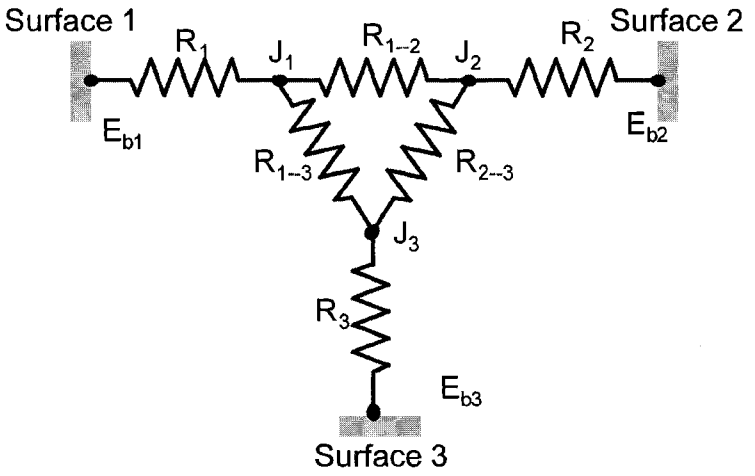


Figure 2.3: The radiation network between three non-blackbody surfaces consists of a surface resistance at each of the surfaces, R_1, R_2, and R_3, and space resistances between each surface, $R_{1\to2}$, $R_{1\to3}$. and $R_{2\to3}$.

Returning to the electrical circuit comparison, the resistors are not all in series, so one can not say that the total resistance is the sum of the surface and space resistances (as was the case for a two surface enclosure). Cengel (1998) demonstrates the following method for determining the heat flow between the three surfaces. Since the emissive powers, E_{b1}, E_{b2}, and E_{b3}, are known (using Eqn. 2.6), and the resistances are known (using Eqns. 2.15 and 2.17), there are only three unknown values: J_1, J_2, and J_3. To solve for these, one makes use of the requirement that the heat flow into a circuit node must be equal to the heat flow out of the node (simmilar to current in

an electrical circuit). This leads to a system of three simultaneous equations:

$$\begin{aligned}
\frac{E_{b1}-J_1}{R_1}+\frac{J_2-J_1}{R_{1\to2}}+\frac{J_3-J_1}{R_{1\to3}} &= 0\\
\frac{J_1-J_2}{R_{1\to2}}+\frac{E_{b2}-J_2}{R_2}+\frac{J_3-J_2}{R_{2\to3}} &= 0\\
\frac{J_1-J_3}{R_{1\to3}}+\frac{J_2-J_3}{R_{2\to3}}+\frac{E_{b3}-J_3}{R_3} &= 0
\end{aligned} \tag{2.19}$$

which can be solved for J_1, J_2, and J_3. Then the net radiative heat flow into or out of each of the surfaces is

$$\begin{aligned}
\dot{Q}_1 &= \frac{J_1-J_2}{R_{1\to2}}+\frac{J_1-J_3}{R_{1\to3}}\\
\dot{Q}_2 &= \frac{J_2-J_1}{R_{1\to2}}+\frac{J_2-J_3}{R_{1\to3}}\\
\dot{Q}_3 &= \frac{J_3-J_1}{R_{1\to3}}+\frac{J_3-J_2}{R_{2\to3}}
\end{aligned} \tag{2.20}$$

The radiation network for a three surface enclosure in Fig. 2.3 can be expanded and generalized for an n surface enclosure. The general form of Eqns. 2.19 becomes

$$\frac{E_{bi}-J_i}{R_i}+\sum_{j=1}^{n}\frac{J_j-J_i}{R_{i\to j}}=0 \tag{2.21}$$

which can be rewritten in matrix form and numerically solved for all the Js. The net heat flow for each surface is found by generalizing Eqns. 2.20, which become

$$\dot{Q}_i=\sum_{j=1}^{n}\frac{J_i-J_j}{R_{i\to j}} \tag{2.22}$$

2.2 Space Thermal Environment

Because a satellite in orbit is surrounded by the vacuum of space, its only thermal interaction with its environment is through radiation. There are three main sources of external radiation hitting a spacecraft in Earth orbit. Direct solar radiation hits the panels that face toward the sun, direct Infrared (IR) Radiation from the Earth

hits the panels that face toward the Earth, and reflected solar radiation (or albedo) also hits the panels that face toward the Earth. Fig. 2.4 shows the geometry a spacecraft surface, with a normal vector $\vec{n}$, in orbit around the Earth

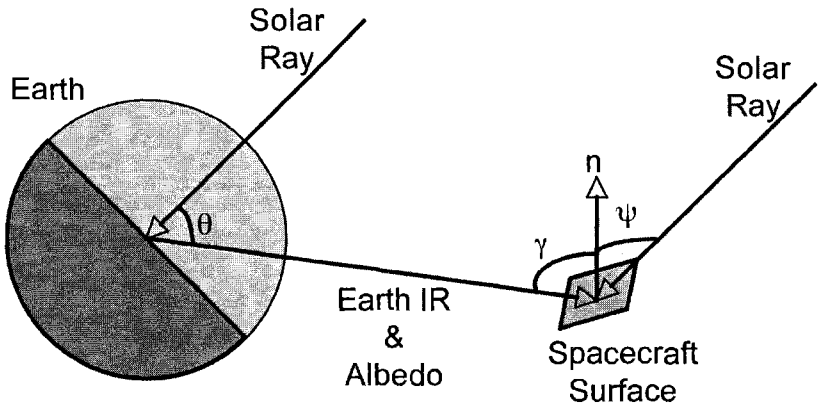


Figure 2.4: A surface on a spacecraft in orbit is exposed to direct solar, direct Earth IR, and reflected solar (albedo) radiation.

2.2.1 Spacecraft Position in Earth Orbit

The amount of external radiation reaching a spacecraft depends on its position with respect to the Earth and the Sun, so knowing the orbit and its position within the orbit is crucial to its thermal analysis. An elliptical orbit can be defined within various sets of coordinate systems. The Geocentric-Equitorial System has its origin at the center of the Earth, with unit vectors $\vec{I}$, $\vec{J}$, and $\vec{K}$ along the x, y, and z axes, respectively (see Fig. 2.5). $\vec{I}$ points toward the direction of Vernal Equinox, $\vec{K}$ points toward the North Pole, and $\vec{j}$ completes the right-handed coordinate system. A different coordinate system, the Perifocal System, has its origin at the center of the Earth, with its fundamental plane in the plane of the orbit. The unit vectors $\vec{P}$, $\vec{Q}$, and $\vec{W}$ lie along the x, y, and z axes respectively. $\vec{P}$ points toward the orbit periapsis, $\vec{W}$ points toward the positive angular momentum vector, normal to the orbit plane,

and $\vec{Q}$ completes the right-handed coordinate system (Bate, Mueller, & White, 1971).

Six classical elements characterize an elliptical orbit:

- The semi-major axis of the ellipse, a
- The eccentricity of the ellipse, e
- The inclination of the orbit, i, which is the angle between vectors $\vec{K}$ and $\vec{W}$, as shown in Fig. 2.5
- The right ascension of ascending node (RAAN), Ω, which is the angle between vector $\vec{I}$ and the point where the spacecraft crosses the Equatorial Plane in a Northerly direction, measured in a counterclockwise direction
- The Argument of Periapsis, ω, which is the angle between the ascending node and the point of periapsis
- The time of Periapsis.

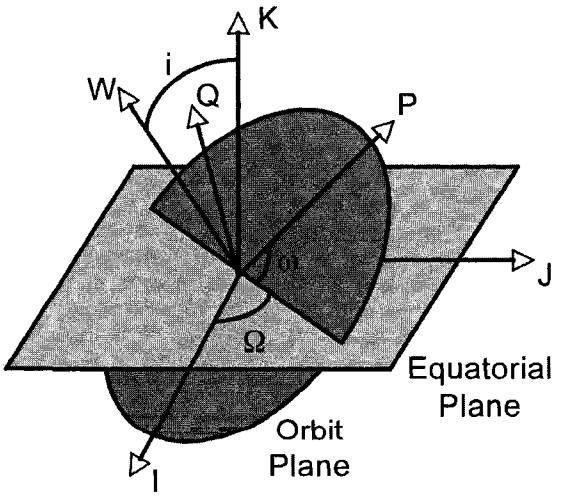


Figure 2.5: The Geocentric-Equitorial Coordinate System is defined by vectors $\vec{I}$, $\vec{J}$, and $\vec{K}$, and the Perifocal Coordinate System is defined by vectors $\vec{P}$, $\vec{Q}$, and $\vec{W}$.

If the eccentricity, e, and maximum altitude above the Earth's surface, h_{max}, are given, then the semi-major axis, a, is

$$a = \frac{R_e + h_{max}}{1 + e} \tag{2.23}$$

where R_e is the radius of the Earth.

There are two other useful elements to characterize the position of the spacecraft within its orbit. Fig. 2.6 illustrates an elliptical orbit with center, C, and primary focal point, F. The ellipse is circumscribed by its auxiliary circle, which has a radius equal to the semi-major axis of the ellipse (Bate et al., 1971). The angle, ν, is called the true anomaly, and the angle, E, is called the eccentric anomaly. Both of these angles are measured from the major axis line in the direction of the spacecraft motion.

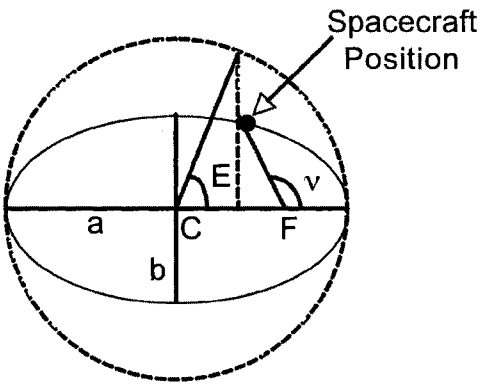


Figure 2.6: An elliptical orbit, circumscribed by its auxiliary circle. The angles ν and E are the true and eccentric anomalies, respectively.

If the time is measured such that the spacecraft is at periapsis at time, t=0, then the relationship between time and eccentric anomaly is

$$t = \sqrt{\frac{a^3}{\mu}}(E - e \sin E) \tag{2.24}$$

where μ is the gravitational parameter of the Earth. Eqn. 2.24 can be solved numerically for the eccentric anomaly, E, at any time, t. The true anomaly, ν, is related to E such that

$$\cos\nu = \frac{e - \cos E}{e\cos E - 1} \tag{2.25}$$

which can be solved for ν, depending on which quadrant the spacecraft is in:

$$\begin{aligned} For E \leq \pi: \quad & \nu = \cos^{-1}\frac{e - \cos E}{e\cos E - 1} \\ For E > \pi: \quad & \nu = 2\pi - \cos^{-1}\frac{e - \cos E}{e\cos E - 1} \end{aligned} \tag{2.26}$$

The distance of the spacecraft from the center of the Earth is the magnitude of the spacecraft-position vector, $|\vec{r}|$, and can be calculated as

$$|\vec{r}| = a\,(1 - e\cos E) \tag{2.27}$$

The components of the spacecraft-position vector in the Perifocal Coordinate System are

$$\begin{aligned} r_P &= |\vec{r}|\cos\nu \\ r_Q &= |\vec{r}|\sin\nu \\ r_W &= 0 \end{aligned} \tag{2.28}$$

To transform the spacecraft-position vector from the Perifocal Coordinate System to the Geocentric-Equitorial System, Bate et al. (1971) show that the vectors $\vec{P}$, $\vec{Q}$, and $\vec{W}$ must be rotated by the angles Ω, i, and ω, respectively. The position vector in Perifocal Coordinates, $\vec{r}_{PQW}$ is matrix multiplied by the 3x3 rotation matrix for this transformation, $[R]$, to find the position vector in Geocentric-Eqitorial Coordinates, $\vec{r}_{IJK}$:

$$\begin{bmatrix} r_I \\ r_J \\ r_K \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} r_P \\ r_Q \\ r_W \end{bmatrix} \tag{2.29}$$

Where the elements of the rotation matrix, $[R]$, are defined as

$$\begin{aligned}
R_{11} &= \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i \\
R_{12} &= -\cos\Omega\cos\omega - \sin\Omega\cos\omega\cos i \\
R_{13} &= \sin\Omega\sin i \\
R_{21} &= \sin\Omega\cos\omega + \cos\Omega\sin\omega\cos i \\
R_{22} &= -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i \\
R_{23} &= -\cos\Omega\sin i \\
R_{31} &= \sin\omega\sin i \\
R_{32} &= \cos\omega\sin i \\
R_{33} &= \cos i
\end{aligned} \tag{2.30}$$

Following the steps outlined by Eqns. 2.24 - 2.30 allows one to calculate the position of a spacecraft, $\vec{r}_{IJK}$, in a given orbit at any time, t.

2.2.2 Environmental Radiation

The direct solar energy per unit time per unit area, or radiation flux $[\frac{W}{m^2}]$, at a given location, is called the solar constant, G_s. The value of G_s is usually considered constant throughout an orbit, because the variation in position of a spacecraft within an orbit is insignificant compared to the distance from the Earth to the Sun. At a distance from the sun of 1AU (ie. in an Earth orbit), $G_s = 1367\ \frac{W}{m^2}$ on average, but can range from 1322 $\frac{W}{m^2}$ to 1414 $\frac{W}{m^2}$ depending on the time of year and solar cycle (Clawson et al., 2002). The direct solar radiation flux absorbed by a spacecraft surface is

$$q_s = G_s \alpha_s \cos\psi \tag{2.31}$$

where G_s is the solar constant, α_s is the solar absorptivity of the surface material, and ψ is the angle between the surface normal vector, $\vec{n}$, and the direction of the solar rays, as shown in Fig. 2.4.

The direct Earth IR radiation flux $[\frac{W}{m^2}]$ absorbed by a spacecraft surface is

$$q_e = \sigma T_e^4 \alpha_{IR} F_e \tag{2.32}$$

where σ is the Stefan Boltzmann constant, T_e is the effective blackbody temperature of the Earth (255 K on average), α_{IR} is the IR absorptivity of the material, and F_e is the view factor from the spacecraft surface to the Earth's disk. This view factor is discussed in Sec. 2.1.3 and Appendix A. Also note that because of the relationship stated in Eqn. 2.10 the IR absorptivity is equal to its emissivity ($\alpha_{IR} = \epsilon_{IR}$). The IR flux radiated from Earth and escaping the atmosphere (σT_e^4) can range from a minimum of 218 $\frac{W}{m^2}$ to a maximum of 275 $\frac{W}{m^2}$ depending on the inclination of the orbit and atmospheric conditions (Gilmore et al., 2006).

The albedo factor, AF, is the fraction of solar radiation that is reflected off of the Earth's atmosphere. This fraction can range between 0.18 and 0.55 depending on the orbit and Earth weather (Gilmore et al., 2006). The radiation flux absorbed by a spacecraft surface due to albedo is given by Thornton (1996) as

$$q_a = G_s(AF)\alpha_s F_e \cos\theta \tag{2.33}$$

when $\theta < \pi/2$. When $\theta \geq \pi/2$ it is

$$q_a = 0 \tag{2.34}$$

Here G_s is the solar constant, AF is the albedo factor, α_s is the solar absorptivity, F_e is the view factor from the spacecraft surface to the Earth's disk, and θ is the angle between the solar rays and position vector of the surface (see Fig. 2.4). Eqns.

2.33 and 2.34 are approximate because, when the spacecraft is directly above the terminator (the line between light and shadow on the Earth's surface, $\theta = 90°$), the calculated albedo heat flux would be equal to zero. However, in reality, the spacecraft surface would still be receiving a small amount of albedo radiation.

The total external radiation heat flux, q_{ext}, absorbed by the spacecraft surface is the sum of the direct solar, direct Earth IR, and albedo radiation:

$$q_{ext} = q_s + q_e + q_a \tag{2.35}$$

Along with these three external heat sources, a spacecraft may also be subjected to internal heat sources. Electrical equipment dissipate most of the power they use into heat, so the amount of heat produced on board a spacecraft will depend on the power consumption of its bus and payload.

2.2.3 Eclipse

Depending on its orbit, a spacecraft may spend a fraction of its time in eclipse, shaded from solar radiation by the Earth. During these periods of time, a spacecraft surface would not receive any direct solar or albedo radiation, but would still receive direct Earth IR radiation if it is pointed toward the Earth. If the position of the spacecraft, r_{sc}, is known (see Sec. 2.2.1), and if the position of the Sun is known, then it can be checked whether the spacecraft is in eclipse or not. Assuming that the Sun's rays are parallel as they approach the Earth and that there is no penumbra (area of partial shading), then the Earth creates a cylindrical umbra (area of complete shading) behind it. Fig. 2.7 illustrates the Earth, the spacecraft-position vector, $\vec{r}_{sc}$, at the very edge of the cylindrical umbra, the sun-position vector $\vec{r}_s$, and the line-of-sight vector. The perpendicular line between the central point and the line-of-sight

vector determines the two angles, θ_1 and θ_2, which are

$$\theta_1 = \cos^{-1} \frac{R_e}{|\vec{r}_{sc}|}$$
$$\theta_2 = \cos^{-1} \frac{R_e}{|\vec{r}_s|} \tag{2.36}$$

where R_e is the radius of the Earth. The sum of these two angles define the transition point from sun to shade.

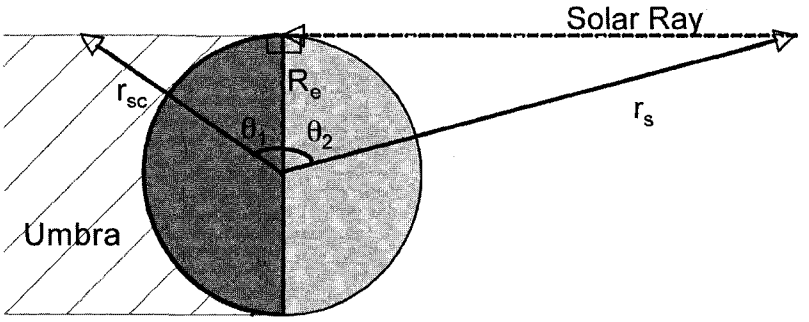


Figure 2.7: Parallel solar rays form a cylindrical umbra behind Earth. When the spacecraft is in the umbra (in eclipse), no line of sight exists between it and the Sun.

The angle, θ_{actual}, between the spacecraft's actual position vector and the sun's position vector is

$$\theta_{actual} = cos^{-1} \left(\frac{\vec{r}_{sc} \cdot \vec{r}_s}{|\vec{r}_{sc}||\vec{r}_s|} \right) \tag{2.37}$$

Vallado (2001) explains that if $\theta_{actual} \geq \theta_1 + \theta_2$ there is no line of sight between the Sun and the spacecraft, meaning that the spacecraft is in eclipse. If $\theta_{actual} < \theta_1 + \theta_2$, line of sight does exist, and the spacecraft is in sunlight.

2.3 Thermal Modeling

2.3.1 The Structure of a Thermal Model

One can create a mathematical thermal model to estimate the thermal behavior of a spacecraft. The spacecraft is broken down into finite subdivisions called nodes,

and the system as a whole can be viewed as a network of these nodes. Each node is described with a temperature and capacity. The thermal capacity (also known as thermal mass) is the product of the node's mass and the specific heat of the material. The properties of a node are considered to be located at a single point (the nodal center), but the values represent the mass average for the corresponding subdivision. Because the subdivisions represent finite volumes, the temperature spatial distribution takes the form of a step function. In reality, within a homogeneous material, the temperature distribution is a continuous function. To estimate a more accurate temperature between node centers, one can linearly interpolate between the node point temperatures (Gilmore & Collins, 2002), or create a finer mesh of nodes.

Some experience is needed to know how many nodes an object should be divided into. In general, the more nodes, the higher the resolution is of the results. However, a large number of nodes increases the complexity of the thermal model, which increases the man-hours required to build it and the computation time to solve for the temperatures.

There are three types of nodes that can be used in a model: diffusion nodes, arithmetic nodes, and boundary nodes (Gilmore & Collins, 2002). Diffusion nodes have a finite thermal capacitance. These represent what one would think of as normal material. Heat flow through a diffusion node results in a change in the node's temperature as described by the following:

$$\sum \dot{Q} = \frac{C\Delta T}{t} \tag{2.38}$$

where $\sum \dot{Q}$ is the net heat flux into or out of the node, C is the capacitance, T is the change in temperature, and t is time. This relationship will be looked at in more detail in Sec. 2.3.2.

Arithmetic nodes are defined as having zero capacitance and the net heat flow

into or out of the node is therefore

$$\sum \dot{Q} = 0 \tag{2.39}$$

No physical object has a truly zero capacitance, but arithmetic nodes can be used to represent objects with very low capacitance compared to their surroundings, or very small objects, such as bolts or fillets (Gilmore & Collins, 2002). The third type of node, the boundary node, is defined as having infinite capacitance. These nodes are used to represent boundaries or sinks, whose temperature remains essentially constant.

After the system has been divided into nodes, these nodes must be networked with the adjacent nodes, using what are called conductors. A conductor may represent conduction between two nodes in a material, or between two objects whose surfaces are touching. It may also represent convection, or radiation between nodes on different surfaces. Once a full thermal model network has been defined and initial conditions given, a transient solution can be found for the temperature at each node throughout a period of time.

2.3.2 The Finite Difference Temperature Solution

To understand how the temperature of an object relates to the various sources of heat flow discussed in Sec. 2.1 and 2.2, first consider the simple case of a one dimensional object with only conduction effects. The temperature, T, of this object is governed by the non-steady conduction equation, which says that

$$k\frac{\partial^2 T}{\partial x^2} = \rho c_p \frac{\partial T}{\partial t} \tag{2.40}$$

where k is the thermal conductivity, ρ is the density, and c_p is the specific heat of the material. This equation shows that the second partial derivative of temperature with

respect to space, x, relates to the partial derivative of temperature with respect to time, t. To solve this partial differential equation for temperature, T(x,t), a numerical approximation technique called the Finite Difference Method is used.

The Finite Difference Method uses the Taylor Series Expansion of a function about small increments of time, Δt, and space, Δx. Ignoring the higher order terms, Chapman (1987) shows that the partial differivatives in Eqn. 2.40 are approximately

$$\frac{\partial T}{\partial t} \approx \frac{1}{\Delta t}\left(T\left(x, t+\Delta t\right) t - T\left(x, t\right)\right) \tag{2.41}$$

and

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{1}{(\Delta x)^2}\left(T(x+\Delta x, t) - 2T(x,t) + T(x-\Delta x, t)\right) \tag{2.42}$$

Now take that one dimensional object and break it into many nodes of increment size Δx, as described in Sec. 2.3.1. Single out one of these nodes, call it node n, and let the nodes on either side of it be called nodes $n-1$ and $n+1$. Then $T(x) = T_n$, $T(x-\Delta x) = T_{n-1}$, and $T(x+\Delta x) = T_{n+1}$. Also, let the temperature at the current time be $T(t) = T$, and let the temperature at a future time (after one incremental step Δt) be $T(t+\Delta t) = T'$. Making these substitutions, Eqn. 2.40 becomes

$$\frac{k}{(\Delta x)^2}(T_{n-1} - 2T_n + T_{n+1}) = \frac{\rho c_p}{\Delta t}(T'_n - T_n) \tag{2.43}$$

Multiplying both sides of this equation by the volume of node n, $V = A\Delta x$, gives

$$\frac{T_{n-1} - T_n}{\frac{\Delta x}{kA}} + \frac{T_{n+1} - T_n}{\frac{\Delta x}{kA}} = \frac{V\rho c_p}{\Delta t}(T'_n - T_n) \tag{2.44}$$

where A is the area of node n, through which the heat is flowing. Substituting the conductive resistance from Eqn. 2.2 and the thermal capacity, $C = V\rho c_p$, we have

$$\frac{T_{n-1} - T_n}{R_{n,n-1}} + \frac{T_{n+1} - T_n}{R_{n,n+1}} = \frac{C}{\Delta t}(T'_n - T_n) \tag{2.45}$$

Chapman (1987) shows that this can be generalized beyond the one dimensional object with nodes n, $n-1$, and $n+1$, to any node i in contact with any other node j, such that

$$\sum_j \frac{T_j - Ti}{R_{ij}} = \frac{C_i}{\Delta t}(T_i' - T_i) \tag{2.46}$$

which can be solved for T_i'

$$T_i' = T_i + \frac{\Delta t}{C_i} \sum_j \frac{T_j - Ti}{R_{ij}} \tag{2.47}$$

Here, T_i' is the future temperature of node i, T_i is the current temperature of node i, Δt is the time-step, and C_i the thermal capacity of i. The term $\frac{T_j - T_i}{R_{ij}}$ is the heat flow into node i from node j (See Eqn. 2.3). Expanding Eqn. 2.47 to also include convection, radiation, and internal heat sources, it becomes

$$T_i' = T_i + \frac{\Delta t}{C_i} \left(\sum_j \dot{Q}_{ij-Cond.} + \sum_j \dot{Q}_{ij-Conv.} + \sum_j \dot{Q}_{ij-Rad.} + \dot{Q}_{i-Int.} \right) \tag{2.48}$$

Therefore, by setting an initial temperature for an object and calculating the net heat flow for each node, one can determine what the temperature of each node will be after a small time-step. Iteration of this calculation leads to the solution for the temperature as a function of time.

2.3.2.1 Stability of the Solution

As mentioned previously the Finite Difference Method utilizes an approximation of the solution to the partial differential equations that govern temperature. Truncation errors come from the fact that high order terms of the Taylor Series Expansion were left out (Chapman, 1987). Terms on the order of the fourth power of Δt and the square of Δx were neglected. Another source of error comes from the rounding of numbers that is necessary in numerical computations and the over simplification of the nodal network.

The stability of the transient solution should also be taken into consideration. Chapman (1987) describes a solution as unstable if the errors at one point in time propagate and are amplified in the time-steps that follow.

By factoring out the T_i terms, Eqn. 2.47 can be written as

$$T_i' = T_i \left(1 - \sum_j \frac{\Delta t}{R_{ij} C_i} \right) + \sum_j \frac{\Delta t T_j}{R_{ij} C_i} \tag{2.49}$$

The term in parentheses can become negative if the time-step is too large compared to the product of the net resistance and capacity of the node. If the term in parentheses is negative, it means that the larger the current temperature, T_i, is then the smaller the future temperature, T_i', would be. This, of course, does not make sense and would cause the solution to oscillate largely from one time step to the next, eventually going to $\pm\infty$. An example of an unstable temperature solution is plotted in Fig. 2.8. Though more detailed studies of stability are available, Chapman (1987), suggests that if the term in parenthesis above is positive, then the Finite Difference solution will be stable. This requires that

$$\sum_j \frac{\Delta t}{R_{ij} C_i} \leq 1 \tag{2.50}$$

For networks of irregularly sized nodes with heat flow through different modes (i.e. conduction, convection, or radiation), the total thermal resistances and the thermal capacity for each node may be different, which would lead to a different requirement on the time step for each node. But since it is necessary to advance each node by the same time-step, the whole system is controlled by the node that requires the smallest time-step. The time-step chosen, should be

$$\Delta t \leq \left(\frac{1}{\sum_j \frac{\Delta t}{R_{ij} C_i}} \right)_{min} \tag{2.51}$$

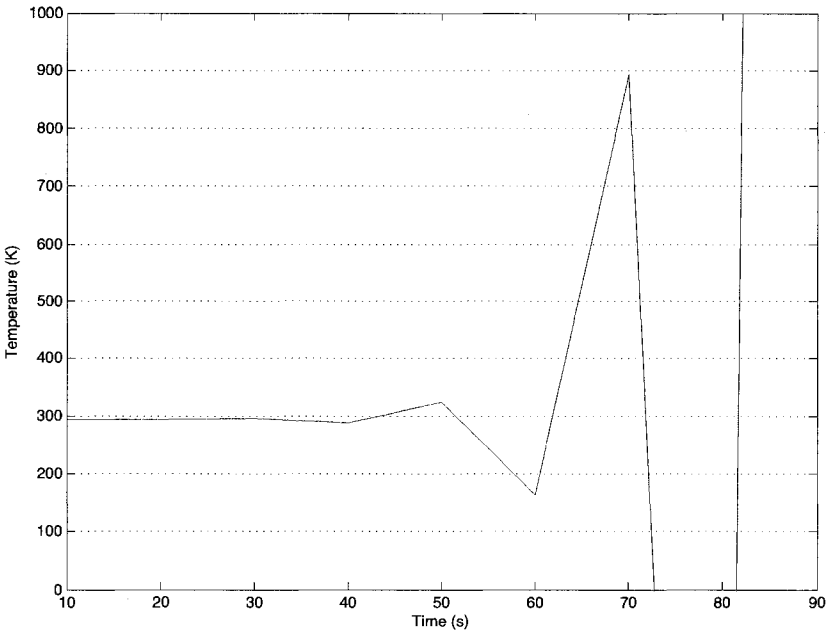


Figure 2.8: If the time-step chosen is too large, the Finite Difference solution will oscillate wildly, eventually going to ∞.

CHAPTER 3

SATTHERM USER'S MANUAL

SatTherm is a simple thermal analysis software tool, designed for small spacecraft in the early design stages of a mission. It uses the finite difference method to solve for the non-steady temperature of spacecraft components. The calculations are done using the Matlab programing language, by MathWorks, while the user-interface is set in Microsoft Excel. Matlab and Excel are connected using the Matlab toolbox called Excel Link, which allows Microsoft's Visual Basic (VBA) to execute commands in Matlab.

SatTherm was designed for simple thermal models, with the total number of nodes being on the order of tens, not hundreds or thousands, as may be the case for more complex models. It was designed such that each spacecraft component, such as a wall, battery, or circuit board be modeled by a single node, not split into a number of nodes. The results found are the average temperatures for individual components.

3.1 Inputs

The Excel based user-interface has four pages of inputs, in which the user can fully define the thermal model. Example screen shots of these four pages are displayed in Figs. 3.1 - 3.4. In general blue cells are inputs that the user must define, and green cells are items that the program calculates, based on the inputs that have been defined. These green items are displayed for use in the temperature calculations and/or for the user's reference, to double check that the model has been accurately defined.

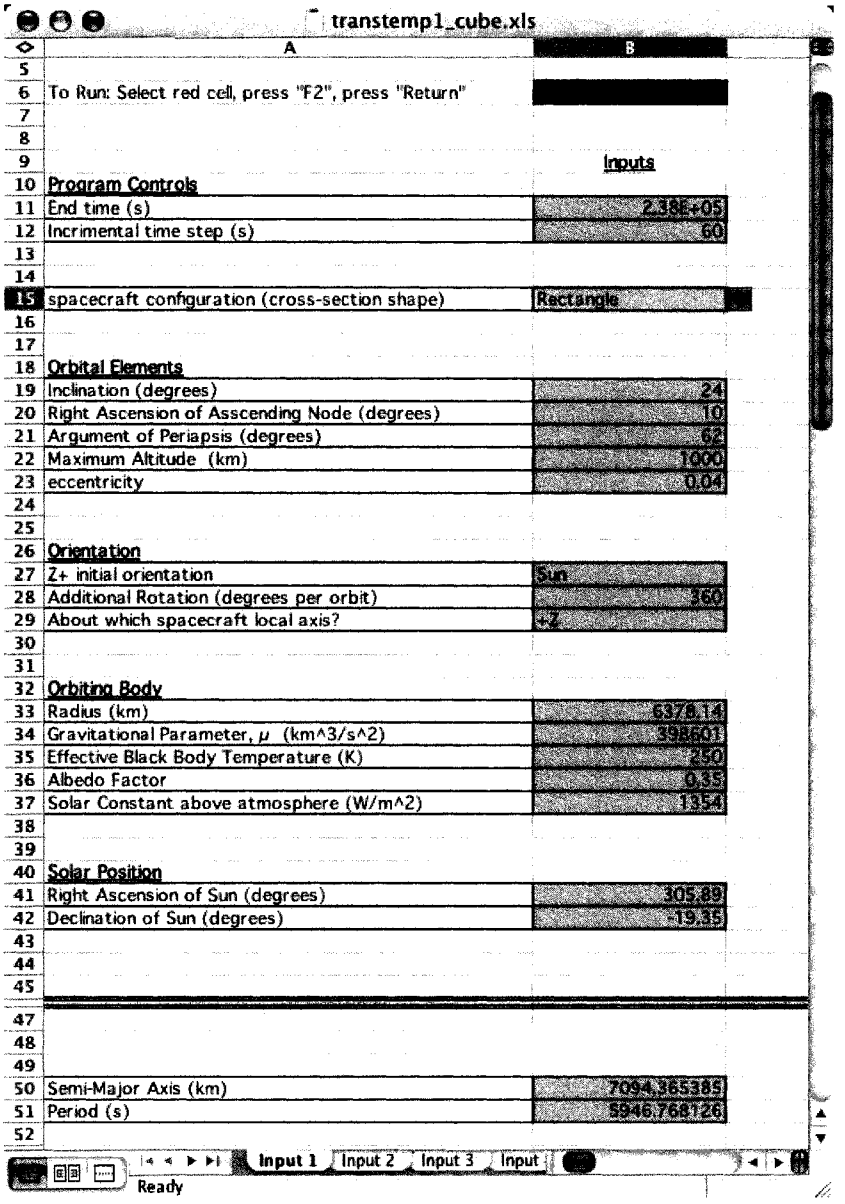
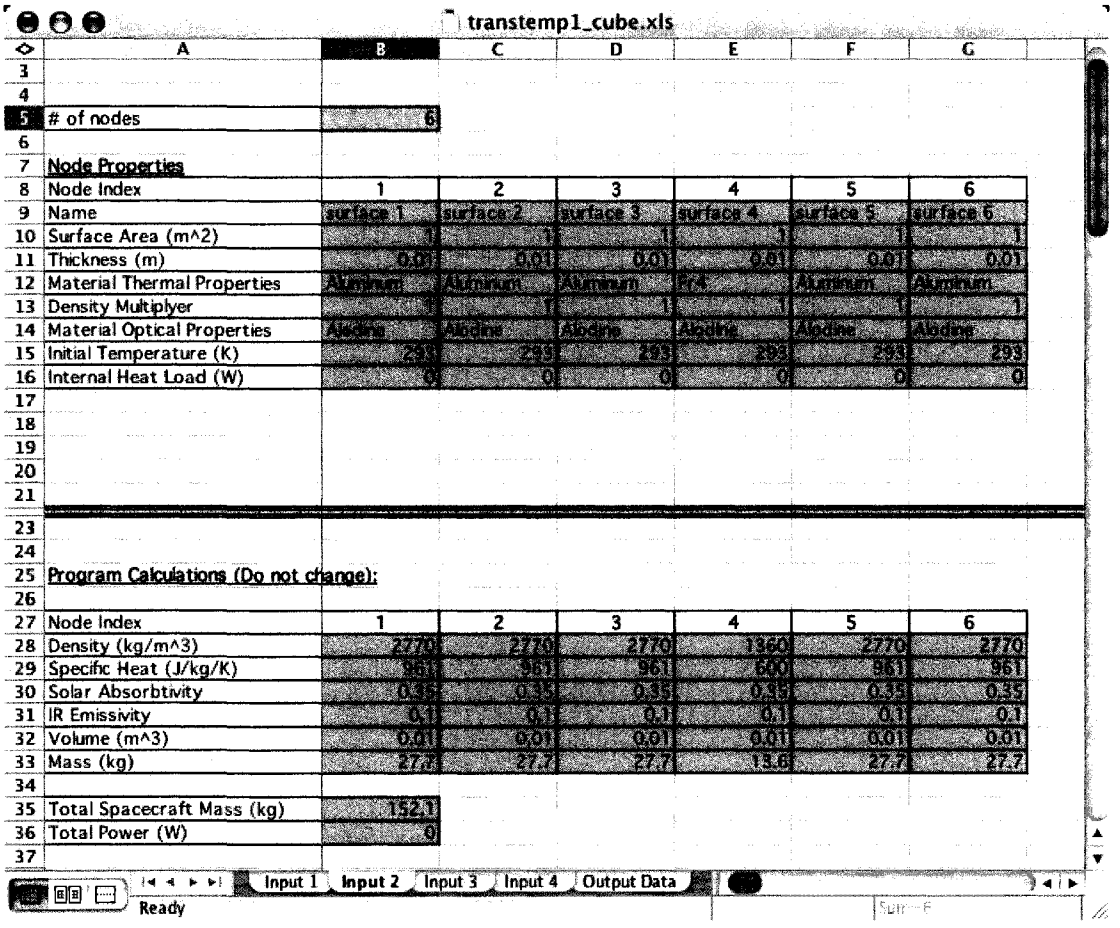


Figure 3.1: A screenshot of the first page of SatTherm inputs. This is where the program controls, orbital elements, spacecraft orientation, the body that is being orbited, and the solar position are defined.



	A	B	C	D	E	F	G
3							
4							
5	# of nodes	6					
6							
7	Node Properties						
8	Node Index	1	2	3	4	5	6
9	Name	surface 1	surface 2	surface 3	surface 4	surface 5	surface 6
10	Surface Area (m^2)	1	1	1	1	1	1
11	Thickness (m)	0.01	0.01	0.01	0.01	0.01	0.01
12	Material Thermal Properties	Aluminum	Aluminum	Aluminum	FR4	Aluminum	Aluminum
13	Density Multiplyer	1	1	1	1	1	1
14	Material Optical Properties	Alodine	Alodine	Alodine	Alodine	Alodine	Alodine
15	Initial Temperature (K)	293	293	293	293	293	293
16	Internal Heat Load (W)	0	0	0	0	0	0
17							
18							
19							
20							
21							
23							
24							
25	Program Calculations (Do not change):						
26							
27	Node Index	1	2	3	4	5	6
28	Density (kg/m^3)	2770	2770	2770	1360	2770	2770
29	Specific Heat (J/kg/K)	961	961	961	600	961	961
30	Solar Absorbtivity	0.35	0.35	0.35	0.95	0.35	0.35
31	IR Emissivity	0.1	0.1	0.1	0.1	0.1	0.1
32	Volume (m^3)	0.01	0.01	0.01	0.01	0.01	0.01
33	Mass (kg)	27.7	27.7	27.7	13.6	27.7	27.7
34							
35	Total Spacecraft Mass (kg)	152.1					
36	Total Power (W)	0					
37							

Figure 3.2: A screenshot of the second page of SatTherm inputs. This is where the nodes and their properties are defined.

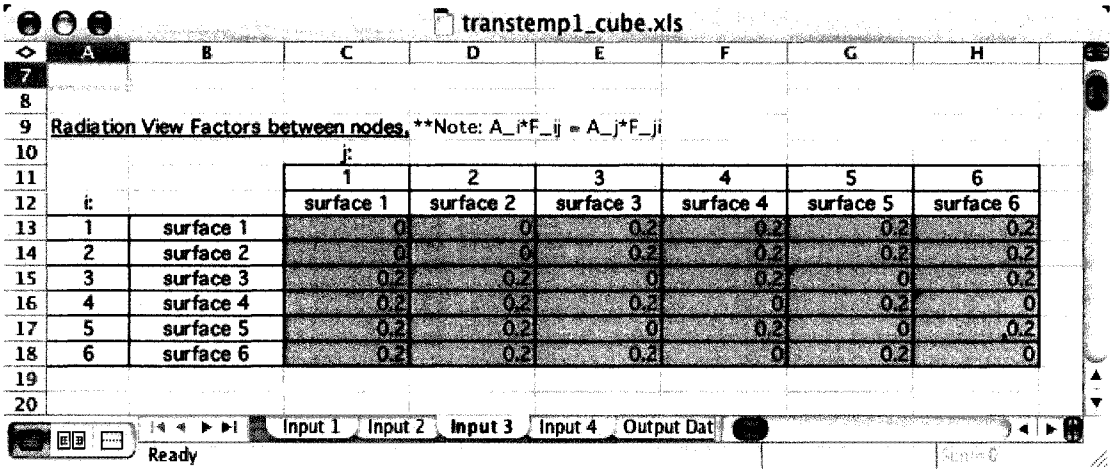


Figure 3.3: A screenshot of the third page of SatTherm inputs. The radiation view factors between each node are defined here.

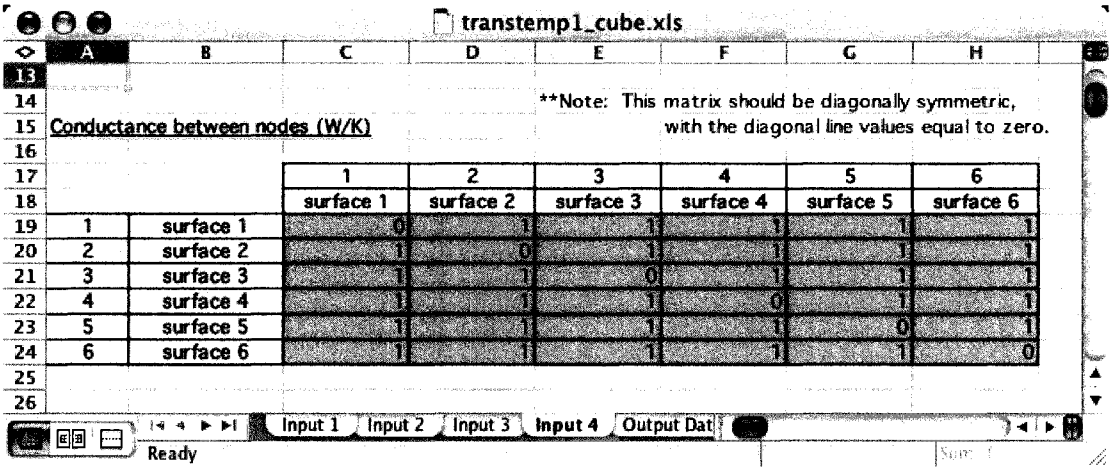


Figure 3.4: A screenshot of the fourth page of SatTherm inputs. The absolute contact conductance between each node is defined here.

The following is a description of each of the inputs that must be defined by the user.

- **Program Controls:**
  - * **End Time:** The length of time that the program will simulate (simulations begin at time $t = 0\ s$) $[s]$.
  - * **Time-Step:** The small increment of time between each temperature calculation $[s]$. Please note the stability criterion for the time-step discussed in Sec 2.3.2.1.
- **Spacecraft Configuration:** The cross-sectional shape of the external walls of the spacecraft. The user can choose from three options in the pull-down menu for this cell: Rectangle (as shown in Fig. 3.1), Hexagon, or Octagon. If the Rectangle option is chosen, the first six nodes will automatically be assigned as the six external walls of the spacecraft (one node for each wall). Node 1 will be assigned to the +Z local axis for the spacecraft, Node 2 will be the side opposite that, at the -Z axis, and the rest of the sides are defined as shown in Fig. 3.5. If the Hexagon or Octagon options are chosen, the first eight or ten nodes will be assigned to the exterior surfaces, respectively. The orientation will be similar to that shown for a rectangular cross-section, with Nodes 1 and 2 assigned to the +Z and -Z local axes, respectively, and the side panels being assigned in a counter-clockwise direction, if looking downward from the +Z axis.
- **Orbital Elements:** Below are the elements needed to define the spacecraft orbit, as described in 2.2.1
  - * **Inclination:** $[degrees]$

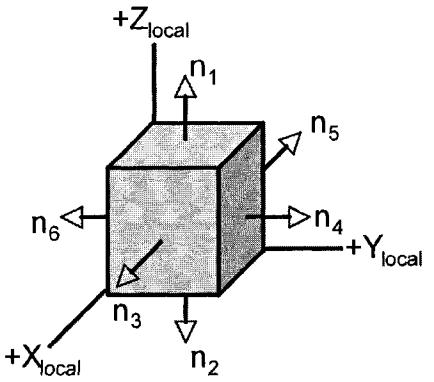


Figure 3.5: If the Rectangle Configuration is chosen, the first six nodes will be assigned to the exterior walls of the spacecraft as shown.

* **Right Ascension of Ascending Node:** [*degrees*]
* **Argument of Periapsis:** [*degrees*]
* **Maximum Altitude:** [*km*]
* **Eccentricity:** (Unitless)

- **Orientation:**
  * **Z+ Initial Orientation:** The user can choose from three options in the pull-down menu for the +Z Initial Orientation of the spacecraft: Sun, Nadir, or Set R.A. and Dec. If the spacecraft local +Z axis is chosen to be Sun facing, then the local +X axis will be set in the plane of the orbit (in the direction of the spacecraft motion), and the local +Y axis will complete the right-handed three-axis system. If the Nadir facing option is chosen, local +Z axis is pointed toward the center of the Earth, the local +X axis will again be set in the plane of the orbit (in the direction of the spacecraft motion), and the local +Y axis will complete the right-handed three-axis system. If the Set R.A. and Dec. option is chosen, the

user will be asked to define a Right Ascension (R.A.) and Declination (Dec.) toward which the local +Z axis will be pointed. The local +X axis will be orthogonal to the +Z axis and in the plane of the orbit, and the +Y axis will complete the right-handed three-axis system. Note that if +Z is normal to the orbit plane any vector in the plane could be orthogonal to +Z. In this case, the +X axis is defined in the direction of spacecraft motion.

  * **Additional Rotation:** For spinning spacecraft, the user can define an additional rotation. The units for the rotation is *degrees per orbit*, so the user would enter 360 for one full rotation per orbit, or 720 for two rotations per orbit, and so on.

  * **Spacecraft Local Axis**: If an additional rotation was defined, the user must choose the local axis about which the spacecraft is spinning (following the right hand rule). The choices from the drop-down menu are: +X, +Y, +Z.

- **Orbiting Body:** Below are the properties of the body that the spacecraft is orbiting. These cells are, by default, set to the average values for the Earth. However, they may be changed to account for natural variations in the radiation environment, or to model an orbit around a different planet or moon.

  * **Radius:** $[km]$

  * **Gravitational Parameter:** $[km^3/s^s]$

  * **Effective Black Body Temperature:** $[K]$

  * **Albedo Factor:** (Unitless)

  * **Solar Constant:** As described in Sec. 2.2.2 $[W/m^2]$

- **Solar Position**

  * **Right Ascension:** $[degrees]$

  * **Declination:** $[degrees]$

- **Number of Nodes:** The total number of nodes that will be included in the model. As described in the discussion of Spacecraft Configuration above, the first several nodes will be automatically defined as the exterior panels of the spacecraft. So when the Spacecraft Configuration is set to Rectangle, Hexagon, or Octagon, the total number of nodes must be at least six, eight, or ten (respectively).

- **Node Properties:** While defining the nodes, it is important to keep in mind that, depending on the Spacecraft Configuration option that was chosen (Rectangle, Hexagon, or Octagon), the first several nodes should be defined as the external side panels, as described in the discussion of Spacecraft Configuration, above.

  * **Name:** Each node is automatically assigned a number, but should also be given a descriptive name, such as "Battery," or "Side Wall."

  * **Surface Area and Thickness:** SatTherm is limited to two-dimensional nodes, meaning that the shape of each node should be thought of as a very thin plate with a surface area, and a thickness. The surface area is used to determine the amount of radiation heat flux into and out of the node, and both the area and thickness are used to determine the nodes volume and therefore its mass. For thin components, such as walls or

circuit boards, this two-dimensional shape is intuitive, but for objects that are very three-dimensional, the surface area and thickness defined should be approximated such that the node has roughly the same surface area and volume as the actual component. An example of how this is done is given in Sec. 4.3.1, in the discussion of modeling the PharmaSat cylindrical batteries.

* **Material Thermal Properties:** There is a built-in database of commonly used materials and their thermal and optical properties. If a needed material is not in the database, the user can define his own properties and add it to the database. The thermal material can be chosen for each node from a drop-down list. Once the material is chosen, the program automatically assigns the appropriate density, ρ, and specific heat, c_p, to that node.

* **Density Multiplier:** This adjusts the density of the material so that the total mass of the node can be adjusted, if necessary. If no adjustment is needed, the density multiplier should be left as the default value of 1.

* **Material Optical Properties:** Like the thermal material, the optical material can be chosen for each node from a drop-down list. Once the material is chosen, the program automatically assigns the appropriate solar absorptivity, α, and IR emissivity, ϵ, to that node. Note that the optical material may be different from the thermal material if, for example, the node is modeling an aluminum plate with a white paint coating.

* **Initial Temperature:** The temperature of the node at the start of the calculation $[K]$. The default value is 273 K, or approximately room

temperature.

* **Internal Heat Load:** Heat produced within the node [W].

- **Radiation View Factors Between Nodes:** The radiation view factors between each node are defined within a matrix with column and row headers for each of the nodes in the model (as seen if Fig. 3.3). The matrix row number represents i, and the column number represents j, where the view factor F_{ij} is the fraction radiated from surface i, that is absorbed by surface j. For example, the cell in the third row and first column is F_{13}, while the cell in the first row and third column is F_{31}. When filling in the view factor matrix, the user should remember the relationship described by Eqn. 2.11. Values of the view factors can be approximated using the methods discussed in Appendix A.

- **Contact Conductance Between Nodes:** The contact conductance between each node is input in to a matrix with column and row headers for each node (as seen in Fig. 3.4). This matrix should be diagonally symmetric because the conductance between say Nodes 1 and 2 will be the same as between Nodes 2 and 1. Also, the diagonal line values of the matrix should be equal to zero because a node should not be conducting to itself.

Finally, after all the inputs have been defined, the program can be run. The red cell, labeled "To Run:" at the top of the first page (Fig. 3.1) acts as a button to execute the non-steady temperature calculations. After all the inputs have been entered, the user should select the red cell, hit the F2 key, and then hit Return. This will cause the VBA module to feed the inputs to Matlab, run the Matlab script, and display the temperature data as a function of time for each of the nodes.

3.2 Program Structure

The SatTherm Matlab code includes of a number of functions that are provided in Appendix B for reference purposes. The program consists of a large loop, performing the necessary calculations for each point in time, then incrementing forward by one time-step and repeating the process.

First the program calculates the position of the spacecraft using the process outlined in Sec. 2.2.1, assuming that the spacecraft is at periapsis at time $t = 0$ s. Next the orientation of the spacecraft determines the local axes, defined in the Geocentric-Equatorial Coordinates. If the spacecraft local +Z axis is Sun facing, the local +X axis will be set in the plane of the orbit (in the direction of the spacecraft motion), and the local +Y axis will complete the right-handed three-axis system. If it is Nadir facing, the local +Z axis is pointed toward the center of the Earth, the local +X axis will again be set in the plane of the orbit (in the direction of the spacecraft motion), and the local +Y axis will complete the right-handed three-axis system. If +Z is set to a specified R.A. and Dec., the local +X axis will be orthogonal to +Z and in the plane of the orbit, and the +Y axis will complete the right-handed three-axis system. Note that if +Z is normal to the orbit plane any vector in the plane could be orthogonal to +Z. In this case, the +X axis is defined in the direction of spacecraft motion.

The initial normal vector of each of the exterior surfaces is defined in the spacecraft local coordinate system, such that Node 1 coincides with the +Z local axis for the spacecraft, Node 2 is the side opposite that, at the -Z axis, and the rest of the sides are assigned numerically in a counter-clockwise direction when looking downward from the +Z axis as shown previously in Fig. 3.5. If an additional rotation was specified by the user, the normal vectors are then rotated accordingly. Finally, the

normal vectors are converted from the spacecraft local coordinates to the Geocentric-Equatorial coordinates using a rotation matrix, as described by Eqns. 2.29 and 2.30.

Once the orientation of the external surfaces is known, the environmental radiation heat flow into each of these nodes is calculated as discussed in Sec. 2.2.2. The net heat flow through each node due to conduction is calculated using a sum of Eqn. 2.3 for each of the node's contacts. The net heat flow through each node due to the radiation network inside the spacecraft is calculated following the procedure outlined in Sec. 2.1.3.

Next the program calculates the new temperature of each node as described by Eqn. 2.48. Before incrementing forward in time, the program checks that the time-step size satisfies the stability requirement defined by Eqn. 2.51. If the time-step does satisfy this requirement, the program goes on using this time-step, but if the time-step is too large, it will be reset to 0.9 times the value of the maximum allowable time-step, and that will be used in the next loop. If the initial guess for the time-step is much larger than the maximum allowable one, the program will run slowly and the accuracy of the results may be skewed. Therefore it is recommended that the first time a model is run, it should be run for a short period of time. If the program reassigns the time-step (which can be determined from the output data), the user should reset the initial input time-step to the reassigned value (or slightly smaller), and then the program can be rerun for a full analysis.

The program loops through time, until it reaches the end time specified by the user. The outputs are the temperature of each node at each of the time points. This data is automatically plotted in a Matlab figure, and is also listed in the Outputs page of the Excel file.

CHAPTER 4

SATTHERM VALIDATION

To verify the accuracy of the SatTherm program, a number of benchmarking cases were established comparing SatTherm to the commercially available Thermal Desktop program by Cullimore & Ring Technologies. Thermal Desktop is a well established, CAD-based, thermal analysis tool, which is capable of both Finite Difference and Finite Element Analysis. While numerous benchmarking scenarios were examined during the writing of the SatTherm code, only three are described here. They start with a very simple model (a single rectangular plate) and slowly build up in complexity (a full small spacecraft model). In each case, equivalent models were built in both SatTherm and Thermal Desktop, with identical, randomly chosen orbits and external environments. Ideally the two programs would produce matching results.

4.1 Single Rectangular Plate

The first benchmarking model consists of one single rectangular plate. The structure of the models built in SatTherm and PharmaSat are identical. The plate is 1 m x 1 m across, 0.01 m thick, and has the thermal properties of aluminum ($\rho = 2770$ $\frac{W}{m^3}$ and $c_p = 961$ $\frac{J}{kg \cdot K}$) and the optical properties of an alodine coating ($\alpha = 0.23$ and $\epsilon = 0.03$). The rectangular surface is oriented with its normal vector towards a constant Right Ascension of 260° and a Declination of 45°. The randomly chosen orbit has a maximum altitude of 9500km, an eccentricity of 0.3, an inclination of 71°, a RAAN of 98°, and an argument of periapsis of 85°. The position of the sun was chosen

for Jan. 23, 2015, giving an Right Ascension of 305.9° and a Declination of −19.35°. The external radiation absorbed by the surface (including direct solar, albedo, and Earth IR radiation) is plotted for both the SatTherm and Thermal Desktop models in Fig. 4.1. One can see that the two models produce similar results but differ slightly at times. The Albedo curve calculated by SatTherm goes to zero before the Thermal Desktop curve. This may be because of the approximation described in Sec. 2.2.2 with regards to Eqn. 2.33 and 2.34. SatTherm approximates the albedo heat flux as zero when the spacecraft is directly above the terminator ($\theta = 90°$), but in reality, the spacecraft surface will still receive a small amount of albedo radiation.

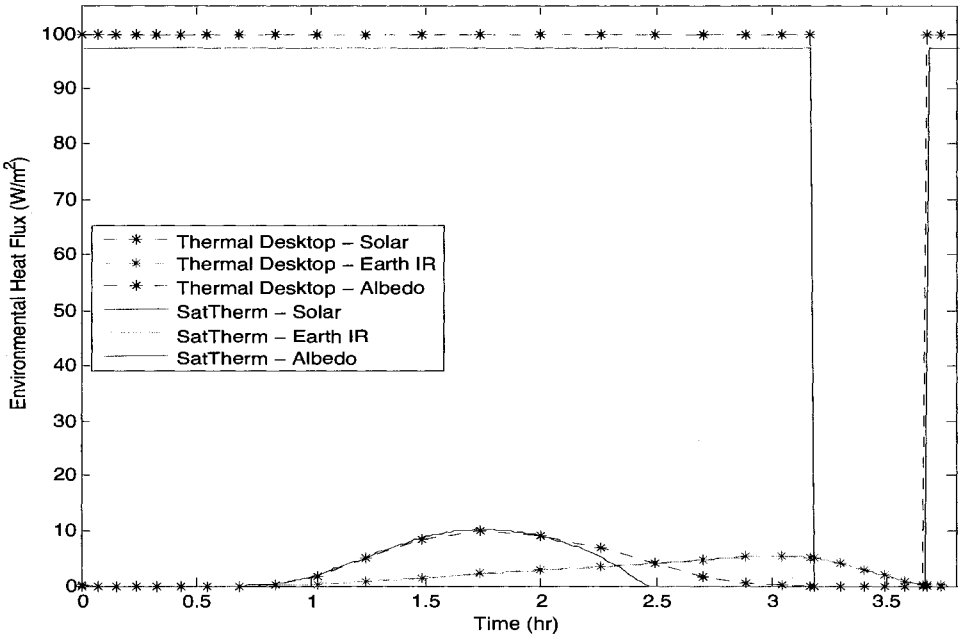


Figure 4.1: The environmental radiative heat flux on a single rectangular plate in Earth-orbit, as found by SatTherm and Thermal Desktop

4.2 Cube

This benchmarking model expands the single plate model to a simple hollow cube with vacuum inside. Again, the structure of the models built in SatTherm and Thermal Desktop is identical. They both consist of six nodes, one for each of the side walls of the cube, which are 1 m x 1 m across, 0.01 m thick, and have the thermal properties of aluminum and the optical properties of an alodine coating. The +Z surface is oriented toward the sun, and no rotation is included. The randomly chosen orbit has a maximum altitude of 1000 km, an eccentricity of 0.04, an inclination of 24°, a RAAN of 10°, and an argument of periapsis of 62°. The position of the sun was chosen to be 305.9° Right Ascension, and −19.35° Declination (corresponding to Jan. 23, 2015).

First, to verify that SatTherm accurately calculates the heat flow due to conduction between nodes, contact conductances were included in the model, set at $1W/K$ between the walls that touch each other. A sample of one of the wall temperatures is shown in Fig. 4.2. These results from SatTherm and Thermal Desktop match each other closely.

Next, to verify the accuracy of the radiation network calculations, radiation between the nodes was included in the cube model. The view factors were set at 0.1998 for sides opposite to each other (using Eqn. A.2), and 0.2 for sides adjacent to each other (using Eqn. A.3). For this case, the conduction between nodes was set to zero, so that the conduction effects would not overpower the radiation effects. A sample of one of the wall temperatures is shown in Fig. 4.3 with only radiation included. Again, the results from SatTherm and Thermal Desktop match closely.

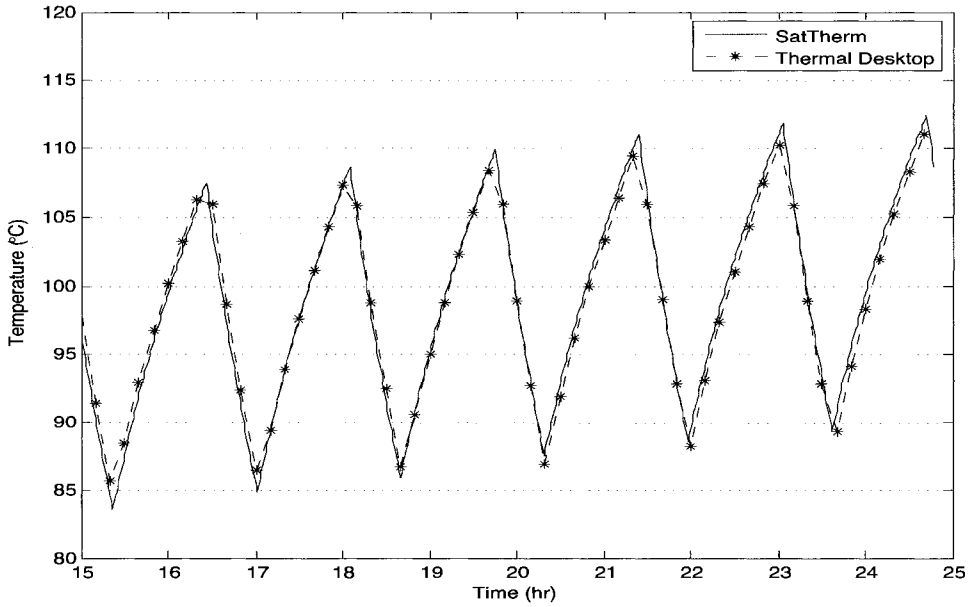


Figure 4.2: The temperature of one side of a cube in Earth-orbit with conduction between nodes, as predicted by SatTherm and Thermal Desktop.

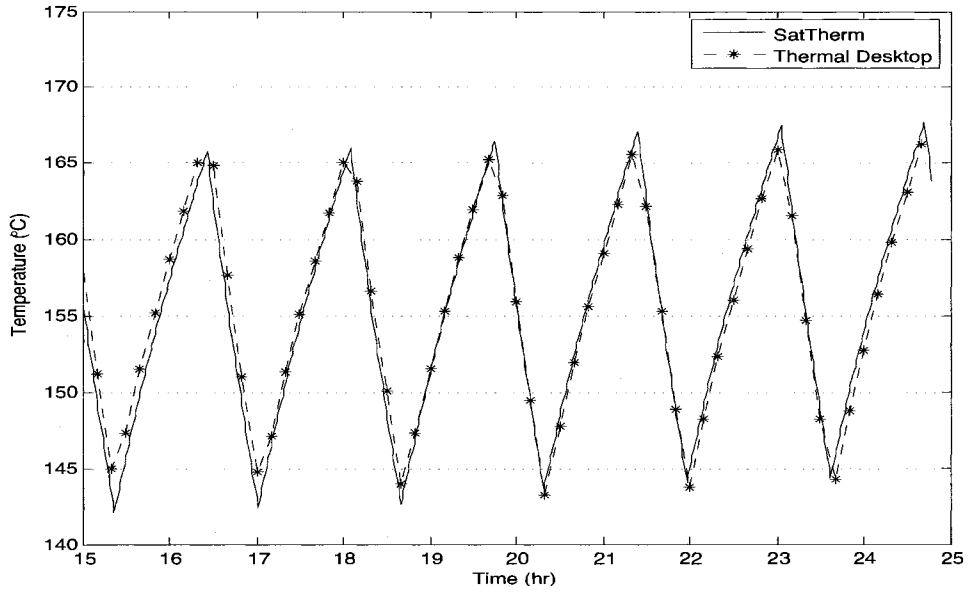


Figure 4.3: The temperature of one side of a cube in Earth-orbit with radiation between nodes, as predicted by SatTherm and Thermal Desktop.

4.3 PharmaSat Spacecraft

PhamaSat is a small satellite, built at the NASA Ames Research Center, and is scheduled to be launched in Jan. 2009. It is comprised of three CubeSats, with the bus occupying one of the cubes and the payload occupying the other two. It carries a biological experiment to study the effects of microgravity on yeast and it's ability to fight an antifungal agent. A picture of PharmaSat is shown in Fig. 4.4. It is approximately 40 cm long and amasses 5 kg. GaAs solar panels are body mounted to four of the side panels of the spacecraft. PharmaSat will fly in a Low Earth Orbit, at an altitude of 460 km and an inclination of 40.5 °C. The spacecraft is oriented by a set of magnets, which align the +Z axis of the spacecraft (pointing towards the right hand side of Fig. 4.4) with the Earth's magnetic field lines. The spacecraft spins about the +Z axis. The temperature requirements for several of the spacecraft bus components are given in Table 4.1 (M. Diaz-Aguado, personal communication, Apr. 4, 2008).

Figure 4.4: Photograph of the PharmaSat spacecraft during preparation for thermal vacuum testing. (M. Diaz-Aguado, personal communication, Oct. 13, 2008)

Table 4.1: Bus component temperature requirements

Component	Operational Range		Survival Range	
	T_{min} °C	T_{max} °C	T_{min} °C	T_{max} °C
Beacon	−15	85	−40	85
PCB Boards	−40	85	−40	85
Batteries	0	45	−10	60

4.3.1 SatTherm PharmaSat Model

A thermal model of the PharmaSat spacecraft was created in the SatTherm program. A single node was defined for each bus component and wall, coming to a total of 20 nodes. The spacecraft configuration option was set as a rectangular cross-section, and for simplicity the spacecraft orientation was modeled with the +Z axis nadir pointing (toward the center of the Earth). The rotation was set to 3600 degrees per orbit, which corresponds to 10 rotations per orbit.

The four long side panels (as seen in Fig. 4.4) each consist of one node, which must combine the aluminum substrate panel and the thin solar cells. To model this combination, the thermal properties of aluminum were chosen, since most of the mass consists of the aluminum substrate. However, because two thirds of the surface area is GaAs solar cells ($\alpha = 0.674$ and $\epsilon = 0.85$) and one third of the area is gold coated aluminum ($\alpha = 0.23$ and $\epsilon = 0.03$), an area-weighted average emissivity and absorptivity were calculated ($\alpha = 0.509$ and $\epsilon = 0.588$) and added to the optical properties database for these nodes.

Each of the four batteries also consist of a single node, with the thermal and optical properties of aluminum. Because the batteries are three-dimensional cylinders and SatTherm is limited to two-dimensional nodal regions, the size must be approximated such that they have roughly the same surface area as the cylinders and the

same volume (and therefore the same mass) as the cylinders. These nodal regions were chosen to be 6.5 cm long (the same length as the cylinders) and 1.85 cm wide (the diameter of the cylinders), making the input area 0.00121 m^2. Then the thickness was choses to be 1.45 cm, so that the volume of the nodal region ($area \times thickness$) is 1.75×10^{-5} m^3 (the same volume as the cylinders). The batteries are surrounded by a two piece battery case, each of which were modeled with their own nodes.

There are also five printed circuit boards (PCBs) in the PharmaSat bus. Each of these was easily modeled as a two-dimmensional nodal region with appropriate surface area and thickness, and with the material properties of Fr4 (a fire resistant material used to make PCBs, $\rho = 1360\ \frac{W}{m^3}$ and $c_p = 600\ \frac{J}{kg \cdot K}$). The spacecraft's micro-hard processor was also modeled as a single node. The material properties of aluminum were used, but because the micro-hard is not actually a solid block of aluminum, a density multiplier of 0.5 was defined so that the node has the correct mass.

Because the focus of this model was limited to the spacecraft bus, the entire payload was modeled as a single node with the properties of aluminum, an appropriate density multiplier to adjust the node mass, and the total internal heat load of all the payload components.

With knowledge of the bus layout and how each component is oriented with respect to the others, the radiation view factor between each node was approximated using Eqns. A.2 and A.3. The contact conductivity was also approximated, knowing the contact areas between each component and using Gluck and Baturkin (2002) as a guide.

After the model was built, it was run for a short period of time, with a first estimate of the time-step of 60 seconds. The program output showed that this time-step was too large, and recommended a maximum time-step of 2.2 seconds. The

time-step was then reset to 2.1 seconds and run for a full analysis which produced stable results.

As mentioned above, there was a total of 20 nodes in the model. It took the author approximately two 8-hour days to build the model in SatTherm and takes approximately 3.5 minutes to run a simulation of 20 orbital periods, or roughly 30 hours, with a time-step of 2.1 seconds.

4.3.2 Thermal Desktop PharmaSat Model

M. Diaz-Aguado, the Thermal Engineer for the PharmaSat mission created a finite difference thermal model of the PharmaSat spacecraft with the Thermal Desktop program (private communication, Apr. 4, 2008). Thermal Desktop has an advantage over SatTherm in that it offers a visualization of the model, as it is being built. An example of this is shown in Fig. 4.5. Thermal nodes appear in the image as rectangles with small white spheres at the center.

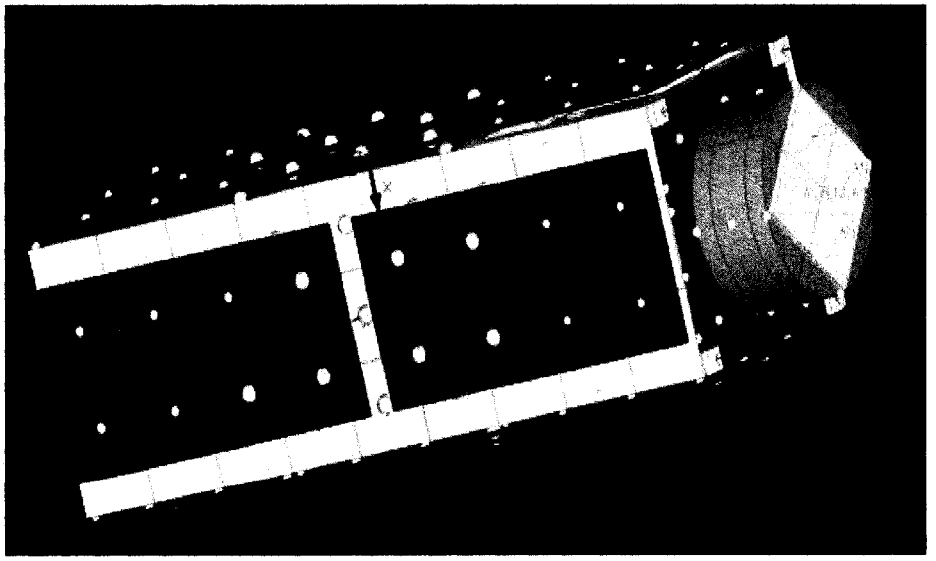


Figure 4.5: An outside view of PharmaSat thermal model in Thermal Desktop

The contents of the Thermal Desktop model are very similar to the model built in SatTherm (described in Sec. 4.3.1). However, in the Thermal Desktop model, each bus component was divided into several nodes, and Thermal Desktop can have three-dimensional nodal regions, unlike SatTherm, which is limited to two-dimensional nodal regains (as discussed in Sec. 3.1). In this model, the side panels (shown in yellow in Fig. 4.5) and the solar cells (shown in blue) are modeled as separate pieces, as opposed to the SatTherm model, which combined them.

With the outside panels removed and the image zoomed in, one can see the details of the spacecraft bus in Fig. 4.6. The cylinders in the center of Fig. 4.6 are the batteries and each was split into 12 three-dimensional nodal regions. Many of the flat plates in Fig. 4.6 are PCBs for the command and data handling system, electrical power system, and communications system.

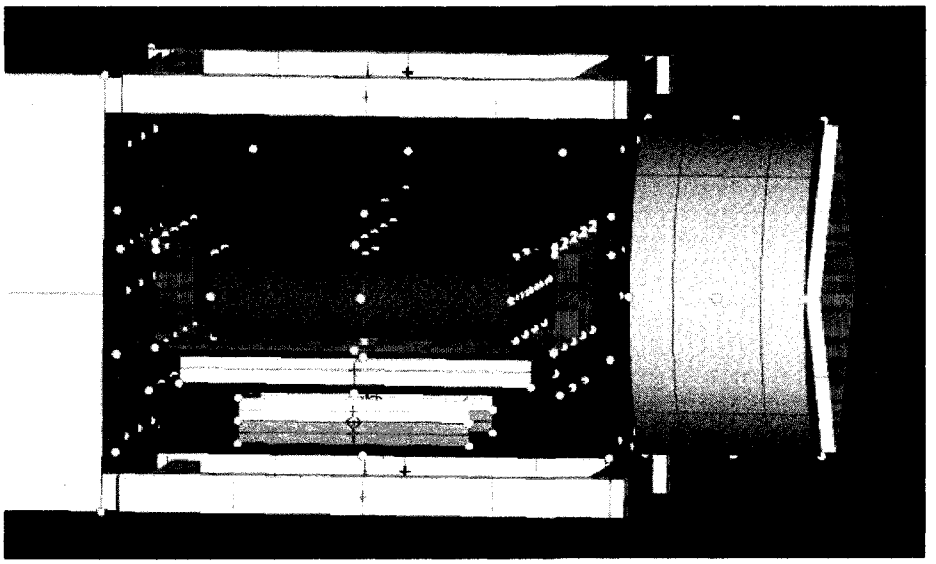


Figure 4.6: An inside view of the PharmaSat bus thermal model in Thermal Desktop.

The material properties and internal heat loads were defined similarly to those in the SatTherm model. Contact conductances were defined for each pair of surfaces or edges that touch. Thermal Desktop calculates the radiation view factors between nodes using a random ray tracing technique.

Altogether the Thermal Desktop model has a total of 461 nodes. It took approximately two full work weeks to build, and takes about 2.5 minutes to run a simulation of 20 orbital periods, or roughly 30 hours.

4.3.3 Simulation Cases

Before a spacecraft is launched, the environment that it will encounter while in orbit cannot be known exactly. As described in Sec. 2.2.2, the external and internal heat loads on a spacecraft vary over some range, depending on time of launch, exact orbit details, and natural variations in the solar output and Earth atmospheric conditions. The internal heat loads of the electronics my vary as well. To assure that a spacecraft will survive in orbit, thermal analysis is done for a hot case (in which all heat loads are assumed to be on the high end of their range) and a cold case (in which all heat loads are assumed to be on the low end of their range). The spacecraft should be able to survive both of these extreme cases, meaning that all spacecraft components should stay within their temperature requirement ranges.

To demonstrate the accuracy of the SatTherm software the SatTherm and Thermal Desktop PharmaSat models were both run though identical hot and and cold case simulations. Table 4.2 lists some of the conditions used for the different cases.

Table 4.2: Conditions chosen for the hot and cold case analyses.

	Hot Case	**Cold Case**
Solar	1414 W/m^2	1322 W/m^2
Earth IR	257 W/m^2	218 W/m^2
Albedo	0.26	0.19
Orbit Inclination	40.5°	40.5°
RAAN	60°	0°
Argument of Perigee	270°	270°
RA Sun	247°	247°
Dec. Sun	−21.7°	−21.7°
Altitude	460 km	460 km
Eccentricity	0	0

4.3.4 Results

For both the hot and cold case, a time-dependent thermal analysis of the SatTherm PharmaSat model was run (with initial conditions starting at room temperature) for a length of 20 orbit periods. This is a sufficient amount of time for the spacecraft temperature to reach steady state conditions. The temperature of several spacecraft components were examined and compared to the results given by the Thermal Desktop model, which was shared by M. Diaz-Aguado (private communication, Apr. 4, 2008). For additional comparison, the results of both models were also compared to flight data from the GeneSat Spacecraft, which was recorded after it's December 2006 launch. GeneSat was another small spacecraft, which carried a biological experiment to study the effects of the space radiation environment on E. coli bacteria. The buses of GeneSat and PharmaSat are almost identical and have similar orbits, so it is reasonable to expect that the bus components will have similar temperatures during flight. The GeneSat flight data was also provided by M. Diaz-Aguado (private communication, May 12, 2008).

The predicted battery temperature for the hot and cold case analyses are dis-

played in Fig. 4.7. The solid lines are the results found by SatTherm, showing that the battery temperature levels off at 34 $^{\circ}C$ for the hot case and 8 $^{\circ}C$ for the cold case. The short-term wobble in the temperature is due to the orbital variation of going in and out of eclipse. Fig. 4.7 also shows that the battery temperature predicted by the Thermal Desktop analysis (dashed lines) is 33 $^{\circ}C$ and 10 $^{\circ}C$ for the hot and cold case respectively. The battery temperature found by the two models agree within appriximately 2 $^{\circ}C$. Fig. 4.8 shows GeneSat's battery temperature flight data. Though the time scale of Fig. 4.8 is different from that of Fig. 4.7, it can be seen that the GeneSat flight data falls within the hot and cold range predicted by the PharmaSat models, as expected.

The temperature of one of the side panels of the bus is shown for the hot and cold cases in Figs. 4.9 and 4.10, respectively. The solid lines are the SatTherm results and the dashed lines are the Thermal Desktop results. The side panel temperature varies more within each orbit than the battery temperature because it is directly exposed to the varying external environment as the spacecraft goes in and out of eclipse. The temperatures predicted by SatTherm match those from Thermal Desktop within a few degrees celsius.

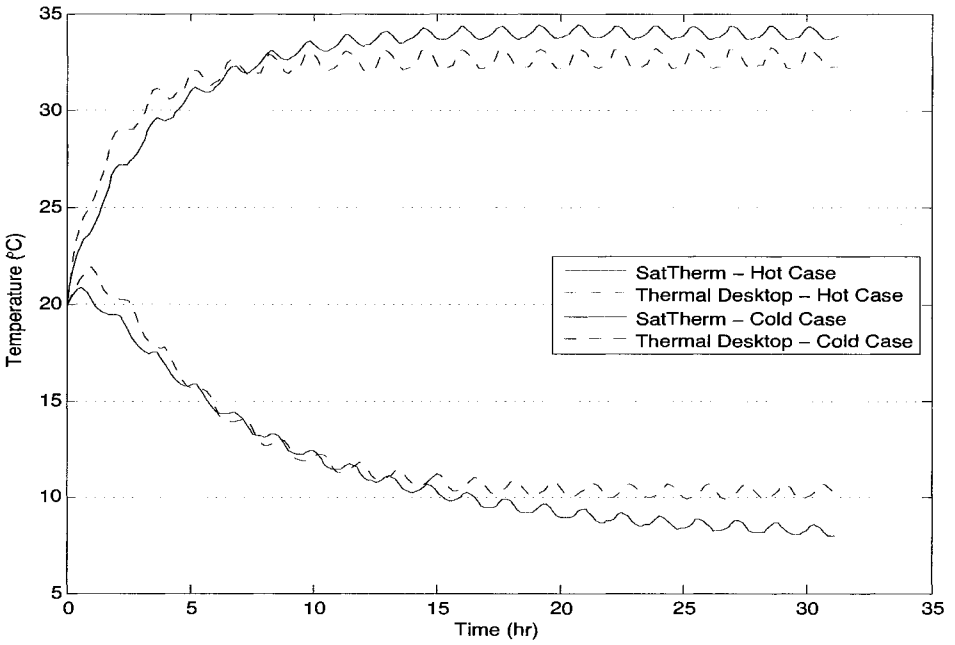


Figure 4.7: PharmaSat battery temperature over the first 20 orbital periods for the hot and cold cases, as found by the SatTherm and Thermal Desktop models

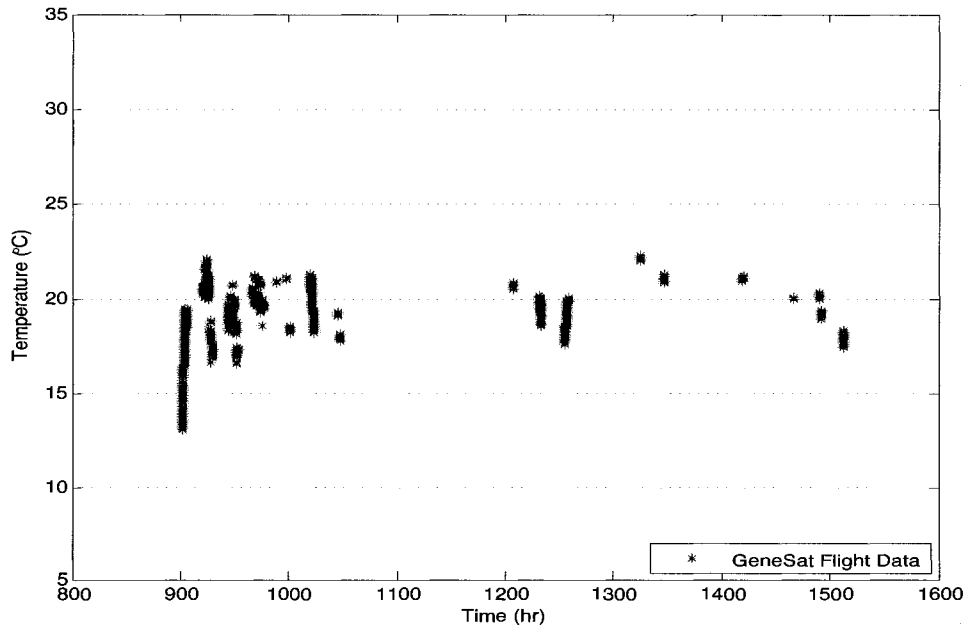


Figure 4.8: GeneSat battery temperature recorded during flight

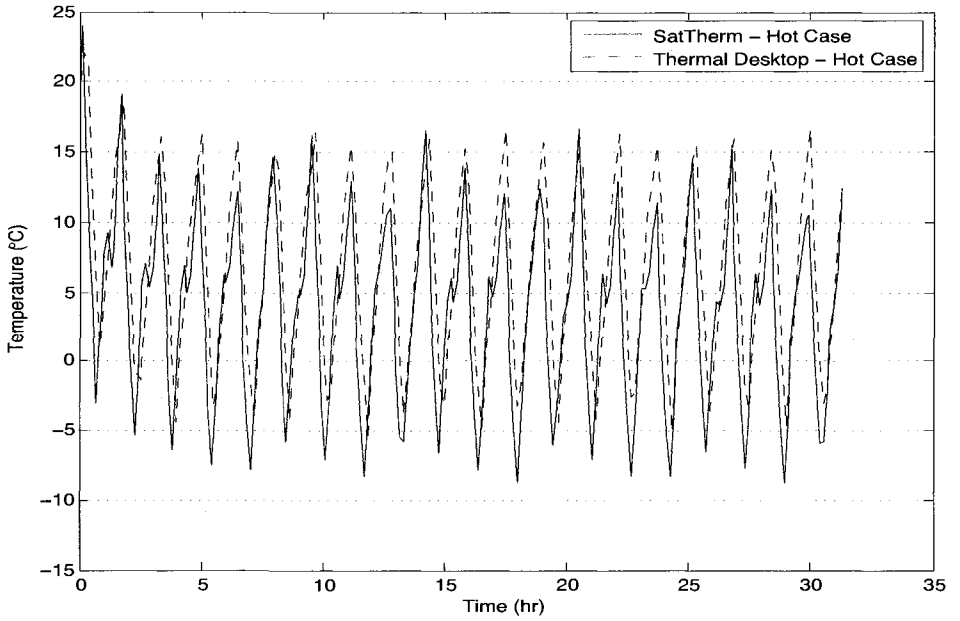


Figure 4.9: PharmaSat side panel temperature over the first 20 orbital periods for the hot case, as found by the SatTherm and Thermal Desktop models

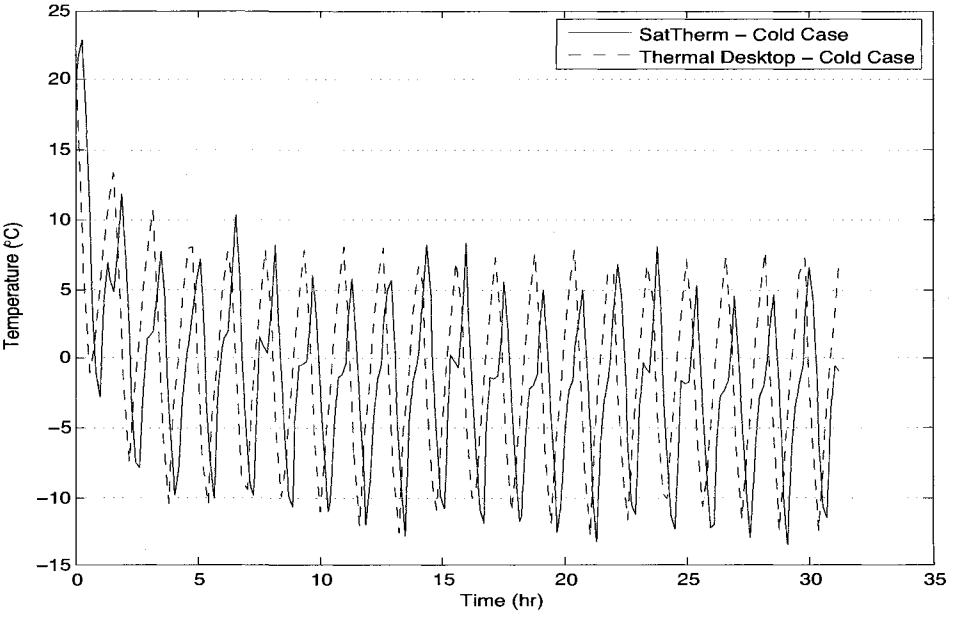


Figure 4.10: PharmaSat side panel temperature over the first 20 orbital periods for the cold case, as found by the SatTherm and Thermal Desktop models

CHAPTER 5

CONCLUSION

SatTherm is a simple, easy to use thermal analysis software tool designed for the early stages of small spacecraft mission design. Several benchmarking cases have been presented, comparing SatTherm to the commercially available and widely used Thermal Desktop. The benchmarking cases involving a single rectangular plate and a simple cube (examined in Sec. 4.1 & 4.2) have shown that SatTherm accurately calculates the environmental radiation that a spacecraft will encounter while in orbit, and accurately solves the differential heat equations for a given model. The discussion of the PharmaSat model in Sec 4.3 also demonstrates that SatTherm is a useful tool for creating a simple thermal model of a small spacecraft, and that such a simple model can produce accurate predictions of spacecraft component average temperatures. The temperatures predicted by the SatTherm model agree with those predicted by the Thermal Desktop model within 4 $^{\circ}C$ or less. Even for the lowest temperature examined in these cases, that is less than 2 %, if considered in the absolute Kelvin scale.

There is no indication that the results found by SatTherm tend to be on the cooler or warmer side of those found by Thermal Desktop. The uncertainty should be thought of as ± 4 $^{\circ}C$. When making a Thermal Model of a spacecraft there are numerous sources of error, including error in the mathematical calculations as well as error due to the simplifications made to approximate reality. For example: three-dimensional objects were modeled as two-dimensional flat surfaces, and the radiation view factors were calculated using the equations for identical, parallel and directly

opposite rectangles (Eqn. A.2) and for two rectangles with a common edge (Eqn. A.3), even if the actual configuration was slightly different.

The discussion of the PharmaSat model in Sec. 4.3 also demonstrates the time-saving value of a simple thermal model built in SatTherm compared to a complex model built in Thermal Desktop. Though the SatTherm PhamaSat model takes slightly longer to complete one run than the Thermal Desktop model (3.5 minutes compared to 2.5 minutes), the real time saved is in how long it takes to learn the program and build a model (on the order of days compared to weeks).

In its current form, SatTherm is a useful thermal analysis tool for small spacecraft. It does have its limitations, though. In the future, it may be expanded to include:

- more complex spacecraft geometry, such as solar panels on booms extending away from the spacecraft body, or payload components attached to the outside of the body walls.
- more complex orientation control, such as aligning the spacecraft with Earth's magnetic field lines, instead of only having the choice of Sun facing, Nadir facing, or setting a constant right ascension and declination.
- the ability to model interplanetary trajectories.
- the ability to define different optical properties on different sides of a single surface.
- a more efficient check and reassignment of the time-step if the initial time-step is too large, which can cause an unstable solution.

REFERENCES

Bate, R. R., Mueller, D. D., & White, J. E. (1971). *Fundamentals of astrodynamics.* New York: Dover Publications, Inc.

Cengel, Y. A. (1998). *Heat transfer: A practical approach.* San Francisco: McGraw-Hill.

Chapman, A. J. (1987). *Fundamentals of heat transfer.* New York: Macmillan Publishing Co.

Clawson, J. F., Tsuyuki, G. T., Anderson, B. J., Justus, C. G., Batts, W., Ferguson, D., et al. (2002). Spacecraft thermal environment. In D. G. Gilmore (Ed.), *Spacecraft thermal control handbook. Vol. I: Fundamental technologies.* El Segundo, CA: The Aerospace Press.

Feingold, A. (1966).
Proceedings of The Royal Society of London. Series A, Mathematical and Physical Sciences, *292*(1428), 51-60.

Gilmore, D. G., & Collins, R. L. (2002). Thermal design and analysis. In D. G. Gilmore (Ed.), *Spacecraft thermal control handbook. Vol. I: Fundamental technologies.* El Segundo, CA: The Aerospace Press.

Gilmore, D. G., Hardt, B. E., Prager, R. C., Grob, E. W., & Ousley, W. (2006). Thermal. In W. J. Larson & J. R. Wertz (Eds.), *Space mission analysis and design* (3rd ed.). El Segundo, CA: Microcosm Press.

Gluck, D. F., & Baturkin, V. (2002). Mountings and interfaces. In D. G. Gilmore (Ed.), *Spacecraft thermal control handbook. Vol. I: Fundamental technologies.* El Segundo, CA: The Aerospace Press.

Hamilton, D. C., & Morgan, W. R. (1952). *Radiant-interchange configuration factors* (NACA Technical Note No. 2836).

Siegel, R., & Howell, J. R. (1996). *Thermal radiation heat transfer* (2nd ed.). New York: Hemisphere Publishing Co.

Spacecraft thermal balance (Aerospace Information Report No. 1168/12). (2004). SAE Aeropsace.

Thornton, E. A. (1996). *Thermal structures for aerospace applications.* Reston, VA: AIAA Educational Series.

Vallado, D. A. (2001). *Fundamentals of astrodynamics and applications* (2nd ed.). El Segundo, CA: Microcosm Press.

APPENDIX A

RADIATION VIEW FACTORS

The view factor (also know as the shape or configuration factor), as described in Sec. 2.1.3, is the fraction of radiation emitted by one surface, i, that is intercepted by another surface, j. The exact solution for this fraction is

$$F_{i-j} = \frac{1}{A_i} \int_{A_i} \int_{Aj} \frac{cos\theta_i cos\theta_j}{\pi S^2} dA_i dA_j \tag{A.1}$$

where A_i and A_j are the respective areas, and θ_i and θ_j are the angles between the normal vectors, n_i and n_j, and the shortest line between the two areas, S (Siegel & Howell, 1996). Fig. A.1 displays this geometry.

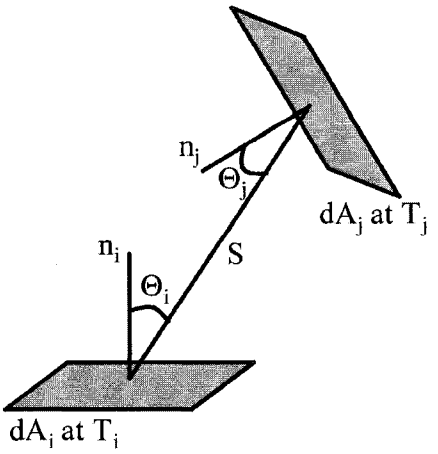


Figure A.1: Two differential areas dA_i and dA_j with normal vectors, n_i and n_j.

While this double integral becomes quite complicated for complex geometry, shape factors for many simple geometric configurations have been approximated and catalogued. Below are several relevant configurations that arise in the thermal analysis of small spacecraft.

Two identical, parallel, and directly opposite rectangular plates are pictured in Fig. A.2. Siegel and Howell (1996) show that the view factor between these two plates is

$$F_{1-2} = \frac{2}{\pi XY}\left(\ln\left(\frac{(1+X^2)(1+Y^2)}{1+X^2+Y^2}\right)^{1/2} + X\sqrt{1+Y^2}\tan^{-1}\frac{X}{\sqrt{1+Y^2}} + Y\sqrt{1+X^2}\tan^{-1}\frac{Y}{\sqrt{1+X^2}} - X\tan^{-1}X - Y\tan^{-1}Y\right) \quad \text{(A.2)}$$

where $X = a/d$, the ratio of the length of one side of the rectangles to the distance between the two, and $Y = b/d$, the ratio of the length of the other side of the rectangles to the distance between the two.

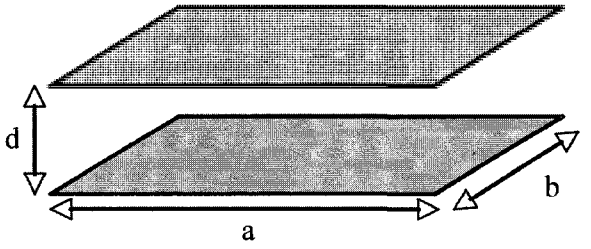


Figure A.2: Two identical, parallel, and directly opposite rectangular plates. The view factor between such plates is given by Eqn. A.2.

Two rectangular plates with with a common edge are shown in A.3. The two plates are at an arbitrary angle Φ with respect to each other. If we define the ratios of edge lengths as $L = (a/c)$ and $N = (a/b)$ then the view factor from Plate 1 to Plate 2 is

$$
\begin{aligned}
&F_{1-2} = \\
&\frac{1}{\pi L}\left(-\frac{1}{4}\sin 2\Phi\left[NL\sin\Phi + \left(\frac{1}{2}\pi - \Phi\right)(N^2+L^2)\right.\right. \\
&\left.+L^2\tan^{-1}\left(\frac{N - L\cos\Phi}{L\sin\Phi}\right) + N^2\tan^{-1}\left(\frac{L - N\cos\Phi}{N\sin\Phi}\right)\right] \\
&+\frac{1}{4}\sin^2\Phi\ln\left\{\left[\frac{(1+N^2)(1+L^2)}{1+N^2+L^2-2NL\cos\Phi}\right]^{\csc^2\Phi+\cot^2\Phi}\right. \\
&\left.\left[\frac{L^2(1+N^2+L^2-2NL\cos\Phi}{(1+L^2)(N^2+L^2-2NL\cos\Phi)}\right]^{L^2}\right\} \\
&+\frac{1}{4}N^2\sin^2\Phi\ln\left[\left(\frac{N^2}{N^2+L^2-2NL\cos\Phi}\right)\left(\frac{1+N^2}{1+N^2+L^2-2NL\cos\Phi}\right)^{\cos 2\Phi}\right] \\
&+L\tan^{-1}\frac{1}{L} + N\tan^{-1}\frac{1}{N} - \sqrt{N^2+L^2-2NL\cos\Phi}\cot^{-1}\sqrt{N^2+L^2-2NL\cos\Phi} \\
&+\frac{1}{2}N\sin\Phi\sin 2\Phi\sqrt{1+N^2\sin^2\Phi}\left[\tan^{-1}\left(\frac{N\cos\Phi}{\sqrt{1+N^2\sin^2\Phi}}\right)\right. \\
&\left.+\tan^{-1}\left(\frac{L - N\cos\Phi}{\sqrt{1+N^2\sin^2\Phi}}\right)\right] + \cos\Phi \\
&\left.\int_0^L \sqrt{1+z^2 sin^2\Phi}\left[\tan^{-1}\left(\frac{N - z\cos\Phi}{\sqrt{1+z^2\sin^2\Phi}}\right) + \tan^{-1}\left(\frac{z\cos\Phi}{\sqrt{1+z^2\sin^2\Phi}}\right)\right] dz\right)
\end{aligned}
\tag{A.3}
$$

Eqn. A.3 is given by Hamilton and Morgan (1952), and was confirmed with the tabulated values in Feingold (1966). Please be warned that, though Feingold (1966) gives the equation for this configuration, it has a typo in it, and does not result in the tabulated values that he also gives.

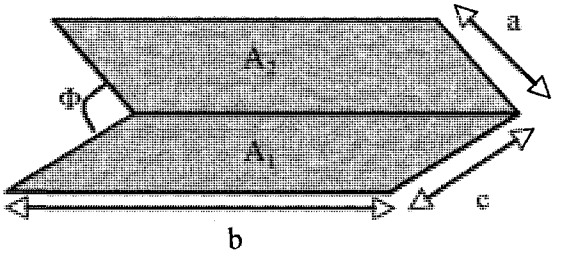


Figure A.3: Two rectangular plates with areas, A, and a common edge of length b, at an angle Φ to each other.

Now consider an elemental area, dA1, who's normal vector makes a generic angle, γ, with respect to a straight line between the infinitesimal area and the center of a sphere of radius, R_e (see Fig. A.4). This configuration can be used to approximate the side wall of a spacecraft in orbit around the Earth, or any other large spherical body. Compared to the large size of the Earth, a spacecraft wall is essentially an infinitesimal area. If R_{sc} is the distance from the center of the Earth to the spacecraft (as shown in Fig. A.4) then the altitude of the spacecraft above the surface is $h = R_{sc} - R_e$. Then we can define $H = R_{sc}/R_e$, and $\Phi_m = \sin(1/H)$. Here, Φ_m is the angle from the line R_{sc} to the outer limb of the Earth's disk as seen from the position of the spacecraft. Thornton (1996) gives the following equations for the view factor from the spacecraft wall to the earth. When $\gamma \leq \pi/2 - \Phi_m$ (the spacecraft surface is oriented such that it can see the entire Earth-disk) the view factor is

$$F_e = \frac{\cos\gamma}{H} \tag{A.4}$$

When $\gamma > \pi/2 - \Phi_m$ and $\gamma \leq \pi/2 + \Phi_m$ (the spacecraft surface is oriented such that it has only a partial view of the Earth-disk) it is

$$\begin{aligned} F_e \;=\; & \frac{1}{2} - \frac{1}{\pi}\sin^{-1}\left(\frac{(H^2-1)^{1/2}}{H\sin\gamma}\right) + \frac{1}{\pi H^2}\cos\gamma\cos^{-1}\left(-\left(H^2-1\right)^{1/2}\cot\gamma\right) \\ & -\frac{1}{\pi H^2}\left(H^2-1\right)^{1/2}\left(1-H^2\cos^2\gamma\right)^{1/2} \end{aligned} \tag{A.5}$$

And, of course, when $\gamma > \pi/2 + \Phi_m$ (the spacecraft surface is pointed away from the earth) it is

$$F_e = 0 \tag{A.6}$$

Fig. A.5 shows the plotted values of Eqns. A.4, A.5, and A.6, giving the view factor from the wall of a spacecraft to the earth for several values of γ. This plot has been verified with Chapman (1987) and *Spacecraft Thermal Balance* (2004)

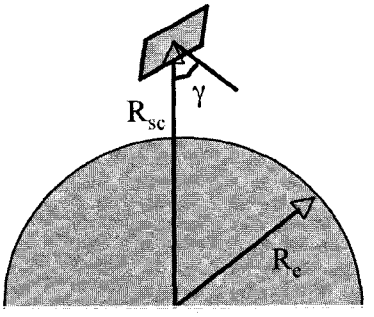


Figure A.4: An infinitesimal area dA_1 at an angle γ with respect to a large sphere. This represents a good approximation for calculating the view factor for a spacecraft wall in orbit above the Earth.

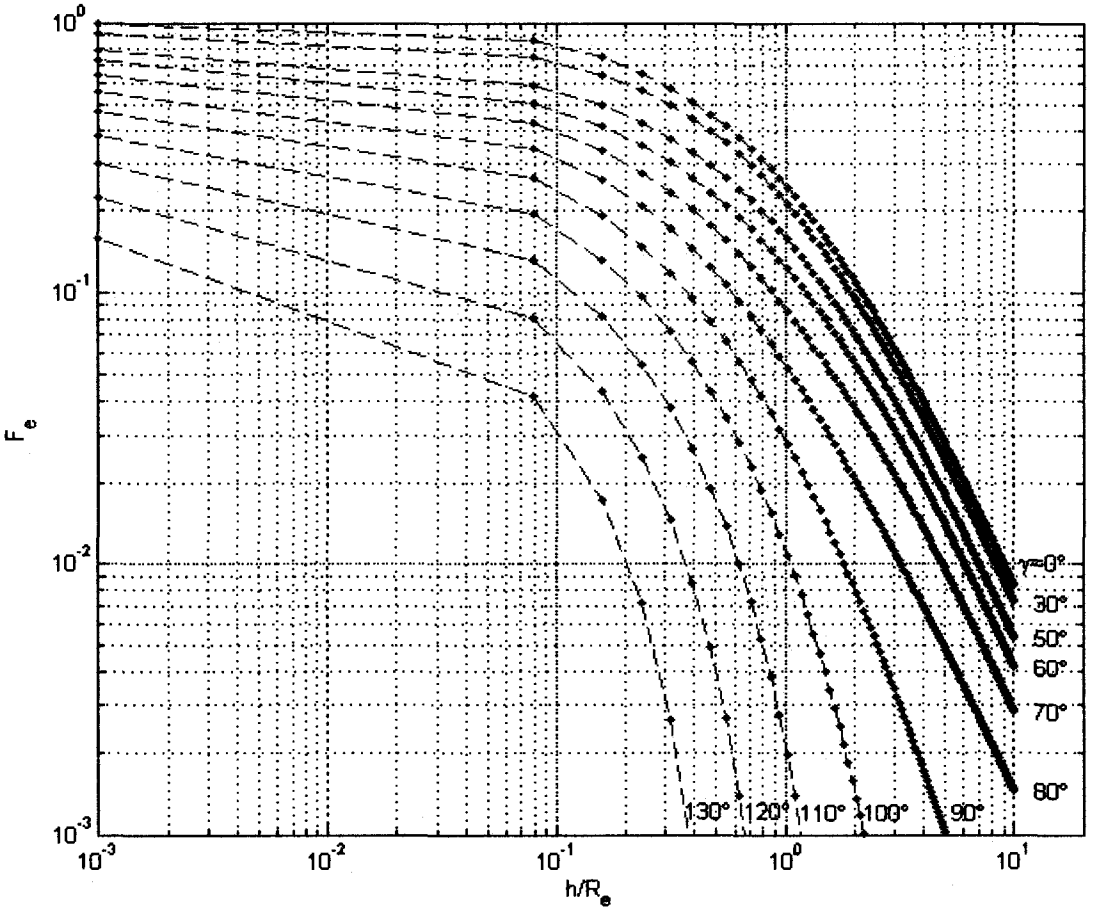


Figure A.5: The view factor from a spacecraft wall to the Earth, as a function of the ratio of the spacecraft altitude and radius of the Earth (h/R_e), plotted for several values of γ (as drawn in Fig. A.4).

APPENDIX B

PROGRAM CODE

B.1 Calculate Spacecraft Position and Velocity

```
function [r_geocen,v_geocen] = geocen_rv(inc, raan, arg_peri,...
 max_alt, ecc, t)

%this function calculates the geocentric position of a
%spacecraft in Earth orbit.  Output is a 3 element matrix
%[ri,rj,rk] in km.

%Inputs:
%inc = inclination (deg),
%raan = right ascension of ascending node (deg)
%arg_peri = argument of periapsis (deg)
%max_alt = maximum altitude (km),
%ecc = eccentricity
%t = time (s) (assuming t=0s when ecc. anom., E=0)

%constants
mu=398601;                          %Earth's grav. parameter(km)
Re = 6378.14;                       %radius of Earth (km)

a = (Re + max_alt)./(1+ecc);          %semi-major axis (km)
```

```
incr = inc.*pi./180;                          %inclination (rad)
raanr = raan.*pi./180;                        %raan  (rad)
arg_perir = arg_peri.*pi./180;       %arg of periapsis (rad)

P=2.*pi.*sqrt(a^3/mu);           %Oribt Period (s)
M = 2.*pi.*t./P;                          %Mean Anomoly (radians)
E_guess = (M.*(1-sin(M+ecc))+(M+ecc).*sin(M))./...
(1+sin(M)-sin(M+ecc));  %Initial guess for E (Prussing, eq2.16))

functE = @(E)sqrt(a.^3./mu).*(E-ecc.*sin(E))-t;  %(BMW eqn4.2-1)
E=fzero(functE,E_guess);           %Ecc Anomoly (rads)

r_pfmag = a.*(1-ecc.*cos(E));
%distance from focus (peri-focal coords) (BMW eqn 4.2-14)
r_pf = a.*[(cos(E)-ecc),(sqrt(1-ecc.^2).*sin(E)),0];
v_pf = sqrt(mu.*a)./r_pfmag.*[-sin(E), sqrt(1-ecc.^2).*cos(E),0];

%rotation matrix (transform from peri-focal to geocentric coords)
rotmat(1,1) = cos(raanr).*cos(arg_perir)-sin(raanr)...
.*sin(arg_perir).*cos(incr);
rotmat(1,2) = -cos(raanr).*sin(arg_perir)-sin(raanr)...
.*cos(arg_perir).*cos(incr);
rotmat(1,3) = sin(raanr).*sin(incr);
rotmat(2,1) = sin(raanr).*cos(arg_perir)+cos(raanr)...
.*sin(arg_perir).*cos(incr);
rotmat(2,2) = -sin(raanr).*sin(arg_perir)+cos(raanr)...
```

```
.*cos(arg_perir).*cos(incr);
rotmat(2,3) = -cos(raanr).*sin(incr);
rotmat(3,1) = sin(arg_perir).*sin(incr);
rotmat(3,2) = cos(arg_perir).*sin(incr);
rotmat(3,3) = cos(incr);

r_geocen = transpose(rotmat*transpose(r_pf));
       %sc position (geocentric coords)(BMW p83)

v_geocen = transpose(rotmat*transpose(v_pf));
       %sc velocity (geocentric coords)

end
```

B.2 Calculate Spacecraft-Earth Radiation View Factor

```
function Fe = view_factor_scearth(h,gamma)

%This function calculates the view factor (also known as configuration
%factor, geometry factor) Fd1-2, from an infinitesimal surface (d1)
%to the Earth (2).  Output, Fe, is unitless.

%Inputs:
%h = altitude of the surface above Earth's surface (km)
%gamma = angle between normal vector and nadir vector (degrees)
```

```
gammar = gamma.*pi./180;          %gamma in radians
Re=6378.14;                       %Radius of the Earth (km)
rsc=Re+h;          %distance of spacecraft from center of Earth (km)

H = rsc./Re;
phi_m = asin(1/H);
b = sqrt(H.^2-1);

%if full Earth is visible to the plate
if gammar <= pi./2-phi_m;
    Fe = cos(gammar)./H.^2;
%if part of the Earth is visisible to the plate
elseif gammar > pi./2-phi_m && gammar <= pi./2+phi_m;
    t1 = 1./2.*asin(b./(H.*sin(gammar)));
    t2 = 1./(2.*H.^2).*(cos(gammar).*acos(-b.*cot(gammar))...
    -b.*sqrt(1-H.^2.*(cos(gammar))^2));
    Fe = 2./pi.*(pi./4-t1+t2);
%if none of the Earth is visible to the plate
else;
    Fe=0;
end

end
```

B.3 Calculate External Heat Flux

```
function qtot = ext_heat_flux_loop(r_sc,r_sun,n,Gs,AF,Te,...
abso_sol,emis_ir)

%This function calculates the external (direct solar, qs, albedo
%radiation, qa and Earth IR) heat flux (W/m^2) absorbed by each
%input surface. The output is a 1 x 3 x nsides vector, where nsides
%is defined below. Each page of the output vector corresponds to
%one of the sides of the spacecraft.  The 3 columns correspond to
%[qs,qa,qe] respectively.  The total heat flux would be sum(output),
%and the total heat flow absorbed by the surface (W) would be sum
%(output)*area of the surface.

%Inputs:
%r_sc = surf. position geocentric-equitorial coords [rx,ry,rz]) (Km)
%r_sun = position of the sun in geocen coords [rx,ry,rz] (km)
%n = normal vector of surface in geocen coords [nx,ny,z]
%Gs = solar constant (W/m^2)
%AF = Albedo Factor
%Te = Effective BB temperature of Earth (k)
%abso_sol = solar absorbtivity of surface (unitless)
%emis_ir = Infrared emissivity of surface (unitless)

%Constants
sigma = 5.6704e-8;      %Steffan Boltzmann const (W/(m^2 K))
```

```
Re=6378.14;                    %radius of Earth (km)

sizen = size(n);  %dimensions of the surfaces' normals matrix
nsides = 1; %the # of pages of n = # of surfaces

qtot = zeros(1,3,nsides);   %pre-alocate, to be filled in loop

for k=1:nsides

    h = norm(r_sc)-Re;              %SC altitude

    gammar=acos(-sum(n(:,:,k).*r_sc)./(norm(n(:,:,k)).*norm(r_sc)));
          %angle between n and r_sc in radians

    gamma = gammar.*180./pi;  %angle between n and r_sc in deg

    Fsce = view_factor_scearth(h,gamma);
          %view factor between Earth and surface

    S = r_sun-r_sc;   %line of sight vector between sc and sun (km)

    psir = acos(sum(n(:,:,k).*S)./(norm(n(:,:,k)).*norm(S)));
         %angle between line of sight vector and surface normal
         %vector (radians)

    thetar = acos(sum(r_sun.*r_sc)./(norm(r_sun).*norm(r_sc)));
```

```
 %solar reflection angle off earth (radians)

%Direct Solar radiation absorbed by sc per sqr meter
qs = Gs.*abso_sol.*cos(psir);

%Reflected Solar (albedo) radiation absorbed by sc W/m^2
qa = Gs.*AF.*Fsce.*abso_sol.*cos(thetar);

%Direct Earth IR radiation absorbed by sc W/m^2
qe = sigma.*Te.^4.*emis_ir*Fsce;    %ir abs.=ir emiss.

if insun(r_sc,r_sun)==0;            %If in ecclipse:
    qs = 0;                 %overwrite qs to zero
    qa = 0;                 %overwrite qa to zero
end

if psir >= pi/2;            %If surface pointing away from sun:
    qs = 0;                 %overwrite qs to zero
end

if qa < 0               %albedo goes to zero for theta>pi/2
    qa = 0;
end

%output environmental radiation abosrbed by surface (W/m^2)
qtot(:,:,k)=[qs,qa,qe];
```

```
end

end
```

B.4 Check for Eclipse

```
function insun = insun(r_sc,r_sun)

%This function determines whether a spacecraft is in sunlight
%(returning 1)or in eclipse (returning 0) given the inputs:

%r_sc =  spacecraft position in geocentric coords(km)
%r_sun =  sun position in geocentric coords (km)

Re = 6378;                                  %radius of Earth (km)

theta1 = acos(Re./norm(r_sc));    %angle (see note book p29a) (rads)
theta2 = acos(Re./norm(r_sun));   %angle (see note book p29a) (rads)

psi = acos(sum(r_sc.*r_sun)./(norm(r_sc).*norm(r_sun)));
      %angle between sc position vector and sun position vector (rad)

%if psi is <= theta1+theta2, its in sun, otherwise its in eclipse
if psi >= theta1+theta2;
    insun=0;
```

```
else
    insun=1;
end
end
```

B.5 Calculate Transient Temperatures

```
function [t,temp]=trans_temp(t_final,delta_t,shape,inc,raan,arg_peri,..
max_alt,ecc,orientation,setraddec,rotation,rotation_axis,Re,mu,Te,...
AF,Gs,RAsun,Decsun,ntot,A,th,rho,cp,abso_sol,emis_ir,temp_init,...
inthl,conductance,rad_vf,nodename)


V=A.*th;                          %node volumes
C=rho.*V.*cp;                     %node thermal capacities

sigma = 5.6704e-8;                %Steffan Boltzmann const (W/(m^2 K)

r_sun=RADec2geocen(RAsun,Decsun,149598000);   %geocen pos of sun (km)

a=(Re+max_alt)./(1+ecc);       %orbit semi-major axis
P=2.*pi.*sqrt(a^3/mu);            %orbit period

if P./delta_t <= 2.*rotation/360
```

```
    disp('Warning: Aliasing may occur. The sampling rate is less than
    the Nyquist Rate for the spacecraft rotation. The time step-size
    should be decreased, or the rotation should be decreased.')
end

%Radiation Network
Res_self_inv = (A.*emis_ir)./(1-emis_ir);
        %1/Ri=((1-eps_i)/(A_i*eps_i))^-1 Cengel Eqn. 9-44
Res_other_inv = (repmat(transpose(A),1,ntot).*rad_vf);
        %1/Rij=(1/Ai*Fij)^-1 Cengel Eqn. 9-49

Res_mat = Res_other_inv;   %Begin building Resistance Network matrix
for j=1:ntot      %continue building Resistance Network Matrix
    Res_mat(j,j)=-Res_self_inv(j)-sum(Res_other_inv(j,:));
end

%Figure out how many sides there are (outside panels)
switch shape
    case 'Rectangle'
        nsides = 6
    case 'Hexagon'
        nsides = 8
    case 'Octogon'
        nsides = 10
    otherwise
        disp('Error: Invalide Shape Option.')
```

```
end

temp_old=reshape(temp_init,1,1,ntot);    %set initial temp
t_old=0;
delta_t_old=delta_t;
k_old=0;

N = t_final./delta_t                              %number of timesteps

%initialize arrays, to be filled in the loop
t = zeros(floor(N+1),1,1);
r_sc = zeros(floor(N+1),3,1);
v_sc = zeros(floor(N+1),3,1);
S = zeros(floor(N+1),3,1);
zloc = zeros(floor(N+1),3,1);
xloc = zeros(floor(N+1),3,1);
yloc = zeros(floor(N+1),3,1);
n = zeros(floor(N+1),3,ntot);
n_inloc = zeros(floor(N+1),3,ntot);
n_inloc_rot = zeros(floor(N+1),3,ntot);
qext_arr = zeros(floor(N+1),3,ntot);
Qext = zeros(floor(N+1),1,ntot);
Qspace = zeros(floor(N+1),1,ntot);
Qcond = zeros(floor(N+1),1,ntot);
Qrad = zeros(floor(N+1),1,ntot);
Qint = zeros(floor(N+1),1,ntot);
```

```
temp = zeros(floor(N+1),1,ntot);
delta_t_lim = zeros(floor(N+1),1,ntot);
delta_t_lim_min = zeros(floor(N+1),1,1);
tcheck_sum = zeros(floor(N+1),1,1);
delta_t_new = zeros(floor(N+1),1,ntot);

while t_old < t_final
    k=k_old+1;

    t(k,:) = t_old+delta_t_old;              %current time (s)
    [r_sc(k,:),v_sc(k,:)] = geocen_rv(inc,raan,arg_peri,max_alt,...
    ecc,t(k,:));    %current position & velocity vects (km) and (km/s)
    perp_2_orbit_plane = cross(r_sc(k,:),v_sc(k,:));
          %a vector that is perpendicular to the plane of the orbit
    S(k,:) = r_sun-r_sc(k,:);
          %line of sight vector between spacecraft and sun (km)

    %Define the Spacecraft local axes, xloc,yloc,zloc, in geocen-eq.
    %corrdinates.  zloc is either sun-facing, nadir-facing, or
    %star-facing. xloc is in the plane of the orbit (in direction of
    %SC velocity if Nadir-facing). And yloc is orthagonal to zloc
    %and xloc to form a right hand coord system.
    if isequal(orientation,'Sun')                %sun-facing
        zloc(k,:) = S(k,:)./norm(S(k,:));
              %SC local axes in Geocen-Eq Coords
        xloc(k,:) = cross(perp_2_orbit_plane,zloc(k,:))./norm(cross...
```

```
    (perp_2_orbit_plane,zloc(k,:)));
    %if xloc perp to orbit plane, cross(perp_2_orbit_plane,zloc)
    %is NaN. Instead define xloc in direction of vel at periapsis
    if isnan(xloc(k,:))
        xloc(k,:) = v_sc(1,:)./norm(v_sc(1,:));
    end
    yloc(k,:) = cross(zloc(k,:),xloc(k,:));
elseif isequal(orientation,'Nadir')                %Nadir-facing
    %SC local axes in geocen_eq coords
    zloc(k,:) = -r_sc(k,:)./norm(r_sc(k,:));
    xloc(k,:) = v_sc(k,:)./norm(v_sc(k,:));
    yloc(k,:) = cross(zloc(k,:),xloc(k,:));
elseif isequal(orientation,'Set R.A. & Dec.')  %Set-orientation
    %SC local axes in geocen_eq coordinates
    zloc(k,:) = RADec2geocen(setraddec(1),setraddec(2),1);
    xloc(k,:) = cross(perp_2_orbit_plane,zloc(k,:))./norm(cross...
    (perp_2_orbit_plane,zloc(k,:)));
    %if xloc perp to orbit plane, cross(perp_2_orbit_plane,zloc)
    %is NaN. Instead define xloc in direction of vel at periapsis
    if isnan(xloc(k,:))
        xloc(k,:) = v_sc(1,:)./norm(v_sc(1,:));
    end
    yloc(k,:) = cross(zloc(1,:),xloc(1,:));
else
    disp('Error: Invalid Orientation Option.')
    break
```

```
end

%Define the normals for all side surfaces

%Initial Orientation: local axes in SC local coords
xloc_inloc = [1,0,0];
yloc_inloc = [0,1,0];
zloc_inloc = [0,0,1];

%rotation matrix to convert from local coords to geocen-eq coords
M = [xloc_inloc;yloc_inloc;zloc_inloc]\[xloc(k,:);yloc(k,:);...
zloc(k,:)];

%normal vectors in SC local coords
n_inloc(k,:,1) = zloc_inloc;
n_inloc(k,:,2) = -n_inloc(1,:,1);
n_inloc(k,:,3) = xloc_inloc;

for j=1:ntot              %loop through nodes
    if j <= nsides        %outside panels
        if j >= 4
            side_ang = 360./(nsides-2);   %side angles (deg)
            n_inloc(k,:,j) = rot3(n_inloc(k,:,3),(j-3)*side_ang);
        end

        %Additional Rotation
```

```
%NOTE: rotating the normals, but not the
%local axes means that you can only do rotation about
%one axis at a time.
rotat=rotation.*t(k)./P
switch rotation_axis
    case '+X'
        n_inloc_rot(k,:,j)=rot1(n_inloc(k,:,j),rotat);
    case '+Y'
        n_inloc_rot(k,:,j)=rot2(n_inloc(k,:,j),rotat);
    case '+Z'
        n_inloc_rot(k,:,j)=rot3(n_inloc(k,:,j),rotat);
    otherwise
        disp('Error: Invalid Rotation Axis Option.')
end

%convert normal vectors from local coords back
%to geocen-eq. coords
n(k,:,j) = n_inloc_rot(k,:,j)*M;

%%%%Calculate Heat Flow%%%

%environmental radiation flux [qs,qa,qe] (W/m^2)
qext_arr(k,:,j) = ext_heat_flux_loop(r_sc(k,:),r_sun,...
n(k,:,j),Gs,AF,Te,abso_sol(j),emis_ir(j));
%environmental radiation input on the node (W)
Qext(k,:,j)= A(j).*sum(qext_arr(k,:,j),2);
```

```
    Qspace(k,:,j)=A(j).*emis_ir(j).*sigma*...
    (0-temp_old(:,:,j).^4);     %radiation to space

else %Now for the inside objects
    Qext(k,:,j) = 0;
    Qspace(k,:,j) = 0;
end

%array of the node temp. differences (linear and 4th power)
temp_dif_mat = reshape((temp_old-temp_old(:,:,j)),1,ntot);
temp4_dif_mat = reshape((temp_old.^4-(temp_old(:,:,j)).^4),...
1,ntot);

%Conduction Heat flow
Qcond(k,:,j) = sum(conductance(j,:).*temp_dif_mat);

%%% Radiation Network %%%
E_mat = transpose(sigma.*(reshape(temp_old,1,ntot)).^4...
.*Res_self_inv);
J_mat = Res_mat\(-E_mat);
Qrad(k,:,j) = sum(transpose(J_mat-J_mat(j,:))...
.*Res_other_inv(j,:));          %Radiative heat Cengel Eq9-50

Qint(k,:,j)= inthl(j);      %Internal Heat source
```

```
    %%%check time-step for stability%%%
    tcheck_cond = 1./C(j).*conductance(j,:);
    ind = find(temp_dif_mat==0);
    tcheck_rad = 1./C(j).*(transpose(J_mat-J_mat(j,:))...
    .*Res_other_inv(j,:))./(temp_dif_mat);
    tcheck_rad(ind) = 0;
    tcheck_ext = 1./C(j).*Qext(k,:,j)./temp_old(:,:,j);
    tcheck_space = 1./C(j).*Qspace(k,:,j)./(-temp_old(:,:,j));
    tcheck_sum(k,:,j) = sum(tcheck_cond)+sum(abs(tcheck_rad))...
    +sum(tcheck_ext)+sum(tcheck_space);
    delta_t_lim(k,:,j) = 1./tcheck_sum(k,:,j);

end

delta_t_lim_min(k) = min(delta_t_lim(k,:,:));   %time-step limit

if delta_t > delta_t_lim_min(k)  %if time-step too large
    delta_t_new = 0.9.*delta_t_lim_min(k);
else
    delta_t_new = delta_t;
end

%New temperatures
temp(k,:,:) = temp_old(:,:,:)+delta_t./(reshape(C,1,1,ntot))...
.*(Qext(k,:,:)+Qspace(k,:,:)+Qcond(k,:,:)+Qrad(k,:,:)+Qint(k,:,:));
```

```
    %Advance for next time loop
    t_old=t(k,:);
    delta_t_old = delta_t_new;
    temp_old = temp(k,:,:);
    k_old=k;

end

for h=1:nsides
    %Plot external radiation on each side
    figure(h)
    plot(t,qext_arr(:,1,h),'--r',t,qext_arr(:,2,h),':b',t,...
    qext_arr(:,3,h),'-.g',t,sum(qext_arr(:,:,h),2),'-k')
    legend('Solar','Albedo','Earth IR','Total')
    xlabel('Time (s)')
    ylabel([nodename(h),' Environmental Heat Flux (W/m^2)'])
end

%Plot the temperature of each node
cell(1,ntot)
col=repmat(['r','g','b','c','m','y','k'],1,37);
figure(h+1)
for hh=1:ntot
    plot(t,temp(:,:,hh),col(hh))
    hold on
    leg(hh)=nodename(hh);
```