

Adaptive Thermal Modeling Architecture for Small Satellite Applications

2Lt. John Anger Richmond, USAF
Colonel John Keesee, USAF Retired

May 2010

SSL # 4-10

Adaptive Thermal Modeling Architecture for Small Satellite Applications

2Lt. John Anger Richmond, USAF
Colonel John Keesee, USAF Retired

May 2010

SSL # 4-10

This work is based on the unaltered text of the thesis by 2Lt. John Anger Richmond submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science at the Massachusetts Institute of Technology.

Adaptive Thermal Modeling Architecture For Small Satellite Applications

by

John Anger Richmond

Submitted to the Department of Aeronautics and Astronautics
on May 11, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The United States Air Force and commercial aerospace industry recognize the importance of moving towards smaller, better, and cheaper spacecraft to support the nation's increasing dependence on space-based technologies. Whether large or small, all spacecraft will require the same basic bus systems and environmental protection, simply scaled to fit the mission. The varying thermal environment in space is particularly important to spacecraft design and operation because of its affect on hardware performance and survivability. The Adaptive Thermal Modeling Architecture (ATMA) discussed in this thesis is meant to bridge the gap between the commercially available thermal modeling tools used for larger, more expensive satellites, and the low-fidelity algorithms and techniques used for simple first order analysis.

The ATMA consists of the MATLAB based Adaptive Thermal Modeling Tool (ATMT) and its user's manual, as well as the process by which an inexperienced engineer can quickly and accurately perform on-orbit thermal trades studies for a range of space applications. The ATMA tools and techniques have been validated with an industry standard thermal modeling program (Thermal Desktop) and correlated to thermal test data taken from MIT's CASTOR nanosatellite. The concepts derived and evaluated within ATMA can be extended to a variety of aerospace modeling applications. The ATMT program and modeling architecture are currently being utilized by members of MIT's Space Engineering Academy (SEA) and undergraduate satellite team as well as the U.S. Air Force Academy's FalconSAT-6 program.

DISCLAIMER CLAUSE: The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government

Thesis Supervisor: Colonel John Keese, USAF Retired
Title: Senior Lecturer

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Acknowledgments

This work was performed from fall 2008 through fall 2009 under AF SMC contract FA8721-05-C-0002 as part of the Space Engineering Academy (SEA) pilot program. The work completed from fall 2009 through summer 2010 was performed under AF SMC contract FA9300-06-C-0023 in continuing support of the full state SEA program. The author gratefully thanks the sponsors for their generous support that enabled this research. The author also thanks the SMC/EA and the U.S. Air Force Academy for their technical support and research guidance.

Contents

1	Introduction	19
1.1	Problem Statement	21
1.2	Thesis Motivation	21
1.3	Thermal Modeling Options	22
1.4	Research Gap Analysis	26
1.5	Thesis Overview	30
2	Background	33
2.1	Spacecraft Size Hierarchy	34
2.2	Small and Nanosat Thermal Control Systems	37
2.2.1	Thermal Margin	37
2.3	Keplerian Orbit	39
2.3.1	Spacecraft Position and Velocity	41
2.3.2	Spacecraft Attitude On-orbit	42
2.4	Heat Transfer in the Space Thermal Environment	43
2.5	Conduction	44
2.6	Radiation	45
2.6.1	External Radiation	46
2.6.2	Internal Radiation	49
2.6.3	Radiation View Factors	50
2.7	Material Properties	52
2.7.1	Thermal Properties	52
2.7.2	Optical Properties	53

3	Adaptive Thermal Modeling Architecture (ATMA)	57
3.1	Prediction Model (ATMT)	58
3.1.1	View Factor Algorithm	59
3.1.2	Ray Polygon Intersection Algorithm	66
3.1.3	View Factor from a Surface to Earth	69
3.1.4	Conductance Algorithm	71
3.2	Thermal Model Validation Techniques	74
3.2.1	Component Level Testing	76
3.2.2	CASTOR Thermal Vacuum Test	78
3.2.3	Summary	82
4	Adaptive Thermal Modeling Tool (ATMT)	85
4.1	Modeling Assumptions	86
4.2	Thermal Model Inputs	87
4.3	Thermal Model Outputs	94
4.4	Model Functionality	98
4.4.1	Test Case Overview	100
4.5	ATMT Validation	115
4.5.1	Validation of Test Case 1: 2D Surface	117
4.5.2	Validation of Test Case 2: 3D Solid Rectangle	119
4.5.3	Validation of Test Case 3: 2D Six Surface Box	121
4.5.4	Validation of Test Case 4: 3D Six Panel Box with Avionics Stack	124
4.6	Model Performance	125
4.6.1	ATMT Accuracy	125
4.6.2	ATMT Usability	126
5	ATMT User's Manual	129
5.1	Overview	130
5.2	ATMT Home Tab	131
5.3	Geometry Tab	132
5.4	View Factors Tab	143
5.5	Conductance Tab	146
5.6	Material Properties Tab	150

5.7	Simulation Tab	155
5.8	Output Tab	162
6	Conclusion and Future Work	167
6.1	Thesis Summary	167
6.2	Contributions	168
6.3	Future Work	169
A	Tables	173
B	Figures	177
C	ATMT Source Code	183
C.1	ray_poly	183
C.2	Conductance	185
C.3	vf_sc_earth	187

List of Figures

1-1	Spacecraft Toolbox Thermal Imaging Demo [14]	25
1-2	ATMT Model Examples	28
2-1	Spacecraft Size Hierarchy	36
2-2	Thermal Margin Terminology for JPL/NASA Programs [7]	38
2-3	Classical Orbital Elements [21]	39
2-4	Determining the Position and Velocity Vectors	42
2-5	Spacecraft Attitude On-orbit	43
2-6	Determining Satellite Lighting Conditions	47
2-7	Radiation Network for Three Nodes [9]	50
2-8	View Factors between Two Isothermal Surfaces	51
3-1	ATMA Block Diagram	57
3-2	ATMT Block Diagram	59
3-3	View Factors Between Two Isothermal Surfaces	61
3-4	Comparison of View Factor Calculation Times for Two Algorithms	61
3-5	Surface and Vertex Numbering Scheme	62
3-6	View Factors Between Surfaces on Solid Objects	64
3-7	Blocked View Factors Example	66
3-8	Internal Objects Shielded from External Heat Fluxes	67
3-9	View Factor from a Surface to Earth	70
3-10	Sample Conductance Matrix Generated by the Conductance Algorithm	72
3-11	Conductance Between Surfaces	73
3-12	Aluminum Block Test Setup and Thermal Desktop Model	76
3-13	Baseline Model-Data Correlation	77

3-14	Final Model-Data Correlation	78
3-15	CASTOR Structural Engineering Model and TVac Test Setup	79
3-16	Thermal Desktop Model of CASTOR Structure and TVac Test Setup	81
3-17	TVac Test Model-Data Correlation	83
4-1	ATMT Geometry Tab: 15 User Defined Input Categories	88
4-2	ATMT Scenario Tab: 24 User Defined Input Categories	91
4-3	ATMT Output Tab: Thermal Model Outputs and Sample Results	95
4-4	ATMT Model Results for the 2D Six Sided Box Example	97
4-5	Test Case 1: 2D Surface Properties and Graphical Display	101
4-6	2D Surface Temperature and External Heat Fluxes	102
4-7	Varying the Surface Finish for a 2D Surface	104
4-8	Increasing the Thickness of a 2D Surface	105
4-9	Test Case 2: 3D Solid Rectangle	106
4-10	3D Solid Rectangle View Factor and Conductance Matrices	107
4-11	Test Case 3: 2D Six Surface Box	108
4-12	2D Six Surface Box View Factor and Conductance Matrices	108
4-13	Temperatures Response for a 2D Six Surface Box	109
4-14	Comparison of the External Heat Fluxes on the -Z and +X Surfaces	110
4-15	Increasing the Conductance Value Between Surfaces	111
4-16	Temperature Response for a 2D Box with +Z in the Velocity Direction	111
4-17	Temperature Response for a 2D Box in a Molniya Orbit	112
4-18	Test Case 4: 3D Six Panel Box with Internal Avionics Stack	113
4-19	Example of Individual Component Orientation	114
4-20	Temperature Response for a 3D Box with an Internal Avionics Stack	115
4-21	Temperature Limits for the +Z Panel and Avionics Stack	116
4-22	2D Surface Modeled in Thermal Desktop and ATMT	117
4-23	Thermal Desktop Validation of Test Case 1: Temperature at the Surface's Centroid	118
4-24	3D Solid Rectangle Modeled in Thermal Desktop and ATMT	119
4-25	Thermal Desktop Validation of Test Case 2: Temperature at the Object's Centroid	120

4-26	2D Six Surface Box Modeled in Thermal Desktop and ATMT	122
4-27	Thermal Desktop Validation of Test Case 3	123
4-28	3D Six Panel Box with Avionics Stack Modeled in Thermal Desktop and ATMT	125
5-1	ATMT at a Glance	129
5-2	ATMT Home Tab	131
5-3	ATMT Geometry Tab	133
5-4	Geometry Tab: Object Editor Panel	133
5-5	Geometry Tab: Object Property Table	135
5-6	Example of a Rectangular and Cylindrical Object	136
5-7	Geometry Tab: Check Facing Example	140
5-8	Geometry Tab: Geometric Configuration	141
5-9	Geometry Tab: Plot External Example	142
5-10	Geometry Tab: Hide Table Example	143
5-11	ATMT View Factors Tab	144
5-12	View Factors Tab: Matrix Editor Panel	144
5-13	View Factors Tab: View Factor Matrix	146
5-14	ATMT Conductance Tab	147
5-15	Conductance Tab: Conductance Table Editor	148
5-16	Conductance Tab: Conductance Matrix	149
5-17	ATMT Materials Tab	151
5-18	Materials Tab: Material Property Editor	152
5-19	Materials Tab: Surface Finish Editor	153
5-20	ATMT Scenario Tab	155
5-21	Scenario Tab: Orbital Parameters Panel	156
5-22	Scenario Tab: Scenario Panel	157
5-23	Scenario Tab: Initial Orientation Panel	158
5-24	Scenario Tab: Orbiting Body Panel	159
5-25	Scenario Tab: Simulation Data File Panel	161
5-26	Scenario Tab: Running the ATMT GUI	161
5-27	ATMT Output Tab	163

5-28	Output Tab: Output Plot Editor	164
A-1	CASTOR Structural Engineering Model Resistor Spreadsheet	175
B-1	TVac Chamber Internal Temperatures	177
B-2	Model Data Correlation for the CASTOR Xenon Tank	178
B-3	Model Data Correlation for the CASTOR +Y Fin	178
B-4	ATMT Material Properties Database	179
B-5	ATMT Test Case Baseline Scenario	179
B-6	Thermal Desktop On-orbit Display of a 2D Surface	180
B-7	Nanosatellite Thermal Test Bed	180
B-8	Nanosatellite Thermal Test Bed: Side Panel Drawing	181
B-9	Nanosatellite Thermal Test Bed: Avionics Stack Drawing	181
B-10	ATMT Results Plotted External to the GUI	182

List of Tables

1.1	Advantages and Disadvantages of Industry Standard Thermal Modeling Programs	22
1.2	Advantages and Disadvantages of Satellite Specific Thermal Algorithms	24
2.1	CASTOR Nanosat Component Temperature Constraints:	37
2.2	Thermal vs. Electrical Circuit Analogy	45
2.3	Environmental Heat Sources: Average, Hot, and Cold Cases	49
4.1	Validation Results: Test Case 1	119
4.2	Validation Results: Test Case 2	121
4.3	Validation Results: Test Case 3	124
A.1	View Factor Calculation Results: Version 1	173
A.2	View Factor Calculation Results: Version 2	173
A.3	Initial Model Data Correlation Results	174
A.4	Final Model Data Correlation Results	174

Nomenclature

AF Albedo Factor

ATMA Adaptive Thermal Modeling Architecture

ATMT Adaptive Thermal Modeling Tool

CASTOR Cathode Anode Satellite Thruster for Orbital Repositioning

CCDs Charged-Coupled Devices

CDD Capabilities Development Document

CDIO Conceive Design Implement Operate

CDR Critical Design Review

COEs Classical Orbital Elements

CONOPs Concept of Operations

COTS Commercial Off the Shelf

CPD Capabilities Production Document

DCFT Divergent Cusped Field Thruster

DoD Department of Defense

EELV Evolved Expendable Launch Vehicle

EHF External Heat Flux

EPS Electrical Power System

ESPA EELV Secondary Payload Adapter

FAQ Frequently Asked Questions

FEM Finite Element Model

FOV Field of View

FRR Flight Readiness Review

GN&C Guidance Navigation and Control

GPS Global Positioning System

ICD Initial Capabilities Document

IJK Geocentric-equatorial Coordinate System

IMU Inertial Measurement Unit

IR Infrared

JPL Jet Propulsion Laboratory
KDP Key Decision Point
LEO Low Earth Orbit
LOS Line-of-sight
MILspec Military Specification
MIT Massachusetts Institute of Technology
MLB Motorized Light Band
MLI Multi-layered Insulation
MPPT Maximum Peak Power Tracker
NASA/JSC National Aeronautics and Space Administration/ Johnson Space Center
OOH On-orbit Handbook
PDR Preliminary Design Review
PDU Power Distribution Unit
PPU Power Propulsion Unit
R.A.&Decl. Right Ascension and Declination
SCT Spacecraft Control Toolbox
SEA Space Engineering Academy
SEM Structural Engineering Model
SMC Space and Missiles Systems Center
SR Steradian
SRR System Requirements Review
STK Satellite Tool Kit
STP Space Test Program
TCS Thermal Control System
TD Thermal Desktop
TSS Thermal Synthesizer System
TVac Thermal Vacuum
UCCS University of Colorado Colorado Springs
uitable User Interface Table

UNP University Nanosatellite Program

USAFA United States Air Force Academy

VF View Factor

3U 3 Unit

Chapter 1

Introduction

The driving force in the 21st century world of modern aerospace technology is the idea of getting “more for less.” This means that aerospace engineers must be able to design smarter, more efficient, and more reliable systems using less money, less time, and less manpower. These requirements have led to a culture shift in the aerospace industry from billion dollar school bus-sized spacecraft, traditionally produced by aerospace conglomerates, to multi-million dollar small satellites (smallsats) designed, built, and operated by small businesses and universities. Supporting the demands of this new “lean enterprise” generation of spacecraft will require innovative aerospace technologies and procurement techniques.

The acquisition of a smallsat follows a similar design sequence to that of a large satellite simply scaled to meet the mission objectives. The primary phases of the satellite acquisition sequence are: conception, design, build, testing, launch, and operations. All phases inevitably are shortened and restructured in order to develop smallsats cheaper, faster and with less manpower. Smallsats take advantage of their size and relative importance by using either components designed in-house which are often non-space qualified hardware, or by purchasing commercial of the shelf (COTS) hardware. Often times only critical subsystems will have multiple inhibits or redundant components. The combination of un-proven hardware, fewer inhibits, and less redundancy introduces risk and the perception that smallsats are unreliable in the space environment.

The tradeoff between increased risk and resource constraints is inherent to a smallsat

program. The risks manifest themselves as on-orbit hardware and software failures due to the harsh environmental conditions in space. The space environmental conditions include: near zero gravity, magnetic fields, ambient temperature, vacuum, and particles (radiation, space debris, and charged particles) [18]. The ambient temperature, vacuum, and radiation effect can be combined into the space thermal environment. Rarely do thermal related issues lead to mission failure, but they do have a significant impact on system performance and reliability. For this reason a satellite's response to the space thermal environment is consistently an area of emphasis.

Many of the risk and reliability concerns can be overcome through extensive modeling and simulation of a satellite's behavior in the space environment. Innovative modeling and testing processes are allowing the transition from larger, more expensive satellites to smaller, cheaper, and more expendable satellite programs. In particular, the thermal modeling and testing tools and techniques used in smallsat programs have become a major area of research and are the motivation for this thesis.

Satellite thermal analysis requires modeling and testing of both the launch and space environments. This task is often complex, expensive, and time consuming. Sacrifices are made by testing specific components or areas of concern and using the results to validate the thermal models and analysis. Once validated, the model can be used to design a Thermal Control System (TCS) that either passively maintains thermal equilibrium or is actively responsive to the changing thermal environment in space.

The existing thermal modeling tools fall into two categories: commercially available industry standard tools and satellite specific algorithms. Both have advantages and disadvantages but neither is fully suited for a smallsat program. Bridging the gap between the commercially available industry standard tools and the satellite specific algorithms will require a modeling architecture that is flexible to thermal modeling applications throughout all phases of the smallsat acquisition lifecycle. The Adaptive Thermal Modeling Architecture (ATMA) presented in this thesis provides a modeling tool, model validation methodology, and the model implementation techniques to satisfy the thermal modeling requirements and constraints of a smallsat program.

At the center of ATMA is the Adaptive Thermal Modeling Tool (ATMT). The goal

of ATMT was to simplify the thermal modeling process while offering functional tools and sufficient accuracy for a smallsat program. The modeling tool is flexible to numerous smallsat applications with the potential to model a variety of mission profiles. ATMT's user friendly interface allows an engineer to quickly develop a thermal model, run orbital simulations, and use the results to make important TCS design decisions. Ideally an engineer with little knowledge of satellite thermal design and control could use ATMT to predict, test, and understand a spacecraft's thermal response in order to meet the satellite's thermal requirements.

The ATMA modeling tools and techniques have been tested and validated using the MIT CASTOR satellite. Currently, ATMA is being tested on the MIT CASTOR and Exoplanet satellite's as well as the U.S. Air Force Academy's FalconSAT-6. The university smallsat arena provides the perfect setting to demonstrate the capability of new modeling architectures. In comparison to commercial or military space programs, university space research programs have relatively low budgets, short schedules, and inexperienced manpower. For these reasons, the thermal modeling techniques discussed in this thesis can be implemented and evaluated first at the university level and if successful, applied to any small satellite program.

1.1 Problem Statement

Currently the modeling tools and techniques used to simulate a satellite's thermal on-orbit lifecycle are geared toward programs with large budgets, extended schedules, and designated experts. Small satellite programs cannot afford large margins in these areas and as a result they frequently struggle and often fail due to thermal design complexities.

1.2 Thesis Motivation

Adapt the current thermal modeling tools and techniques to smallsat programs with low budgets, short schedules, and inexperienced engineers. Use ATMA to design and implement an efficient, functional, and sufficiently accurate thermal modeling approach for small satellite applications.

1.3 Thermal Modeling Options

Spacecraft thermal modeling is used primarily for the design and evaluation of a spacecraft’s Thermal Control System (TCS) whose purpose is to ensure the survivability and performance of the payload and spacecraft subsystems. In the past, large space programs have relied on commercially available industry standard thermal modeling tools and highly skilled thermal engineers to model a spacecraft’s on-orbit thermal response. Today, the scale of space programs is moving toward the small and nano-class range which require a different set of thermal modeling tools and techniques. Due to time, budget, and personnel constraints, many small satellite programs are moving toward satellite specific thermal algorithms for TCS design and analysis. This section compares the industry standard tools to the satellite specific algorithms to determine how well each is able to meet the thermal modeling requirements of a smallsat program.

Industry standard tools and satellite specific thermal algorithms each have distinct advantages and disadvantages. The advantages of using an industry standard tool are accuracy, modeling flexibility, heritage, and pre and post processing capabilities while the disadvantages are cost, modeling time, and complexity. Table 1.1 shows a summary of the advantages and disadvantages for industry standard thermal modeling programs.

Table 1.1: Advantages and Disadvantages of Industry Standard Thermal Modeling Programs

Advantages	
Accuracy	Highly accurate: ± 1 Kelvin
Flexibility	Many ways to model the same thing; very detailed models
Heritage	Proven to work; used for multiple mission applications
Pre/Post Processing	Extensive construction, visualization, and analysis tools
Disadvantages	
Cost	$\sim \$2k \rightarrow \sim \$10k$ per license per year
Time	Time to learn software; time to build and test models
Complexity	Multitude of variable parameters
Generic	User starts from scratch; no baseline design

There are a variety of commercially available thermal-analysis software packages recommended for spacecraft applications [7].

- Thermal Desktop (TD) by Cullimore and Ring Technologies; TD requires AUTOCAD
- Thermal Synthesizer System (TSS) by SPACEDESIGN under license to NASA/JSC
- THERMICA by Network Analysis Inc. under license to ASTRIUM
- FEMAP/SINDAG Modeling System by Network Analysis Inc.
- IDEAS TMG Thermal Modeling System by MAYA
- ITAS by Analytix Corporation
- Thermal Analysis System by Harvard Thermal

Thermal Desktop (TD) is one of the most commonly used programs for smallsat applications. As an extension to AutoCAD, TD offers a full computer aided design (CAD) modeling suite that allows the user to customize component geometries and orientations. TD offers an extensive array of tools that allow the user to specify material properties, surface finishes, and meshing dimensions. After constructing and orienting a series of components the user must define the contact method between each component. Finally, the user is able to create an on-orbit scenario for thermal analysis. Many flexible options allow the user to tweak the model through each step of the process. At times these options are overly complex adding ambiguity and uncertainty to the model. TD requires extensive training and an understanding of the heat transfer mechanisms in the space environment. However, a complete user's guide and technical support make TD an extremely useful commercial tool for small spacecraft thermal modeling.

While platforms such as TD, Solidworks, and Abaqus certainly have the capacity and flexibility to accomplish a wide variety of thermal analyses, generating and testing a comprehensive thermal model requires extensive time and user expertise. The satellite specific thermal algorithms are a means of generating sufficiently accurate results with less time and experience. Their advantages include lower cost, higher specificity, simplicity, and ease-of-use while their disadvantages include reduced flexibility, accuracy, and pre and post processing capabilities. Table 1.2 shows a summary of the advantages and disadvantages for satellite specific thermal algorithms.

Table 1.2: Advantages and Disadvantages of Satellite Specific Thermal Algorithms

Advantages	
Cost	Open source code - \$free
Specificity	Allows user to analyze issues that are geometry/mission specific
Simplicity	Utilizes simplifying assumptions
Ease of Use	Generate quick sufficiently accurate results for simple models
Disadvantages	
Development Time	Code must be modified for various geometries and scenarios
Flexibility	Limited flexibility to perform trade studies
Accuracy	The use of simplifying assumptions lead to inaccuracy
Pre/Post Processing	User left to analyze and interpret a model's results

The demand for smaller, better, cheaper satellites has led to the development of many new professional and educational satellite modeling tools tailored toward smallsat community. Princeton Satellite Systems offers a professional commercially available integrated satellite design and analysis tool known as the Spacecraft Control Toolbox (SCT). The MATLAB-based SCT offers analysis tools for a variety of spacecraft systems with a focus on Guidance Navigation and Control (GN&C) applications.

Thermal modeling is offered as a module within SCT. The Thermal Module offers several very useful features however, it comes with the caveat that it is meant to “help GN&C analysts uncover potential thermal problems caused by GN&C activity,” and not as a substitute for programs such as TD, Solidworks or Abaqus [14]. The thermal modeling tools allow the user to get a basic idea of the equilibrium temperatures at the satellite level. This means that temperatures are projected for a few key nodal points such as the Earth-facing side or Sun-facing side. Figure 1-1 shows a visual of a satellite modeled using SCT’s thermal imaging function. The main limitation with the SCT Thermal Module is that the thermal algorithms are meant to provide a first order estimate of the temperatures. Smallsat thermal modeling beyond the concept study phase requires the ability to easily increase the number of components while modifying their geometry, spatial location, and operating parameters through time.

SatTherm is a MATLAB and Excel-based small spacecraft thermal analysis tool developed by Cassandra VanOutryve in collaboration with the Mission Design Center at NASA Ames [24]. The goal of the SatTherm project was to develop a conceptual design phase tool

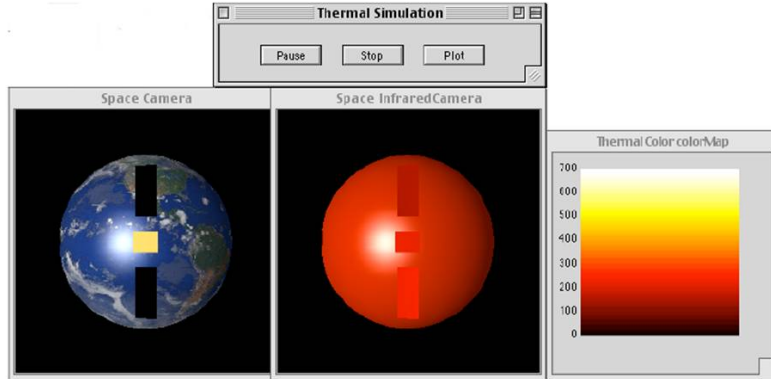


Figure 1-1: Spacecraft Toolbox Thermal Imaging Demo [14]

that could model the on-orbit temperature response for basic spacecraft configurations. The advantage of the SatTherm program is its ease-of-use and ability to perform quick trade studies on material properties and orbital scenarios.

The SatTherm program uses an Excel input file to run a series of thermal algorithms in MATLAB. The MATLAB algorithms generate the transient temperature and external heat flux results for an on-orbit simulation. The Excel input file allows the user to specify the shape (square box, hexagon, or octagon) and material properties of the spacecraft. All of the objects in SatTherm are modeled as 2D surfaces that both radiate and conduct heat to one another. The Excel input file also controls the on-orbit scenario parameters. These parameters include the simulation run-time, orbital elements, spacecraft orientation, and solar position.

The algorithms used in the SatTherm program were validated against TD with an accuracy of ± 4 Kelvin. SatTherm satisfies fulfills its purpose as a conceptual design and analysis tool. Unfortunately SatTherm is limited to a very narrow range of small spacecraft thermal modeling applications. The geometric constraints make it difficult to model spacecraft configurations that deviate from the three prescribed shapes and the 2D surface approach reduces the model's accuracy for truly 3D objects. While SatTherm is a great basis for small spacecraft thermal modeling, several adaptations and extensions may be added to increase its functionality and its ability to meet the thermal modeling requirements of a smallsat program.

1.4 Research Gap Analysis

A plethora of good information was attained by evaluating and testing the various industry standard thermal modeling tools and satellite specific thermal algorithms. As shown in Section 1.3, each tool had many strengths and only few weaknesses. However, through experience with two university smallsat programs it seems that spacecraft thermal modeling remains a daunting task for the inexperienced thermal engineer. As the 2007-2008 Chief Engineer for the U.S. Air Force Academy's FalconSAT-5 satellite and the 2008-2010 lead thermal engineer for the MIT CASTOR program, the author has witnessed many of the challenges associated with thermal modeling in smallsat programs.

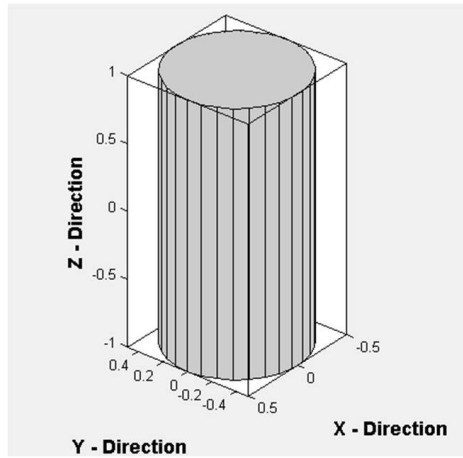
Due to the time, budget, and personnel constraints placed on smallsat programs, designing and testing a spacecraft's Thermal Control System (TCS) usually comes as a last priority. Instead, most smallsat programs rely on thermal modeling to predict and analyze their spacecraft's on-orbit thermal performance. Thermal modeling occurs throughout multiple phases of a program from initial conception through testing and into the on-orbit operations [7]. The Adaptive Thermal Modeling Architecture (ATMA) presented in this thesis, is a set of techniques designed to address the thermal modeling requirements of each phase. The focus of this architecture is on the design, validation, and implementation of the Adaptive Thermal Modeling Tool (ATMT). ATMT is a MATLAB-based thermal modeling tool designed for smallsat applications. The goal of ATMT is not to replace the industry standard or satellite specific algorithms, but to supplement their use in the smallsat realm. ATMT can be used for thermal design, on-orbit modeling, and data analysis at the component, assembly, and vehicle levels. A description of each level is provided below along with examples from ATMT shown in Figure 1-2.

Levels

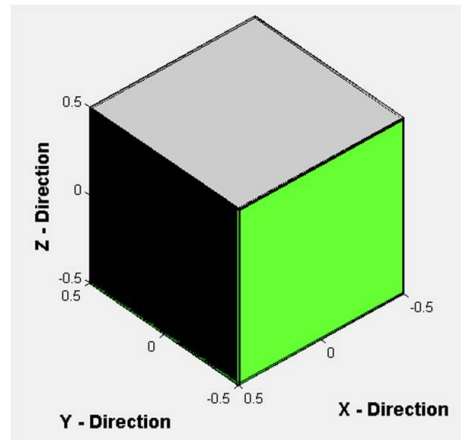
- **Component** - A component is a standalone functional unit that usually requires an interface (mechanical, thermal, electrical etc.) to other units in order to operate. Examples include a battery box, a structural panel, or a xenon propellant tank. Figure 5-6(b) shows an example of 3D cylindrical tank modeled in ATMT.
- **Assembly** - An assembly is the aggregate of multiple components by means of one

or more types of interfaces. Examples include the Electrical Power System (EPS), Thermal Control System (TCS) or the spacecraft structure. Figure 5-6(a) shows an example of satellite's box structure modeled in ATMT.

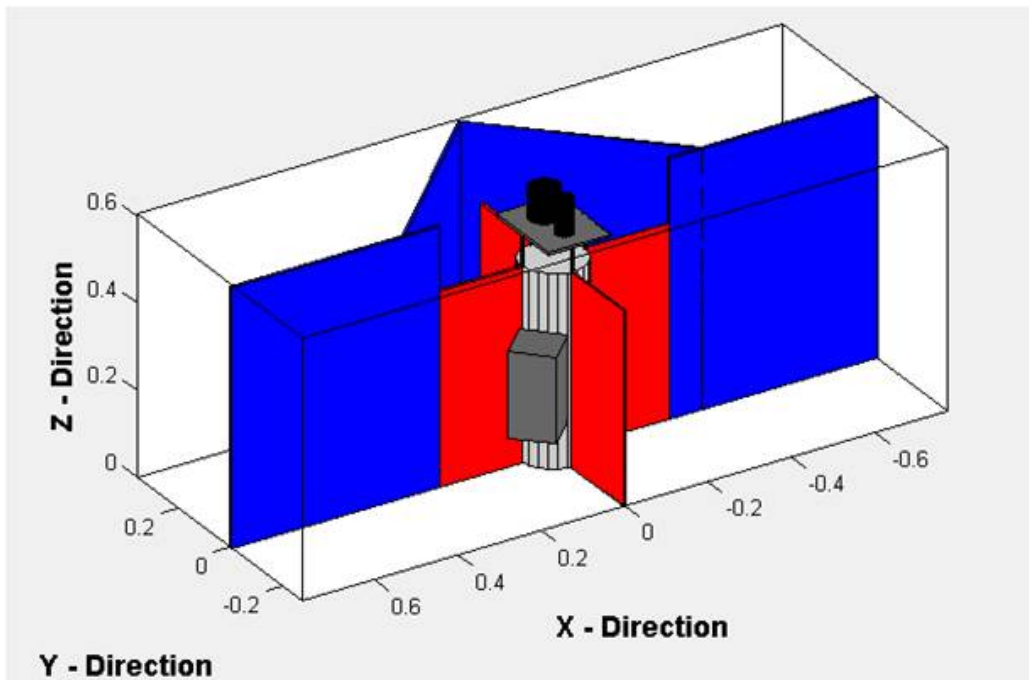
- **Vehicle** - The vehicle includes all of the assemblies integrated into an independently controlled and operated system. The vehicle will undergo all integrated modeling and testing and will be launched as a payload using a launch system. The vehicle may or may not remain in communication with the ground and launch segments. Section 2.1 provides a description of the spacecraft hierarchy including large, small, nano, and cube sized satellites. Figure 1-2(c) shows an example MIT's CASTOR nanosat modeled in ATMT.



(a) Cylindrical Xenon Tank (Component Level)



(b) Six Panel Box Structure (Assembly Level)



(c) Massachusetts Institute of Technology CASTOR Satellite (Vehicle Level)

Figure 1-2: ATMT Model Examples

There are many ways to judge the performance of a thermal model and choosing the right metrics is done based on a program's requirements. Section 1.3 presented a comparison of the advantages and disadvantages of both industry standard thermal modeling programs and satellite specific algorithms. The advantages and disadvantages can be translated into a set of performance metrics for ATMT.

1. **Cost** - The goal is for ATMT to be free to educational institutions. A User's Manual and limited source code are available in Chapter 5 and Appendix C.
2. **Accuracy** - The accuracy metric is a measure of how closely the ATMT results match the results from an industry standard tool. The accuracy goal is to attain temperature responses with $< 5\%$ error and < 11 Kelvin absolute temperature difference. The accuracy metric is a quantitative means of determining the capabilities and limitations of the ATMT program. Understanding the capabilities and limitations allows the thermal engineer to apply appropriate margins when necessary.
3. **Usability** - The usability metric is a measure of how well ATMT satisfies the thermal modeling requirements of a smallsat program. Usability is comprised of a subset of metrics including model functionality, specificity, simplicity, and ease-of-use. The usability metric provides a qualitative means of determining the capabilities and limitations of the ATMT program.
 - (a) **Functionality** - As a preliminary design tool, ATMT must be flexible to changing designs and increasing levels of complexity. The functionality of ATMT is demonstrated by its ability to easily design a model, test a given scenario, and post-process the results. A discussion of ATMT's functionality is available in Section 4.4.
 - (b) **Specificity** - The specificity metric is a measure of how closely a model is tied to its source code. Highly specific algorithms will require the user to actually modify code in order to change the model. Clearly this limits the ability to perform trade studies and to implement design changes. On the other hand, non-specific programs often provide multiple means of designing the same model

leaving it up to the user to know which approach is correct. ATMT takes the middle ground by allowing modifications to be made easily within a graphical user interface (GUI) environment while requiring that a consistent design process be followed in order to limit any ambiguity in the design.

- (c) **Simplicity** - Simplicity in a model is critical for the preliminary design phases where accuracy is balanced with the need to perform quick trade studies. ATMT utilizes a set of appropriate simplifying assumptions that reduce the extremely complex thermal processes to standard equations that are well understood and easily validated. While the simplifying assumptions will inevitably lead to some error, they allow an inexperienced thermal engineer to understand the processes that occur within the model.
- (d) **Ease-of-Use** - The ease-of-use metric refers to how quickly the user is able to learn ATMT and how well ATMT meets a set of thermal modeling requirements. ATMT offers an extensive set of pre and post processing tools that were tailored specifically to address the thermal modeling requirements of a smallsat program. The ATMT User's Manual describes these features with examples of potential applications.

1.5 Thesis Overview

The layout of this thesis is meant to take the reader through the motivation, justification, and demonstration of how the Adaptive Thermal Modeling Architecture (ATMA) was developed, tested, and verified.

Chapter 2 describes the background information and mathematics utilized in the Adaptive Thermal Modeling Tool (ATMT). This chapter provides a quick yet comprehensive summary of the thermal modeling basics.

Chapter 3 takes the reader through each element of the three phase ATMA modeling process. Section 3.1 introduces the ATMT model structure and several of its key thermal algorithms while Section 3.2 addresses the ATMA model-to-model and model-data correlation techniques used to validate the ATMT program.

Chapter 4 introduces the ATMT background methodology and simplifying assumptions as well as the simulation's inputs and outputs. Section 4.4 discusses ATMT's functionality and provides several test cases and analyses demonstrating the ATMT modeling techniques. The test case results are then validated using Thermal Desktop. The chapter is concluded by a summary of the usability tests that were conducted at the U.S. Air Force Academy.

Chapter 5 provides a complete ATMT User's Manual describing how to utilize ATMT for spacecraft thermal modeling purposes. The manual takes the reader through the six tabs of the ATMT GUI providing examples for each function and techniques to aid in the modeling process.

Chapter 6 presents a summary of the thesis, contributions to the field, and suggested future work.

Chapter 2

Background

Designing and modeling an effective spacecraft Thermal Control System (TCS) requires an understanding of the top-level mission objectives as well as the physical interactions between components and their operational environment. The mission objectives translate into the system, assembly and component-level thermal control requirements. Thermal control is accomplished through a variety of methods both passive and active that are used to maintain component temperatures within their operational or survival range.

The scope of this thesis is on TCS design and modeling techniques for small and nano satellite programs. Most small and nano-class satellites strive to use passive thermal control. Passive control can be achieved through the use of special surface finishes and proper orientation of heat producing and heat rejecting components. Some spacecraft components are extremely sensitive to changes in their thermal environment and may require additional active control mechanisms to meet their thermal requirements. Active control may be achieved by maintaining a spacecraft's attitude orientation with respect to external heat sources or by using heating and cooling devices to maintain component temperature ranges.

This chapter provides the background math and processes required to understand on-orbit thermal modeling. Beginning with an overview of the spacecraft size hierarchy in Section 2.1, Chapter 2 goes on to describe the orbital mechanics, heat transfer processes, and material properties used for transient thermal modeling in the Adaptive Thermal Modeling Tool (ATMT).

2.1 Spacecraft Size Hierarchy

The size-name relationship for spacecraft follows primarily from the volume of the structure. The major breakpoint is generally defined between primary and secondary payloads, where the primary payload on a given mission will often be physically larger, more expensive, and more complex than any of the secondary payloads. Another spacecraft size distinction is often associated with the deployment mechanism and in turn the mass of the spacecraft. Using these two size-name distinctions, spacecraft sizes can be broken into four categories: large satellites, small satellites (smallsats), nano satellites (nanosats), and cube satellites (cubesats).

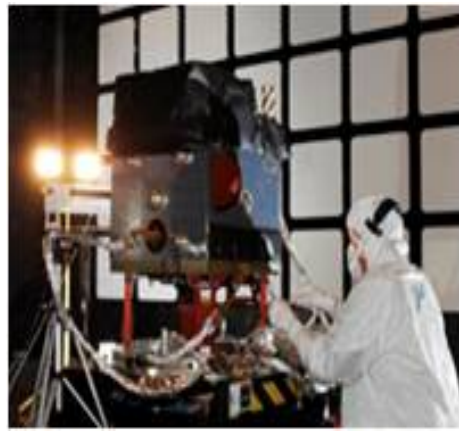
- **Large Satellites** - A spacecraft weighing more than 181 kg or being deployed using a Motorized Light Band (MLB) larger than 15 inches in diameter [5]. Large satellites will range from 181 kg and roughly a cubic meter in volume, to over 10,000 kg and the size of a school bus. Large Satellites will serve as the platform for either one large or several small technical payloads ¹. Milstar-5 shown in Figure 2-1(a) is an example of a large satellite.
- **Smallsats** - A spacecraft weighing less than or equal to 181 kg and greater than 50 kg. These spacecraft are often called ESPA (EELV Secondary Payload Adapter) class spacecraft and will adhere to a very specific set of launch requirements as outlined in the ESPA User's Guide [22]. A smallsat will usually carry one primary payload and may have several supporting secondary payloads. STPSat-2 shown in Figure 2-1(b) is an example of a smallsat.
- **Nanosat and Cubesat** - Derived from the University Nanosatellite Program (UNP) requirements [6], a nanosat is any spacecraft smaller than 50 kg that is separated from the launch vehicle independent of the primary or other secondary payloads. We will assign a lower bound of 4 kg, which is the maximum mass for the standard 3 unit (3U) cube satellite (cubesat). A nanosat is very similar to a smallsat in that they will generally carry one or more small payloads. A cubesat is a standardized 1 → 3 unit

¹A spacecraft's payload refers to the specific experimental apparatus housed by the satellite.

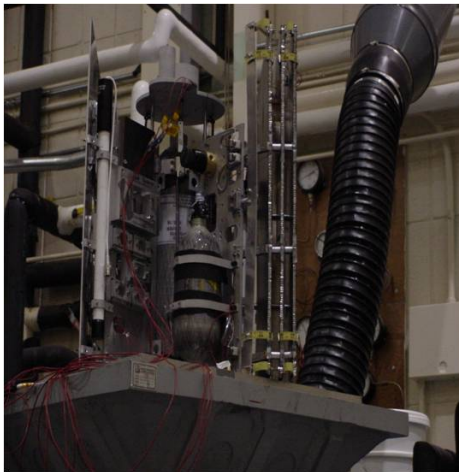
assembly where each unit is a $10x10x10$ cm box containing one or more subsystems. Figure's 2-1(c) and 2-1(d) show examples of MIT's CASTOR nanosat and Space Data System's ISR-3.



(a) Milstar-5 in the Factory (Large Satellite) [15]



(b) STPSat-2 in the Clean Room (Small Satellite) [2]



(c) MIT CASTOR Engineering Model (Nanosatellite)



(d) Space Data Systems ISR-3 (Cube-sat) [16]

Figure 2-1: Spacecraft Size Hierarchy

2.2 Small and Nanosat Thermal Control Systems

The primary objective of the TCS is to keep the spacecraft's components within their required temperature range throughout their lifecycle. Most components are given two temperature ranges; operational and survival. The operational temperature range is the most stringent range that must be maintained while the component is operating to achieve its intended performance. The survival range is the temperature range associated with the non-operating phase of a component's lifetime and provides the maximum and minimum temperatures the component can withstand without being rendered inoperable. Most components are designed to operate nominally at room temperature 273 ± 30 Kelvin. Mission constraints may require a particular component to remain either extremely cold or extremely hot, or even to maintain a very narrow temperature range in order to minimize shock due to thermal cycling. Table 2.1 shows an example of the temperature constraints used for the MIT CASTOR nanosat.

Table 2.1: CASTOR Nanosat Component Temperature Constraints:

Component	Operating Range (K)	Survival Range (K)
Battery	273 \rightarrow 313	263 \rightarrow 318
MPPT	233 \rightarrow 333	228 \rightarrow 353
PPU	233 \rightarrow 358	228 \rightarrow 368
PDU	218 \rightarrow 373	208 \rightarrow 383
Solar Cells	183 \rightarrow 433	173 \rightarrow 443
GPS	253 \rightarrow 323	243 \rightarrow 333
IMU	233 \rightarrow 358	218 \rightarrow 358
Rxn Wheel	253 \rightarrow 333	238 \rightarrow 343
Camera	223 \rightarrow 343	213 \rightarrow 253
Payload	258 \rightarrow 473	243 \rightarrow 358
Temp Sensors	228 \rightarrow 388	218 \rightarrow 398

2.2.1 Thermal Margin

A thermal engineer's primary responsibility is to ensure that component temperature requirements are met throughout all stages of the satellite's operational lifecycle to include: launch, separation, commissioning, normal operations, safe-modes, and decommissioning. To

accomplish this task the thermal engineer must accurately project the worst case hot and cold temperature range for the spacecraft. All spacecraft components have a survivable and operational temperature range and exceeding either at the wrong time can be detrimental to the mission. Modeling a spacecraft's on-orbit thermal response is a critical risk-reduction step in the design phase.

With all of the simplifying assumptions used in thermal modeling and the uncertainties associated with thermal analysis in general, it is nearly impossible to fully characterize a satellite's on-orbit thermal response. Adding margin to the TCS design is an essential means of risk-reduction. Figure 2-2 shows the JPL/NASA thermal margin methodology used for hardware testing and flight operations [7]. Their approach has been adapted by military space programs and is the standard used in ATMA. The diagram shows several layers of margin that should be applied to the thermal model's predicted temperature range throughout the satellite acquisition process.

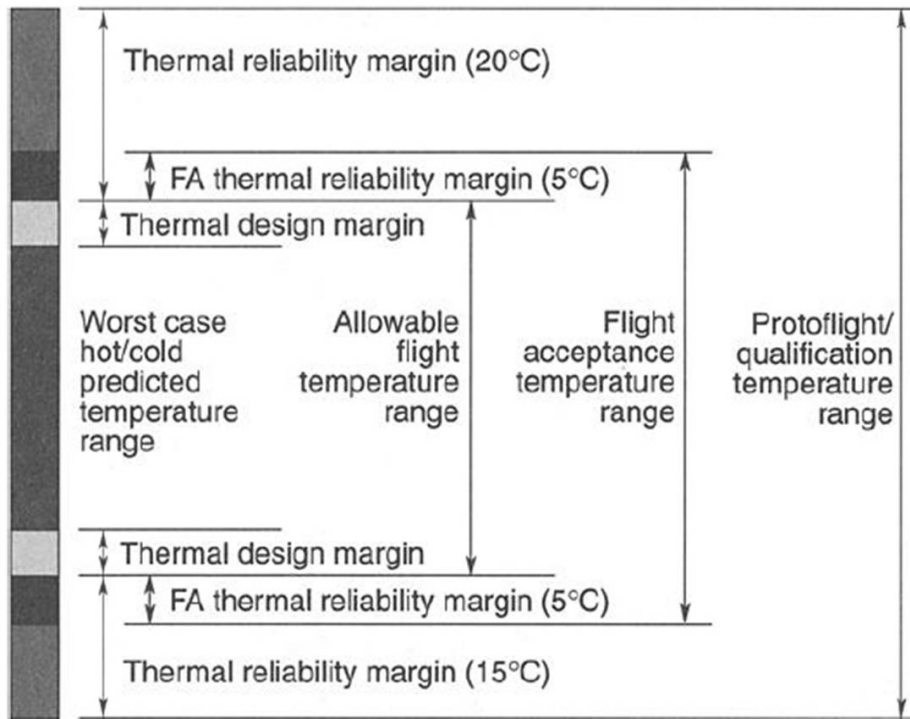


Figure 2-2: Thermal Margin Terminology for JPL/NASA Programs [7]

Even after a satellite's worst case hot and cold temperature ranges are modeled and validated through TVac testing, a thermal margin of ± 11 K is added to account for the

numerous uncertainties and assumptions made in the thermal analysis. On top of that is a flight acceptance margin that is required for acceptance testing of flight hardware. Finally, a thermal reliability margin is applied for the rigorous standards used in thermal qualification testing. The ATMT accuracy requirement discussed in Section 1.4 is based on JPL/NASA's most stringent ± 11 K thermal margin.

2.3 Keplerian Orbit

There are several ways to define a spacecraft's orbit and position in space. The thermal modeling techniques used in the Adaptive Thermal Modeling Tool (ATMT) are based on the six Keplerian element's or Classical Orbital Elements (COEs). The six COEs are: the semi-major axis (a), eccentricity (e), inclination (i), longitude of ascending node Ω , argument of periapsis (ω), and the true anomaly (ν) [18]. Using these COEs, one can define the size, shape, and orientation of the orbit as well as the position of the spacecraft within that orbit. Figure 2-3 shows how the inclination, longitude of ascending node, argument of periapsis, and the true anomaly are defined with respect to an orbital plane of reference.

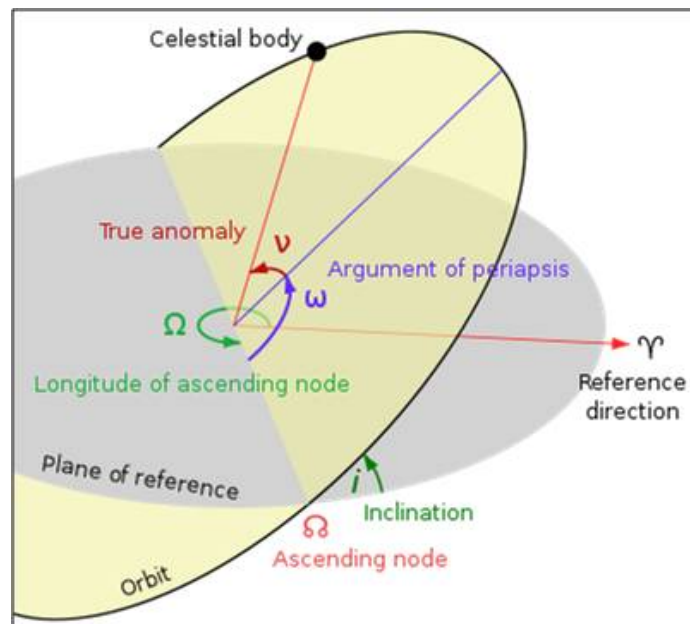


Figure 2-3: Classical Orbital Elements [21]

All of the COEs will be introduced briefly for reference. An in-depth description of each

element and its mathematical derivation can be found in the textbook *Understanding Space* by Jerry Jon Sellers and Associates [18].

1. **Semi-major axis (a)** - The semi-major axis is used to determine the size of an orbit and is related to the specific mechanical energy of the orbit. The semi-major axis is found using Equations 2.1 and 2.2.

$$a = \frac{-\mu}{2\epsilon} \quad (2.1)$$

$$\epsilon = \frac{\|\vec{V}_{SC}\|^2}{2} - \frac{\mu}{\|\vec{R}_{SC}\|} \quad (2.2)$$

where μ ($3.986 \times 10^5 \text{ km}^3/\text{s}^2$) is the gravitational parameter for Earth and ϵ (km^2/s^2) is the specific mechanical energy of the orbit. The specific mechanical energy is found from the magnitudes of the spacecraft's position \vec{R}_{SC} (km) and velocity \vec{V}_{SC} (km/s) vectors.

2. **Eccentricity (e)** - The eccentricity is a measure of the shape of an orbit. Eccentricity ranges indicate the type of orbit with $e = 0$ indicating a circular orbit, $0 < e < 1$ an ellipse, $e = 1$ a parabola, and $e > 1$ a hyperbola. The eccentricity can be calculated directly from the semi-major axis using Equation 2.3 [18].

$$e = \frac{2c}{2a} \quad (2.3)$$

where c (km) is half of the orbital focal length (0 km for circular orbits), and a (km) is the semi-major axis. Alternatively the eccentricity can be determined from the spacecraft's position and velocity vectors \vec{R}_{SC} and \vec{V}_{SC} as shown in Equation 2.4.

$$\vec{e} = \frac{1}{\mu} \left[\left(\|\vec{V}_{SC}\|^2 - \frac{\mu}{\|\vec{R}_{SC}\|} \right) \vec{R}_{SC} - (\vec{R}_{SC} \cdot \vec{V}_{SC}) \vec{V}_{SC} \right] \quad (2.4)$$

3. **Inclination (i)** - The inclination is the angle (deg) from the plane of reference to the orbital plane as shown in Figure 2-3. Inclination may range from $0^\circ < i < 180^\circ$. An

inclination of $0^\circ \leq i < 90^\circ$ indicates a direct or prograde orbit, $90^\circ < i \leq 180^\circ$ is an indirect or retrograde orbit and $i = 90^\circ$ is a polar orbit. The inclination is the first of three parameters (i , Ω , and ω) needed to define the orientation of an orbit in space.

4. **Longitude of Ascending Node (Ω)** - The longitude of ascending node also referred to as the right ascension of the ascending node (RAAN), is the second parameter used in determining the orbit's orientation. RAAN is the angle (deg) from the reference direction (γ) or (I) in the geocentric-equatorial (IJK) frame, to the ascending node of the orbit. Figure 2-3 shows Ω as it is measured from the reference direction γ to the ascending node in the plane of reference.
5. **Argument of Periapsis (ω)** - The argument of periapsis or argument of perigee (argP) is the third element used to describe an orbit's orientation in space. The argument of perigee is the angle (deg) between the ascending node line and perigee of the orbit measured in the orbital plane.
6. **True Anomaly (ν)** - With the size, shape, and orientation of the orbit defined, all that is left is to determine the spacecraft's position within its orbit. The true anomaly is the angle (deg) from perigee to the position of the spacecraft within its orbit. Figure 2-3 represents the spacecraft as the "Celestial Body" with true anomaly measured in the orbital plane in the direction of motion.

2.3.1 Spacecraft Position and Velocity

There are several methods for determining a spacecraft's geocentric-equatorial position \vec{R}_{SC} (km) and velocity \vec{V}_{SC} (km/s) vectors. David Vallado, author of *Fundamentals of Astrodynamics and Applications*, offers a variety of potential algorithms as well as open-source code for each method [23]. One of the more common methods of determining the \vec{R}_{SC} and \vec{V}_{SC} vectors, and the method implemented in the SatTherm orbital propagation algorithm (geocen_rv.m), is to use the COEs discussed in Section 2.3. Given a , e , i , Ω , ω , and time (t), it is possible to propagate the \vec{R}_{SC} and \vec{V}_{SC} vectors through each time step of the simulation [24]. The \vec{R}_{SC} and \vec{V}_{SC} vectors shown in Figure 2-4 are used to fix the spacecraft's

attitude on-orbit as well as determine if the spacecraft is in the Sun or eclipse at each time step.

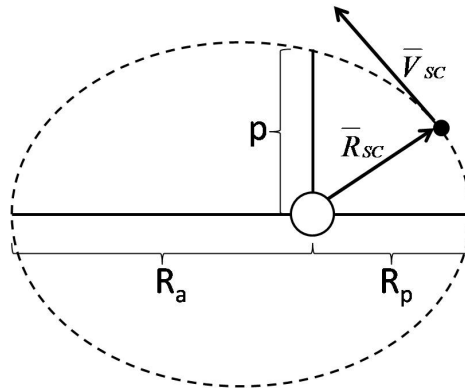


Figure 2-4: Determining the Position and Velocity Vectors

2.3.2 Spacecraft Attitude On-orbit

A spacecraft's attitude on-orbit is a description of its body orientation in the geocentric-equatorial (IJK) coordinate frame. In thermal modeling, a spacecraft's orientation affects how it will absorb heat from external sources such as the Sun and Earth. In actuality, each surface on every component has its own relative attitude that must be rotated from the surface's body frame to the IJK frame. The orientation of each surface with respect to one another governs how heat is radiated internally within the spacecraft and externally to cold space. Fixing a spacecraft's attitude orientation in space is done by choosing a desired pointing direction, selecting the primary axis (+X, +Y, or +Z) to point along that direction, and rotating all of the surfaces from the body frame to the IJK frame accordingly. Some of the most common attitude orientations include: Sun-facing, nadir-facing, velocity-facing, and fixed right ascension and declination (R.A. & Decl.).

Establishing a spacecraft's orientation requires fixing two axes, the primary and a secondary, along two known vectors in inertial space. As discussed in Section 2.3.2, the spacecraft's position \vec{R}_{SC} and velocity \vec{V}_{SC} vectors are the most obvious choices and are used to define the nadir and velocity facing orientations. For the Sun-facing and R.A. & Decl. orientations, two new vectors \vec{S} and \vec{H}_{SC} are used. Where \vec{S} (km) is the spacecraft to Sun vector and \vec{H}_{SC} (km) is the vector perpendicular to the orbital plane at the position of the

satellite. Figure 2-5 shows these four vectors in the IJK frame.

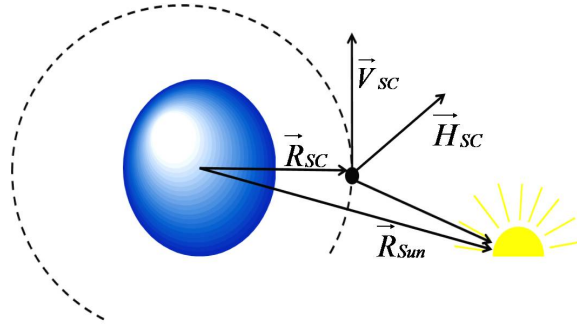


Figure 2-5: Spacecraft Attitude On-orbit

Using these four vectors the primary body axis is aligned with the chosen orientation by setting it equal to the vector's magnitude. The secondary axis is then set to the magnitude of a second chosen direction with the third axis completing the right-hand rule. If needed, the spacecraft may be rotated off-axis by an additional amount or it may have an angular rotation about one or more axes. Equation 2.5 shows how to align the +Z, +X, and +Y body axes for a +Z Sun-facing orientation. The secondary axis +X is aligned in the velocity direction with +Y determined by the cross product of the +Z and +X axes.

$$Z_{body} = \frac{\vec{S}}{|\vec{S}|} \quad (2.5a)$$

$$X_{body} = \frac{cross(\vec{H}, \vec{Z}_{body})}{|cross(\vec{H}, \vec{Z}_{body})|} \quad (2.5b)$$

$$Y_{body} = cross(\vec{Z}_{body}, \vec{X}_{body}) \quad (2.5c)$$

2.4 Heat Transfer in the Space Thermal Environment

The thermal environment in space directly impacts a spacecraft's performance on-orbit by controlling how energy, in the form of heat, is transferred to and from a spacecraft. Operating in the thermal-vacuum environment of space negates convection making conduction

and radiation the dominant modes of heat transfer in space².

Conduction and radiation heat transfer obey both the first and second laws of thermodynamics. The first law requires that the rate of energy transferred into a system be equal to the rate of energy leaving the system such that there is a conservation of energy. The second law states that energy will be transferred in the direction of decreasing temperature [3]. These laws are implemented through the general heat-transfer equation which includes both the internal and external sources of heat described in Section 2.6 [7].

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (K \cdot \nabla T) + q(T, t) \text{ Energy rate per unit volume} \quad (2.6)$$

where ρ (kg/m^3) is the material's density, C_p (J/kgK) is the material's specific heat, ∇ (1/m) is the gradient operator, K (W/mK) is the conductivity tensor, T (K) is the temperature, t (sec) is time, and q (W/m) is the heat input source term. Equation 2.6 forms the basis for both conduction and radiation heat transfer described in Sections 2.5 and 2.6 respectively.

2.5 Conduction

Conduction is heat transfer from the more energetic higher temperature particles of a body to the less energetic cooler particles [3]. This implies that a thermal gradient must exist across the body for conduction to occur. The material properties and geometry of a component will play a significant role in its ability to conduct heat. For simplicity the 1D conduction equation is used in this thesis to model conduction through thin sheets of material. Most spacecraft components are built from thin sheets of material to minimize their mass so the 1D conduction equation is a valid approximation. Equation 2.7 shows the 1D conduction version of the heat-transfer equation.

$$\dot{Q}_{cond} = \dot{m}C_p\Delta T = -kA\frac{dT}{dx} \quad (2.7)$$

²Convection requires a medium such as a gas or liquid for heat transfer to occur. Space is assumed to be a perfect vacuum with negligible atmospheric density. Atmospheric density increases with decreasing altitudes. For extremely low altitudes the affects of convection may no longer be assumed negligible.

where \dot{Q}_{cond} (W) is the rate of heat transfer due to conduction, $\dot{m} = \rho A$ (kg/m) is the mass flow per unit length, C_p (J/kgK) is the specific heat, $\Delta T = \frac{dT}{dx}$ (K/m) is the 1D temperature gradient across the thickness, k (W/mK) is the thermal conductivity, and A (m^2) is the cross sectional area perpendicular to the direction of heat transfer. The conduction between two points on an object or between two different objects may be written in the form [4]:

$$\dot{Q}_{ij} = \frac{-kA}{L}(T_i - T_j) \quad (2.8)$$

where \dot{Q}_{ij} (W) is heat transfer between i and j , L (m) is the thickness, $T_i - T_j$ (K) is the temperature difference between i and j , $\frac{kA}{L}$ (W/K) is equivalent to the thermal conductance. This form of Equation 2.7 alludes to the analogy between thermal and electrical circuits. Quantities such as the potential, flow, resistance, conductance, and capacitance are analogous between the two systems. Table 2.2 shows a summary of the thermal and electrical analogous quantities used for conduction calculations [7].

Table 2.2: Thermal vs. Electrical Circuit Analogy

Quantity	Thermal (Conduction)	Electrical
Potential	ΔT (K)	V (Volt)
Flow	\dot{Q}_{cond} (W)	I (amp)
Resistance	$R = L/kA$ (K/W)	R (ohm)
Conductance	$G = 1/R$ (W/K)	1/R (1/ohm)
Capacitance	$C = \dot{Q}/\Delta T$ (J/K)	C (farad)

2.6 Radiation

The majority of all heat transfer in space is due to the effects of radiation. Radiation heat transfer is the rate of energy absorbed by and emitted from a surface through electromagnetic waves (or photons) [3]. The maximum amount of radiation that can be absorbed or emitted by an object is given by the blackbody radiation equation.

$$\dot{Q}_{rad,max} = \sigma AT_{body}^4 \quad (2.9)$$

where $\dot{Q}_{rad,max}$ (W) is the maximum rate of heat transfer for a black body, σ (5.67×10^{-8}

W/m^2K^4) is the Stefan-Boltzmann constant, A (m^2) is the surface area, and T_{body} (K) is the absolute temperature of the blackbody. Unfortunately the objects used in spacecraft design are never truly blackbodies. In reality, the radiation that is absorbed and emitted from an object is a function of its temperature and surface geometry as well as the wavelength and direction of the incident light. These dynamic properties are extremely difficult to model but can be simplified by using the diffuse-gray and smooth-body assumptions discussed in Section 2.7.2. The result is the addition of an optical coefficient, α for absorbed and ϵ for emitted radiation, that are a function only of the body temperature.

$$\begin{aligned}\dot{Q}_{absorbed} &= \alpha\sigma A(T_{body}^4 - T_{surr}^4) \\ \dot{Q}_{emitted} &= \epsilon\sigma A(T_{body}^4 - T_{surr}^4)\end{aligned}\tag{2.10}$$

In the new radiation equations $\dot{Q}_{absorbed}$ (W) and $\dot{Q}_{emitted}$ (W) are the rates of heat transfer due to absorbed and emitted radiation, ϵ ($0 \rightarrow 1$) is the emissivity, α ($0 \rightarrow 1$) is the absorptivity, A (m^2) is the surface area, T_{body} (K) is the temperature of the body, and T_{surr} (K) is the temperature of the surroundings³.

In spacecraft thermal modeling, radiation can be broken up into two categories; external and internal. External radiation deals with the energy absorbed from external heat sources such as the Sun and Earth as well as the energy emitted by the spacecraft toward cold space. Internal radiation deals with the heat transfer between components on the spacecraft. The geometric orientation of the components plays a major role in the way heat is radiated internally. Sections 2.6.1 and 2.6.2 discuss the methods for determining external and internal radiation.

2.6.1 External Radiation

External radiation accounts for the energy absorbed by the spacecraft from external heat sources as well as the energy emitted by the spacecraft to cold space. The primary sources of external heat include solar flux from the Sun, Sunlight reflected off the Earth (albedo),

³The minimum temperature of cold space is ~ 2.73 K. Cosmic radiation throughout the universe will keep space from ever reaching absolute zero (0 K)

and infrared (IR) energy emitted by the Earth [7]. The amount of external heat absorbed directly from solar energy is a function of an object’s material properties and its orientation with respect to the Sun. Equation 2.11 is used to calculate the solar flux on an object due to direct solar energy.

$$q_{solar} = G_s \alpha_s \cos(\gamma) \quad (2.11)$$

where q_{solar} (W/m^2) is the solar flux absorbed by an object, G_s (W/m^2) is the solar flux constant, α_s ($0 \rightarrow 1$) is the absorptivity of the absorbing surface in the visible range, and γ (rad) is the angle of the absorbing surface with respect to the Sun. The Sun is treated as a diffusely emitting body such that its solar rays are equal in direction, wavelength, and intensity. Clearly the spacecraft must be in the Sun’s line-of-sight (LOS) to absorb its energy. David Vallado offers an LOS algorithm that may be used to determine if the spacecraft is in Sunlight, partial Sunlight (penumbra), or eclipse (umbra). Cassandra VanOutryve’s SatTherm model uses a binary version of the LOS algorithm (insun.m) to determine if the spacecraft is in the Sun (1) or in eclipse (0) [24]. The algorithm extends from Figure 2-6.

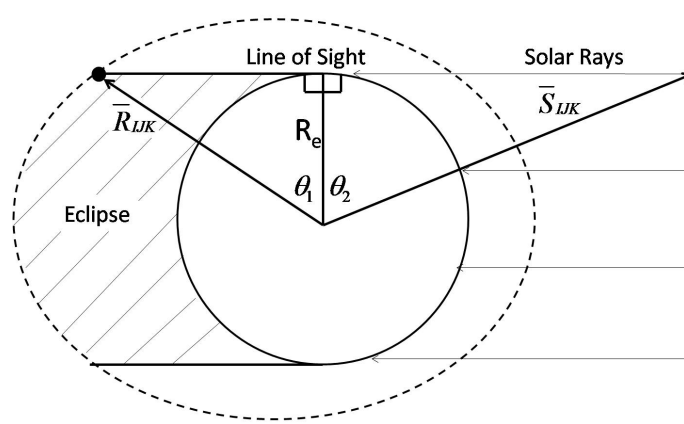


Figure 2-6: Determining Satellite Lighting Conditions

If the sum of the angles (θ_1) and (θ_2) is less than the actual angle between the Sun vector (\vec{R}_{Sun}) and the spacecraft position vector (\vec{R}_{SC}), then the spacecraft is in eclipse. Otherwise, the spacecraft is in the Sun.

$$\text{If } \theta_1 + \theta_2 \leq \cos^{-1} \left(\frac{\vec{R}_{SC} \cdot \vec{S}}{|\vec{R}_{SC}| * |\vec{S}|} \right) \quad (2.12)$$

Then the spacecraft is in the Sun, else it is in eclipse

The second largest heat source after direct solar energy is indirect reflected solar energy from Earth albedo. The Earth is not a diffusely emitting body and simply using the angle of the surface with respect to the Earth is not good enough to achieve the desired accuracy of the ATMT program. Equation 2.13 is used to calculate the external heat flux due to Earth albedo.

$$q_{albedo} = G_s(AF)\alpha_sVF_{i \rightarrow e} \quad (2.13)$$

where q_{albedo} (W/m^2) is the heat flux due to Earth albedo, AF ($0 \rightarrow 1$) is the albedo factor⁴, and $VF_{i \rightarrow e}$ ($0 \rightarrow 1$) is the “view factor” or the amount of a surface’s hemispherical view that is subtended by the Earth. Section 2.6.3 gives a detailed description of view factors (VFs) with the complete surface-to-Earth VF algorithm presented in Section 3.1.

The third source of external heat in space comes from the IR energy emitted by the Earth. Heat flux absorbed in the IR range is treated differently than that of the visible range and requires that the emitting body’s temperature be included in the equation. Equation 2.14 is used to calculate the heat flux due to IR energy emitted by the Earth.

$$q_{IR} = \sigma T_e^4 \alpha_{IR} VF_{i \rightarrow e} \quad (2.14)$$

where q_{IR} (W/m^2) is the heat flux from Earth IR, σ is the Stefan-Boltzmann constant, T_e ($\simeq 250$ K) is the approximate temperature of the Earth as seen by a spacecraft on-orbit, and α_{IR} is the absorptivity of the absorbing surface in the IR range. Summing the external heat fluxes and multiplying by the total surface area A (m^2) gives the total amount of external heat \dot{Q}_{ext} (W) absorbed by an object in space.

⁴The albedo factor is the ability of an object to reflect solar energy.

$$\dot{Q}_{ext} = \sum (q_{solar} + q_{albedo} + q_{IR}) * A \quad (2.15)$$

Unfortunately several of the parameters (G_s , AF , and T_e) involved in the external heat flux equations do not remain constant. Instead, these parameters vary by season, latitude, and the solar cycle. The TCS must be able to meet the spacecraft's temperature requirements for not only the average but also the hot and cold cases of the thermal environment. Table 2.3 provides approximate solar flux, Earth albedo, and Earth IR values for the average, hot, and cold cases. A more detailed description of each heat source is available in Gilmore's *Spacecraft Thermal Control Handbook* [17].

Table 2.3: Environmental Heat Sources: Average, Hot, and Cold Cases

Heat Source (W/m^2)	Average	Hot Case	Cold Case
Solar Flux	1354	1428	1316
Earth Albedo	473.9 ($AF \simeq 0.35$)	571.2 ($AF \simeq 0.4$)	263.2 ($AF \simeq 0.2$)
Earth IR	221.5	261	189

2.6.2 Internal Radiation

Internal radiation accounts for all of the radiation heat transfer that occurs between components on the spacecraft. In this case, the components act as both the heat sinks, receiving heat from external sources, and the heat sources, emitting and reflecting heat to neighboring components according to Equation 2.10. Additionally, all electrical components will have an electrical inefficiency causing some fraction ($0 \rightarrow 1$) of their operating power to be lost as heat. The inefficiency heat loss is constantly being absorbed and emitted by the component.

The objective of this section is show how heat is radiated between components and accounted for in the overall energy balance. To simplify the process the spacecraft is treated as an enclosure such that all of the heat radiated from a particular surface is either absorbed or reflected by other surfaces (actual or imaginary) on the satellite. This concept allows the internal radiation to be modeled as a nodal (surface) network. Figure 2-7 shows the electrical analogy for a three node network.

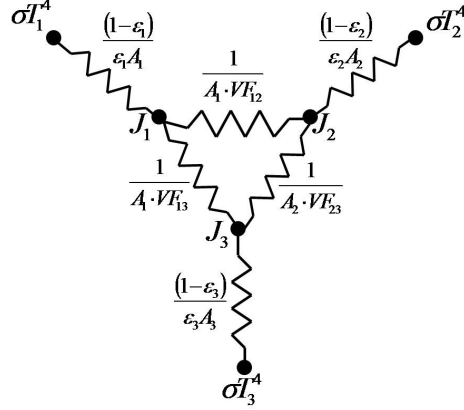


Figure 2-7: Radiation Network for Three Nodes [9]

Using this network analogy it is possible to calculate the heat flow between any two nodes (i) and (j) using Equation 2.16.

$$\dot{Q}_{ij} = \frac{\sigma(T_i^4 - T_j^4)}{\frac{1-\epsilon_i}{A_i\epsilon_i} + \frac{1}{A_iVF_{i \rightarrow j}} + \frac{1-\epsilon_j}{A_j\epsilon_j}} \quad (2.16)$$

where \dot{Q}_{ij} is the heat flow (W) between nodes i and j , $T_i^4 - T_j^4$ (K^4) is the temperature difference between i and j , $\frac{1-\epsilon}{A\epsilon}$ ($1/m^2$) is the surface resistance of a either node, and $\frac{1}{A_iVF_{i \rightarrow j}}$ ($1/m^2$) is the space resistance between nodes i and j . The space resistance includes the VF term which will be discussed in the following section. The total internal radiation for node i includes the input heat \dot{Q}_{input} (W) due to any inefficiencies and the sum of the heat transferred between nodes i and all other nodes ($1 \rightarrow n$).

$$\dot{Q}_i = \dot{Q}_{input} + \sum_{j=1}^n \dot{Q}_{ij} \quad (2.17)$$

2.6.3 Radiation View Factors

The view factor (VF) between two surfaces is the fraction ($0 \rightarrow 1$) of heat radiated from one surface that is incident on a second surface. This fraction is used to calculate the heat transfer \dot{Q}_{ij} (W) between two surfaces i and j as shown in Equation 2.16. Each surface or polygon may be represented as a set of coplanar vertices configured in 3D space. The view factor ($VF_{i \rightarrow j}$) between surfaces i and j is the fraction of surface i 's hemispherical "view"

that is subtended by surface j . Figure 2-8 shows the two polygonal surfaces in an arbitrary orientation.

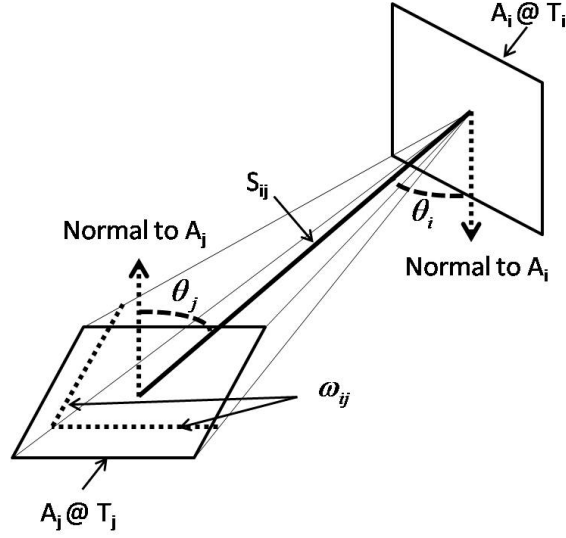


Figure 2-8: View Factors between Two Isothermal Surfaces

The two-dimensional isothermal surfaces can be configured to any size, shape, or orientation in 3D space. The angles θ_i and θ_j are measured from the surface normals to the line S_{ij} and the subtended area ω_{ij} is the solid angle from surface i to j . The solid angle is defined as the ratio of the included area of a segment of a spherical surface to the square of the radius from the origin [9].

$$\omega_{ij} = \frac{A_j}{S_{ij}^2} \quad (2.18)$$

where ω_{ij} (SR or m^2/m^2) is the solid angle from surface i to j , A_j (m^2) is the area of the subtending object, and S_{ij} (m) is the line connecting the centroids of the two surfaces. The double integral shown in Equation 2.19 is ultimately used to calculate $VF_{i \rightarrow j}$. While the VFs are not always equal for a pair of surfaces (i.e. $VF_{i \rightarrow j} \neq VF_{j \rightarrow i}$), calculating the second VF can be done using the transitive relationship shown in Equation 2.20.

$$VF_{i \rightarrow j} = \frac{1}{2\pi A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi S_{ij}^2} dA_i dA_j \quad (2.19)$$

$$A_i VF_{i \rightarrow j} = A_j VF_{j \rightarrow i} \quad (2.20)$$

Calculating the VFs between surfaces becomes extremely difficult and computationally intensive for complex geometric configurations. At times the VFs between two surfaces may even be shaded or blocked by additional surfaces. A thermal model must accurately account for all of the VFs within the system. The ATMT program uses a custom series of VF algorithms to calculate all of the VFs and to populate a VF matrix. Nicolas Lauzier's (`viewfactor.m`) algorithm serves as the core providing the VFs between any two polygons in 3D space. The supporting algorithms described in Section 3.1 then sort the VFs and check for visibility and shading. Determining the VFs between surfaces is essential for modeling a spacecraft's internal radiation on-orbit.

2.7 Material Properties

Spacecraft thermal control relies heavily on the material properties of each component on the vehicle. Two general categories to consider are the material's thermal and optical properties. The thermal properties primarily affect how the material conducts heat while the optical properties affect how the material radiates heat. Both are extremely important and should be considered throughout all levels (component through vehicle) of the satellite design. While conduction only accounts for $\sim 30\%$ of heat transfer in space, selecting a material with the right set of thermal properties can be extremely beneficial for distributing, storing, or isolating heat. Radiation accounts for the other $\sim 70\%$ of heat transfer which means that the material's optical properties play a significant role in the spacecraft's thermal performance.

2.7.1 Thermal Properties

Most satellite missions strive to achieve thermal control through passive techniques. Aside from component placement and mounting configurations, passive thermal control is usually attained through proper material selection [7]. Some components such as batteries and electronics prefer cooler temperature ranges but have a tendency to get hot during normal

operations. Highly conductive heat sinks or radiators are used to distribute the heat to other parts of the satellite or to conduct the heat toward a radiating surface. At the other extreme are components such as ion engines that are required to operate at high temperatures. Insulating materials would be used to maintain the operational temperature range and to protect neighboring components from the effects of such high temperatures. In both of these cases conduction is the primary mode of heat transfer. Section 2.5 showed how conduction is related to the energy balance of a system. Equation 2.7 identified the thermal properties of interest and should be used to select the desired material [3].

Cooling a component such as the batteries or electronics requires an increase in the rate of heat transfer \dot{Q}_{cond} away from the component. This can be accomplished by either selecting a material with a high thermal conductivity such as aluminum, beryllium, or copper or by increasing the cross-sectional area of the heat sink or radiator. These conductive materials generally have lower specific heat and higher densities than their insulating counterparts⁵. Insulators are chosen for their low thermal conductivity which will minimize the amount of heat conducted through the material. It is often difficult to gauge the exact thermal property values for aerospace materials. Fortunately, the Aerospace Corporation has compiled a fairly comprehensive list of aerospace material thermal properties in the *Spacecraft Thermal Control Handbook* [7].

2.7.2 Optical Properties

Radiation to and from a component is highly dependent on the optical properties of the surface finish. Some surface finishes are highly emissive and are used to radiate heat away from a component while others are highly absorbent and are used to draw energy into a component. Utilizing a surface finish's optical properties is the best way to achieve passive thermal control. Components such as the solar arrays are designed to face the Sun for power generation. Inevitably excess heat will be stored on the arrays raising their temperature and degrading their performance. Cooling the arrays is often done by applying a highly emissive coating such as Z93 white paint to the back side of the array. Other components such as

⁵As a rule of thumb the product of the specific heat (C_p) and density (ρ) is nearly constant for all non plastic materials.

the charged-coupled devices (CCDs) used for imaging sensors, are required to operate in a very narrow temperature range. Wrapping the CCD in a Multi-layered Insulation (MLI) blanket will provide insulation by dampening the rate of heat transfer to and from the device. These examples depend on the emissivity (ϵ) and absorptivity (α) of the surface finish. Unfortunately the emissivity and absorptivity are very difficult to determine because they depend heavily on dynamic factors such as the surface texture and temperature of the material as well as the direction and intensity of the incident light. Several key assumptions are used in determining the spectral emissivity and absorptivity for most aerospace surface finishes [20].

Key Assumptions [20]:

- **Diffuse-gray Surfaces** - The spectral emissivity (ϵ), absorptivity (α), and reflectivity (ρ) are independent of wavelength and direction and are solely a function of temperature such that $\epsilon_{(T)} = \alpha_{(T)} = 1 - \rho_{(T)}$.
- The solar and IR absorptivities of a surface are not equal ($\alpha_{solar} \neq \alpha_{IR}$), however the IR emissivity and absorptivity are equal ($\epsilon_{IR} = \alpha_{IR}$).
- **Transparent Space** - The medium between physical surfaces in space is perfectly transparent and has no effect on the radiation interchange between surfaces.
- **Opaque Surfaces** - The radiation heat transfer to and from a physical body occur only at the surface and not within the body.
- **Smooth Planar Surfaces** - Allows heat conduction to occur only through the contact surface area and radiation view factors may be calculated using Equation 2.19.
- **Isothermal Surfaces** - Individual surfaces (represented by a single node at the centroid) are considered uniform in temperature allowing the intensity, direction, and wavelength of the emitted energy to remain uniform over the surface area.
 - **Constant Conductivity** - The conductivity (k) of a material remains constant in the IR temperature range.

- **Constant Emissivity and Absorptivity** - The emissivity (ϵ) and absorptivity (α) of a surface finish remain constant in the IR temperature range.

There are many space-rated surface finishes that are approved for use on smallsats. A list of aerospace material optical properties is made publicly available by NASA/GSFC in reference publication 1121 *Solar Absorbance and Thermal Emittance of Some Common Spacecraft Thermal Control Coatings* [11]. Selecting the proper surface finish is done by modeling the satellite's thermal performance on-orbit to determine which components are outside of their temperature range. Chapters 3 and 4 demonstrate how a surface finish's optical properties affect thermal performance.

Chapter 3

Adaptive Thermal Modeling Architecture (ATMA)

The Adaptive Thermal Modeling Architecture (ATMA) is a three phase process used to develop a thermal modeling tool for smallsat applications. The three phases of ATMA include the development and verification, validation, and implementation of the Adaptive Thermal Modeling Tool (ATMT). The block diagram shown in Figure 3-1 outlines the process that was used to create ATMT.

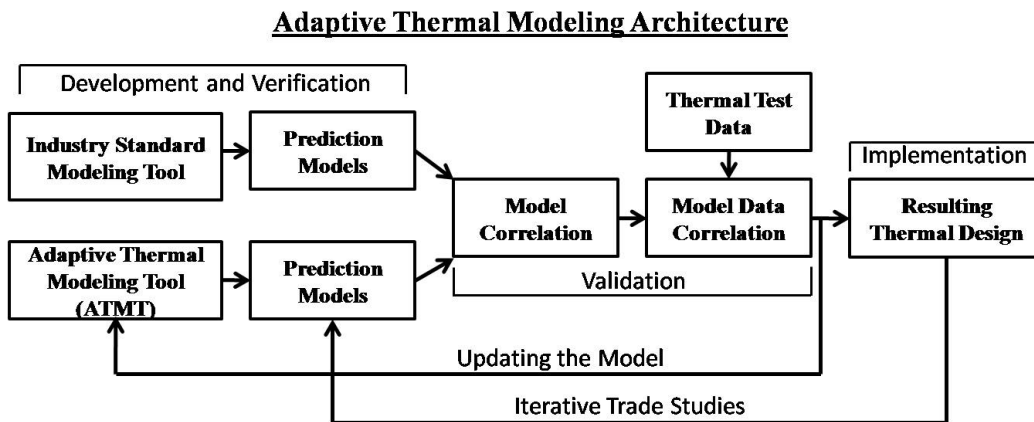


Figure 3-1: ATMA Block Diagram

This chapter follows the first two phases beginning with the development and verification of ATMT. In the development and verification phase, a product is designed, reviewed, inspected, and tested to determine if it meets requirements while satisfying a set of technical

standards [25]. ATMT was designed using the MILspec technical standards to meet the thermal modeling requirements of a smallsat program (see Section 1.4) [7]. The thermal modeling techniques and algorithms used in the develop and verification of ATMT are discussed in Section 3.1.

In the validation phase, a verified product is now evaluated against a certain set of criteria. As discussed in Section 1.4, the two main criteria used to evaluate ATMT are accuracy and usability. In this phase the ATMT prediction models are validated against an industry standard “truth” model as well as actual thermal test data. Section 3.2 discusses the ATMA model-to-model and model-data validation techniques.

3.1 Prediction Model (ATMT)

The Adaptive Thermal Modeling Tool (ATMT) is used to create prediction models of a satellite’s on-orbit thermal performance. The ATMT program is operated through a graphical user interface (GUI), which allows the user to build a thermal model, define an on-orbit scenario and generate predicted results. Following the block diagram shown in Figure 3-2, the GUI first generates an input file containing the material properties, geometric configuration, and the orbital scenario. The Conductance and VF Matrices are calculated internally and stored within the input file. The input file is then passed into the outer loop of the core thermal solver. The outer loop of the solver propagates the satellite’s state matrix (position, velocity, and attitude) through time. The inner loop performs the rotations and thermal calculations for each component before moving forward in time. When the scenario is complete an output file is generated for post-processing. The remainder of this section focuses on the algorithms used to calculate the spacecraft’s internal view factors, the view factors from the spacecraft to Earth, and the conduction between objects.

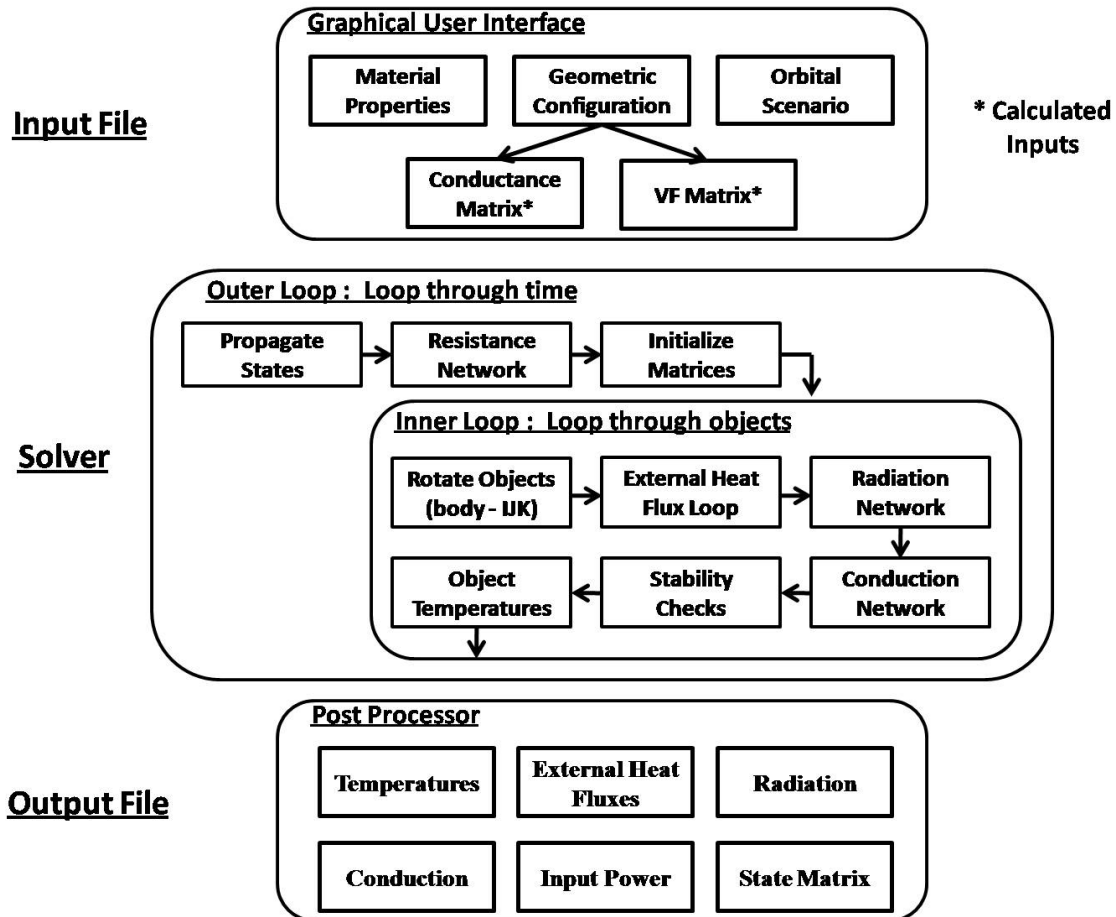


Figure 3-2: ATMT Block Diagram

3.1.1 View Factor Algorithm

The primary form of heat transfer in space is radiation. Radiation is characteristically different from conduction and convection in that it does not require an intermediate medium to transfer energy from one body to another [3]. The fundamental equations for radiation heat transfer discussed in Section 2.6 allow us to understand why radiation in the space environment accounts for the majority of heat transfer both internal and external to a spacecraft. The goal is to determine the fraction of energy emitted from one surface that is absorbed by each neighboring surface. The net energy transfer between two surfaces is a function of the emitting body's surface emissivity (ϵ), the absorbing body's surface absorptivity (α), and the respective 3D orientation of the two surfaces. There are several ways to account for radiation heat transfer within a thermal model. Two of the more

prominent methods are the Monte Carlo ray tracing scheme and the use of view factors (VFs).

The Monte Carlo ray tracing approach to radiation interchange problems relies on the enclosure-theory presented in Section 2.6 and uses a statistical numerical method to simulate the fraction of energy emitted or received by a particular surface [20]. The ray tracing portion of the algorithm is used to simulate rays of energy being released with uniform intensity through all angles from a given surface. The Monte Carlo portion statistically determines how many of the rays should be reflected back to the original surface or received from other surfaces based on the geometric configuration of the objects in space. Thermal Desktop as well as several other industry standard thermal analysis tools use the Monte Carlo ray tracing method because of its computational efficiency and relatively high accuracy.

The second method used for radiation heat transfer is to calculate the VFs between every pair of surfaces using the techniques shown in Section 2.6.3. The VFs populate the VF Matrix, which in turn is used to calculate the thermal resistance network for the model. The VF approach utilizes Equation 2.19 theoretically yielding the exact answer for a given set of objects in any 3D configuration. The only drawback is that the double integral can be computationally tedious and additional measures must be taken to account for directionality, shading, and blockage of the two surfaces. Figure 3-3 shows an example of the VFs between two isothermal surfaces. The centroids of surfaces A_i and A_j are connected by line S_{ij} with θ_i and θ_j being the angles from the respective surface normal vectors to line S_{ij} . In this case the solid angle ω_{ij} represents the fraction of A_i 's hemispherical view that is subtended by A_j or in essence the VF_{ij} .

SatTherm uses the VF approach and requires the user to manually calculate and populate the VF Matrix. For simple geometries this approach is effective as there are a number of closed form solutions available in the literature for common configurations [10]. However, for complex geometries and large numbers of components it is extremely difficult and time consuming to accurately account for all of the VFs by hand. The ATMT program leverages a combination of algorithms to automatically calculate and populate the VF Matrix. The algorithms account for all of the surfaces and their individual geometric orientations. The

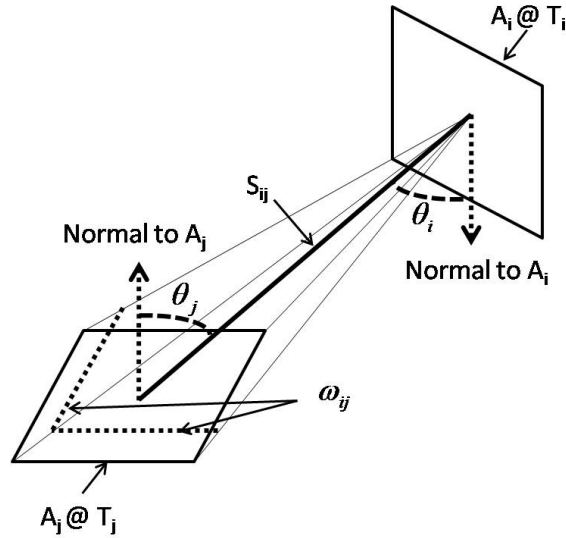


Figure 3-3: View Factors Between Two Isothermal Surfaces

total number of VFs is the number of surfaces squared (N_{surf}^2). While the VF algorithm is the largest time sink of the model, requiring on average 0.041 seconds per VF calculation, the computation time is significantly shorter than would be achieved by hand. Several versions of the VF algorithms have been developed and compared to determine which set of techniques is most accurate and efficient for use in ATMT. The results of two versions are compared in Figure 3-4 with supporting Tables A.1 and A.2 available in Appendix A.

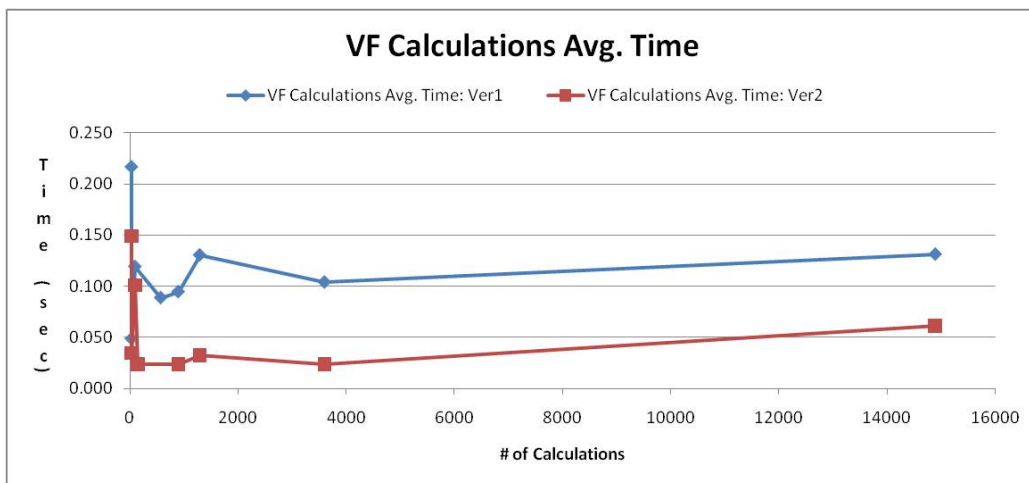


Figure 3-4: Comparison of View Factor Calculation Times for Two Algorithms

For small VF Matrices the average calculation time is actually greater due to the startup costs of opening the algorithm and initializing the variables. As the VF Matrix increases in

size the average calculation time continues to decrease. The complexity of the model plays a significant role in the run-time of the VF algorithms and in the algorithms that are used to calculate the VF Matrix. The basic premise is to identify which surfaces are potentially “in view” of each other by eliminating the impossible candidates. From there, the algorithms must determine if the VFs are shaded, blocked, or so small that they should be truncated to zero. The numbering scheme shown in Figure 3-5 is used to reference particular vertices and surfaces for a given 3D object. It is important to note the surface order (+Z, -Z, +X, +Y, -X, -Y) as it is used to orient the object and ensure the algorithms are implemented properly.

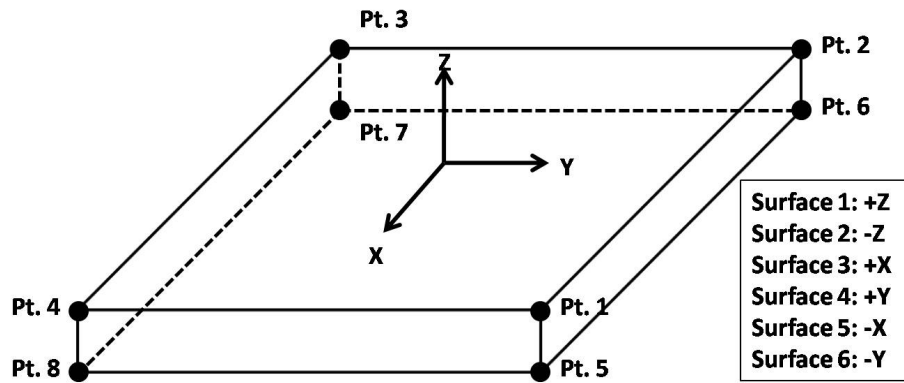


Figure 3-5: Surface and Vertex Numbering Scheme

The VF algorithm runs within a six loop architecture¹ that cycles through all objects and surfaces in order to populate the VF Matrix. The following algorithm outlines the process that is used to calculate the VFs and populate the VF Matrix.

The four objects that will be used in the algorithm are the surface, surface centroid, surface normal vector, surface area, VF Matrix, and Field Matrix.

1. **Surface** - A surface S is defined by a set of P coplanar points where each vertex has coordinates, $V_p = (x_p, y_p, z_p)$.
2. **Surface Centroid** - The centroid of a surface $C_S = (x_c, y_c, z_c)$ is located at the geometric center and is used to define the point at which the surface normal vector

¹The first (outer) loop controls the reference object number. The second loop controls the surface number for the reference object. The third loop handles the object number of the second object in question. The fourth loop controls the surface number for the second object. The fifth loop controls the vertex for the reference surface. Finally the sixth (innermost) loop controls the vertex for the second object.

begins.

3. **Surface Normal Vector** - The normal vector to a surface S_{\perp} faces outward from the centroid of an object.
4. **Surface Area** - The surface area A_S is the total area of surface S used in its VF calculations.
5. **VF Matrix** - The VF Matrix is a $\#S \times \#S$ square matrix containing the VFs between each pair of surfaces within the model.
6. **Field Matrix** - The Field Matrix is an $\#S \times \#S$ square matrix similar to the VF matrix which stores a binary truth value for each VF, where $1 = true$ if $VF > 0$ and $0 = false$ if $VF = 0$.

View Factor Algorithm:

1. **First loop** - For objects $1 \rightarrow \#Obj$, set the reference object variable Obj_i to the first object such that $Obj_i = 1$.
2. **Second loop** - For the surfaces on the reference object $1 \rightarrow Obj_{S_i}$, set the reference surface counter S_j to the first surface such that $S_j = 1$.
3. **Third loop** - For objects $1 \rightarrow \#Obj$, set the comparison object variable Obj_k to the first object such that $Obj_k = 1$.
4. **Fourth loop** - For the surfaces on the comparison object $1 \rightarrow Obj_{S_k}$, set the comparison surface counter S_l to the first surface such that $S_l = 1$.
 - If $Obj_i = Obj_k$, then $VF_{jl} = VF_{lj} = 0$ because two surfaces on the same object are not “in view” of each other. This relies on the assumption that all objects are convex with planar surfaces.
 - Else if $Obj_i \neq Obj_k$ check to see if the surface normals $(S_{j\perp})$ and $(S_{l\perp})$ are in the same direction. If $S_{j\perp} \cdot S_{l\perp} = 1$, then $VF_{jl} = VF_{lj} = 0$ because the surfaces are facing in same direction and will not share a VF.

- Else a VF may exist between surfaces S_j and S_l . One example of this situation is illustrated by Figure 3-6. The image shows a profile view of two solid boxes oriented along the Z axis. The goal is to determine the VFs between surfaces 1 and 9. The two surfaces are on different objects such that $Obj_i \neq Obj_k$ and they are clocked 90 degrees to one another such that $S_{j\perp} \cdot S_{l\perp} \neq 1$.

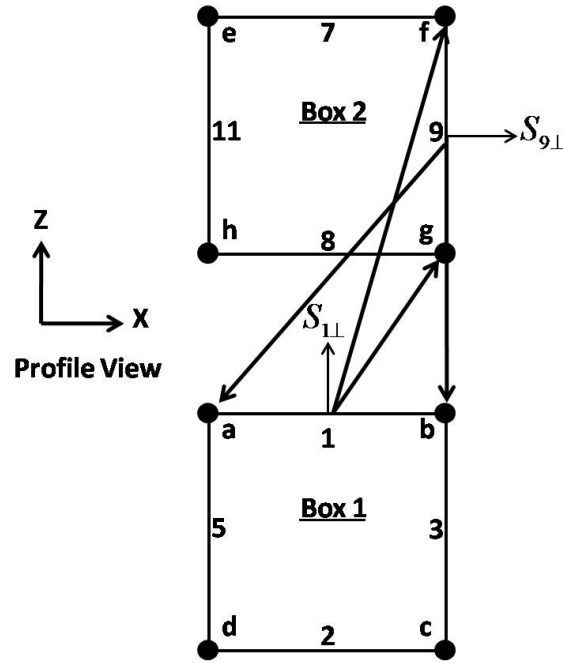


Figure 3-6: View Factors Between Surfaces on Solid Objects

5. **Fifth loop** - For vertices $1 \rightarrow 4$ on surface 9 (remember this is a profile view of surface 9), set the reference surface vertex counter V_m to the first vertex of surface 9 such that $V_m = 1$.

- Draw a line L_1 extending from the centroid of surface 1 (C_1) in the direction of the surface normal ($S_{1\perp}$) using Equation 3.1.

$$L_1 = (C_1 + S_{1\perp}) - C_1 \quad (3.1)$$

- Draw a second line L_2 from C_1 to the first vertex on surface 9 (V_{9f}) using Equation 3.2.

$$L_2 = V_{9f} - C_1 \quad (3.2)$$

- If $L_1 \cdot L_2 > 0$, then the core VF algorithm is run to calculate VF_{19} , VF_{91} , A_1 , and A_9 . The core algorithm was developed by Nicolas Lauzier and Daniel Rouse of Université Laval in 2003 [12]. The algorithm first uses the MATLAB `quadr.m` function to integrate between the points of each surface in order to calculate the two surface areas A_1 and A_9 . The algorithm then uses the MATLAB `dblquad.m` function to evaluate the double integral over the area of each surface as shown in Equation 2.19 to find VF_{19} and VF_{91} .
- Check if the VFs are shaded or blocked. In this example it should be apparent that the 3D nature of the box causes surface 9 to be blocked by surface 8.

6. **Sixth (innermost) loop** - For vertices ($1 \rightarrow 4$) on surface 1, set the comparison surface vertex counter V_n to the first vertex of surface 1 such that $V_n = 1$.

- Draw a line L_1 extending from the centroid of surface 9 (C_9) in the direction of the surface normal ($V_{9\perp}$) using Equation 3.1.
- Draw a second line L_2 from C_9 to the first vertex on surface 1 (V_{1a}) using Equation 3.2.
- If $L_1 \cdot L_2 > 0$, then surface 9 “sees” vertex V_{1a} because the angle is less than 90° . Assign VF_{19} and VF_{91} to their respective locations within the VF Matrix and exit the innermost loop. Increase the counter for the comparison surface to $S_l = S_l + 1$.
- Else if $L_1 \cdot L_2 \leq 0$, surface 9 does not “see” vertex V_{1a} and the next vertex on surface 1 is tested. Increase the counter for the comparison surface vertex to $V_n = V_n + 1$.
- If surface 9 does not “see” any of the vertices on surface 1 then $VF_{19} = VF_{91} = 0$. This indicates that the surfaces are blocked from each other’s view and thus do not share a VF. Increase the counter for the comparison surface to $S_l = S_l + 1$.

7. Repeat the process by looping through all surfaces on all objects until the VF and field matrices are fully populated.

After running the VF algorithm, the VF and Field Matrices will be fully populated. However, the shadowing problem may still exist for certain configurations. The Ray Polygon Intersection Algorithm discussed in Section 3.1.2 is used to address this issue in order to improve the accuracy of the VF calculations. Additional improvements to the VF calculations are suggested as future work in Section 6.3.

3.1.2 Ray Polygon Intersection Algorithm

The basic VF algorithm presented in 3.1.1 was used to eliminate the majority of the inapplicable VFs between surfaces. However, two problems remain unsolved. The first problem is that the VF Matrix may still contain coefficients for surface pairs that are “in view” but are blocked from having a VF. Figure 3-7 illustrates this problem. Up to now the VF algorithm would have allowed for a VF between surfaces 1 and 2. Both surfaces “see” each other because they reside in the outward facing hemisphere and are not blocked by their own 3D box. However, in this case Box B completely blocks the view between boxes A and C.

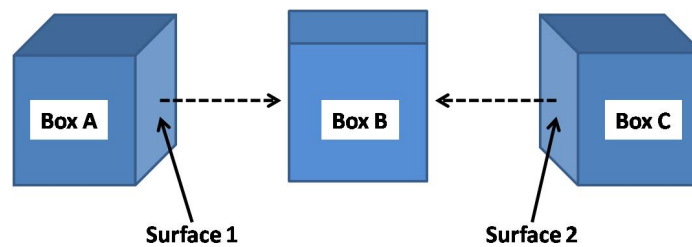


Figure 3-7: Blocked View Factors Example

The second problem comes into play later when the core solver attempts to determine if a surface receives any external heat flux (EHF) from the Sun, Earth IR, or Earth Albedo. While the VFs are fully defined at that point, there is no indication of whether or not the surface is fully shielded from the external space environment. Initially the core solver uses the angle between the surface normal and either the Sun or Earth vector² to calculate

²The Sun vector is the vector from the spacecraft to the Sun in the Earth Centered Inertial (ECI)

the amount of heat a surface receives from external sources. Clearly, if an object resides within a box or is completely shielded, then its surfaces will not receive any EHF's. A flag must be assigned to each surface indicating whether or not it is completely shielded. If the flag is “shielded” then the solver will skip the EHF calculations for that surface and if it is “exposed” then the solver will perform the normal operations. Figure 3-8 shows a likely example of this problem where a battery box is “shielded” by the outer structure. Surface 1 will require an “exposed” flag to allow for EHF calculations while surface 2 needs a “shielded” flag to ensure that it does not receive any EHF's.

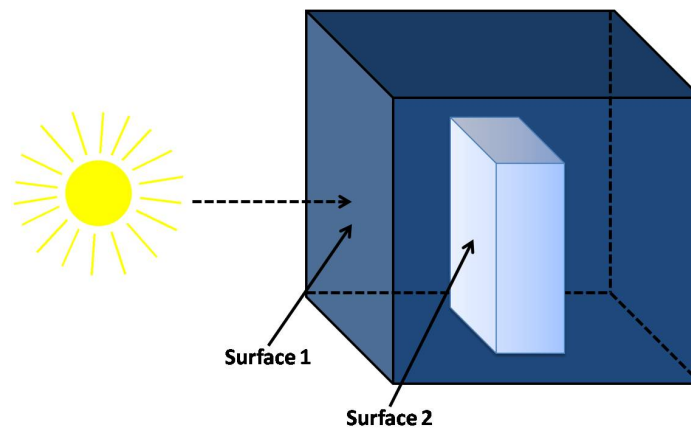


Figure 3-8: Internal Objects Shielded from External Heat Fluxes

These two problems are similar and can be solved using a customized Ray Polygon Intersection Algorithm. The main idea is to extend a ray through space and determine if it goes through the plane generated by the polygon (polygon is synonymous with surface). If the ray intersects the plane then the algorithm must determine if the ray passes through the polygon itself. With the first two criteria satisfied, the algorithm identifies the ray/polygon intersection point³ Finally, a second ray is extended from the intersection point in any direction along the plane. The number of lines or edges of the polygon that are crossed by the second ray determines whether the intersection point was inside (1 crossing) or outside (0 or 2 crossings) of the polygon. The following algorithm was modified from similar computer graphics ray trace methods [8].

coordinate frame. The Earth vector is equivalent to the position vector of the satellite which is also measured in the inertial coordinate frame.

³It is important to note that the ray may pass through either a vertex, line, or the interior space of the polygon.

The four objects that will be used in the algorithm are the polygon, plane, ray, and intersection point.

1. **Polygon** - A polygon S is defined by a set of P coplanar points where each vertex has coordinates, $V_p = (x_p, y_p, z_p)$.
2. **Plane** - A plane is defined by the equation $Ax + By + Cz + D = 0$ with normal vector $S_{\perp} = (A, B, C)$.
3. **Ray** - A ray R begins at a given point $P_1 = (x_1, y_1, z_1)$ and extends in the direction of a second point $P_2 = (x_2, y_2, z_2)$.
4. **Intersection Point** - The intersection point $T_i = (x_i, y_i, z_i)$ is the point of intersection between plane P_n and ray R .

The three main steps are described in detail below [8] with the ray_poly.m code available in Appendix C.

1. Project the polygon onto a 2D plane by dropping the x, y, or z coordinate with the largest magnitude from all points P_n as well as the intersection point $T_i = (x_i, y_i, z_i)$. This creates a new polygon with vertices $P_{uv(n)} = (u_s, v_s)$ and intersection point $T_{uv(i)} = (u_i, v_i)$. For example, if $P_1 = (5, -7, 3)$, drop the y coordinate from all vertices V_p and T .
2. Relocate the polygon to have the ray/plane intersection point become the origin. Subtracting the new $T_{uv(i)}$ from all vertices $V_{uv(p)}$ will place the 2D projected polygon around the intersection point.
3. Extend a ray along any axis and count the number of crossings (NC). If NC is even (either 0 or 2), then the intersection point was originally outside of the polygon. If NC is odd (can only be 1), then the ray passed through the polygon and the intersection point was originally on the inside of the polygon.

- NC is initially set to 0

- Create a sign holder variable (SH) based on the sign of $V_{v(1)}$. If $V_{v(1)} < 0$, then $SH = -1$, else $SH = 1$.
- Create a next sign holder variable (NSH) based on the sign of $V_{v(2)}$. If $V_{v(2)} < 0$, then $NSH = -1$, else $NSH = 1$.
 - If $NSH = SH$, then there is no crossing.
 - Else if $NSH \neq SH$, if $V_{u(1)} > 0$ and $V_{u(2)} > 0$, then there is a crossing and $NC = NC + 1$.
 - Else if either $V_{u(1)} > 0$ or $V_{u(2)} > 0$, then there may be a crossing. Compute the true intersection point $T_{i-12} = V_{u(1)} - V_{v(1)} * (V_{u(2)} - V_{u(1)})$. If $T_{i-12} > 0$, then $NC = NC + 1$, otherwise if $T_{i-12} \leq 0$ there is no crossing.
- Set $SH = NSH$ and loop through all points on the polygon until ending with the first point $V_{uv(1)}$ to close the shape.

4. If NC is odd, then the ray passes through the polygon, otherwise it does not.

3.1.3 View Factor from a Surface to Earth

The view factor VF_{i-e} ($0 \rightarrow 1$) is the fraction of albedo and IR energy that is incident on surface i from the Earth. The VF_{i-e} is calculated from the altitude h (km) of the spacecraft and the angle γ (rad) between the surface normal vector S_{\perp} and the surface's position vector \vec{R}_{SC} as shown in Figure 3-9.

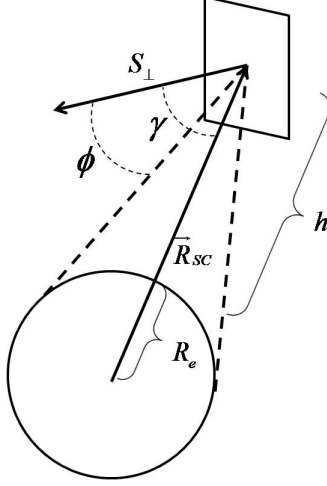


Figure 3-9: View Factor from a Surface to Earth

The surface is at a distance $\|\vec{R}_{SC}\| = h + R_e$ (km) from the center of the Earth. A second angle ϕ is defined between the surface normal S_{\perp} and the scalar H as shown in Equations 3.3a and 3.3b.

$$H = \frac{\|\vec{R}_{sc}\|}{R_e} \quad (3.3a)$$

$$\phi = \sin\left(\frac{1}{H}\right) \quad (3.3b)$$

The angle ϕ is compared with γ to determine whether all, part, or none of the Earth is in view of the surface. These three cases each yield a different VF from the surface to Earth. ATMT evaluates these cases for both 2D surfaces and 3D objects. For a range of orientations it is possible for both sides of a 2D surface to have a partial VF to Earth. For a 3D object, every surface is a face on that object so only the forward facing side is checked for a VF to the Earth. The following algorithm is used to calculate VF_{i-e} for each of the three cases. The `vf_sc_earth.m` code is also available for reference in Appendix C.

Case 1 - The surface sees all of Earth if: $\gamma < (\pi/2 - \phi)$.

$$VF_{i-e} = \frac{\cos(\gamma)}{H^2} \quad (3.4)$$

Case 2 - The surface sees part of Earth if: $\gamma > (\pi/2 - \phi)$ & $\gamma < (\pi/2 + \phi)$

$$b = \sqrt{H^2 - 1} \quad (3.5a)$$

$$t_1 = \frac{1}{2} \left(\arcsin \left(\frac{b}{H \sin(\gamma)} \right) \right) \quad (3.5b)$$

$$t_2 = \frac{1}{2H^2} \left(\cos(\gamma) \arccos(-b \cot(\gamma)) - b \sqrt{1 - H^2(\cos(\gamma))^2} \right) \quad (3.5c)$$

$$VF_{i-e} = \frac{2}{\pi} \left(\frac{\pi}{4} - t_1 + t_2 \right) \quad (3.5d)$$

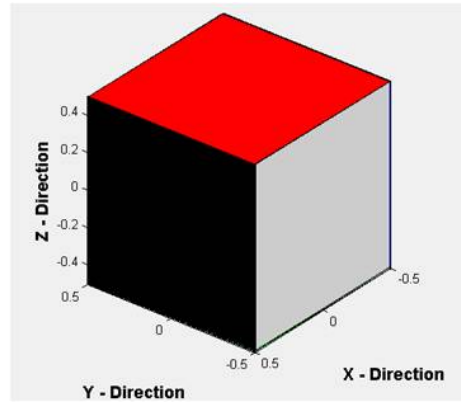
Case 3 - The surface does not see Earth

$$VF_{i-e} = 0 \quad (3.6)$$

3.1.4 Conductance Algorithm

The second form of heat transfer that is addressed within ATMT is conduction. As described in 2.5, conduction is the mechanism through which adjacent objects transfer heat to one another. Similar to the VF Matrix discussed in 3.1.1, a Conductance Matrix is used to relate the conduction heat transfer between objects. There are several ways to determine the amount of heat transferred through conduction. Thermal desktop offers a series of options with the three most prominent being: area conductance, edge conductance, and absolute conductance [1]. Area conductance considers the contact area between two surfaces on a particular object. Edge conductance is similar to area conductance and is used to determine the contact between the edges of 2D surfaces. Absolute conductance takes a user specified conductance value and spreads it evenly over the pair of surfaces or objects that are in contact. Determining how well two surfaces are in contact is often quite difficult because of the unknowns described in 2.7.2 namely; surface roughness, material properties, and the contact method (press fit, bolted, glued etc.). For this reason ATMT uses the absolute conductance method which minimizes the uncertainty and reduces the computational burden on the user as well as the program.

The conduction algorithm is used to autogenerate the Conductance Matrix eliminating the need to manually define the contacts between surfaces or objects as is required by Thermal Desktop and SatTherm. The Conductance Matrix identifies which surfaces or objects are in contact and applies a specified absolute conductance (W/K) to the contact point. All surfaces are represented by a single node located at its centroid and for this reason a surface does not conduct to itself. This allows the main diagonal of the Conductance Matrix to be all zeros. The Conductance Matrix must remain symmetric indicating that a contact between a pair of surface or objects acts equally in both directions. This ensures that heat is conducted from high to low temperatures according to the laws of thermodynamics. A sample Conductance Matrix for a 2D six sided box is shown in Figure 3-10.



(a) 2D Six Sided Box

Conductance Matrix						
	+Z Panel - surf: 1	-Z Panel - surf: 1	+X Panel - surf: 1	+Y Panel - surf: 1	-X Panel - surf: 1	-Y Panel - surf: 1
+Z Panel - surf: 1	0	0	1	1	1	1
-Z Panel - surf: 1	0	0	1	1	1	1
+X Panel - surf: 1	1	1	0	1	0	1
+Y Panel - surf: 1	1	1	1	0	1	0
-X Panel - surf: 1	1	1	0	1	0	1
-Y Panel - surf: 1	1	1	1	0	1	0

(b) Sample Conductance Matrix

Figure 3-10: Sample Conductance Matrix Generated by the Conductance Algorithm

This sample Conductance Matrix is for a 1x1x1 m box with a separation tolerance of 0.05 m and absolute conductance of 1 W/K for all contacts. The separation tolerance is

used to determine if two surfaces are close enough to be in contact. If the surfaces are separated by a distance that is less than or equal to the separation tolerance, they will be in contact. Given the dimensions of the box, the separation tolerance, and the conductance matrix rules discussed above, it should be apparent that each surface must contact exactly 4 other surfaces.

The conduction algorithm is based on the principles discussed thus far. The main function is a search algorithm that loops through each vertex on each surface to determine if a vertex on a neighboring surface is within the separation tolerance. Figure 3-11 will be used to show how the conductance algorithm is applied to the 2D box example.

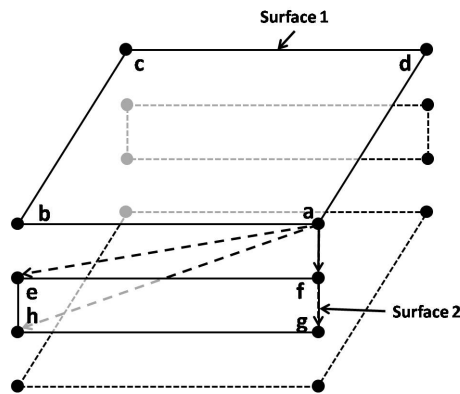


Figure 3-11: Conductance Between Surfaces

The three objects that will be used in the algorithm are the polygon, separation tolerance, and the absolute conductance.

1. **Polygon** - A polygon S is defined by a set of P coplanar points where each vertex has coordinates, $V_p = (x_p, y_p, z_p)$.
2. **Separation Tolerance** - The separation tolerance ST (m) is used to determine if two surfaces are in contact. A smaller separation tolerance is more restrictive and may reduce the number of contacts while a larger value would lead to more conductive couplings between objects.
3. **Absolute Conductance** - The absolute conductance G (W/K) is a coefficient used to determine the strength of the contact between two surfaces. A lower value would

be indicative of poor contact due to either a bad connection or to thermal isolation while a larger value may be used to represent strong thermal coupling.

The conductance algorithm for the 2D box example is described in some detail below with the `Conductance.m` code available in Appendix C. The objective is to determine which surfaces should be in contact based on a user defined separation tolerance.

1. Using surface 1 as the reference surface and V_a as the reference point, find the length $L_{\overline{ae}}$ of line \overline{ae} using Equation 3.7.

$$L_{\overline{ab}} = \left| \sqrt{(x_e - x_a)^2 + (y_e - y_a)^2 + (z_e - z_a)^2} \right| \quad (3.7)$$

2. Create a contact flag $CF = 0$ for the pair of vertices (ae) and check if $\overline{ae} \leq ST$. If so, the surfaces are in contact and $CF = 1$, else loop through all combinations (\overline{af} , \overline{ag} , \overline{ah}) exiting the loop if and when $CF = 1$. If $CF = 0$ at the end of the loop, use the next point on surface 1 (V_b) as the reference point and repeat the process for all points on surface 2.
 - If $CF = 0$ after both loops are completed, then the surfaces are not in contact and $G_{12} = 0$.
 - Else if $CF = 1$, then $G_{12} = G_{21} = G$.
3. Repeat steps 1, 2, and 3 for each surface until the Conductance Matrix is fully populated.

3.2 Thermal Model Validation Techniques

The two validation techniques used in ATMA are model-to-model and model-data correlation. Each technique provides insight into ATMT's accuracy and usability. Model-to-model correlation is done by comparing the performance of ATMT to an industry standard tool. Chapter 4 describes in detail the test cases that were run in ATMT and validated using Thermal Desktop (TD). The TD prediction models are treated as the "truth" source for the

ATMT validation. It is assumed that the model results provided by TD are perfectly accurate. In reality, there are multiple ways to construct the same model in TD each of which produce slightly different thermal responses. The TD validation models were developed to match the ATMT models as accurately as possible. However, there are subtle differences between the prediction models that make it challenging to provide an accurate comparison.

The second validation technique used in ATMA is model-data correlation. Model-data correlation compares the performance of a prediction model to actual test data. The governing rule of model-data correlation is to start simple and build-up in complexity. Beginning with a simple test case minimizes the number of unknown parameters which often lead to model error. Modeling a simple test case also allows the thermal engineer to analyze how different parameters affect the model results. Model-data correlation is done by properly varying parameters until the results are within the desired accuracy range. Determining which parameters can and most importantly should be varied is the most challenging part of model-data correlation.

For a simple test case, most of the model parameters are known values that should not be altered. Parameters such as the model dimensions, material properties, or ambient temperature are generally known values that can be held constant within the model. Some model parameters are flexible or unknown such as the contact strength between two objects or the emissivity of an object's surface. These are the parameters that should be varied in order to match the model to the data. There are several ways to vary the flexible parameters. The ATMA approach is to develop a baseline model incorporating all of the known and unknown parameters. Then each unknown parameter is modified in turn by a percent (usually $\pm 10\%$) of the baseline value and the results are documented. In this manner the thermal engineer will see the affect each parameter has on the thermal model. While the process is slow, it will only need to be performed once on the simple test case and may then be applied with confidence to increasingly complex models.

The ATMA model-data correlation process is demonstrated in Section 3.2.1 and Section 3.2.2 at the component and assembly levels.

3.2.1 Component Level Testing

A simple example of the ATMA model-data correlation process was done at the component level. The objective of the component test was to model the transient temperature response of a $6 \times 3 \times .25$ inch 6061 T6 aluminum block with a 22 Watt heat input. Figure 3-12 shows the test setup and corresponding TD model.

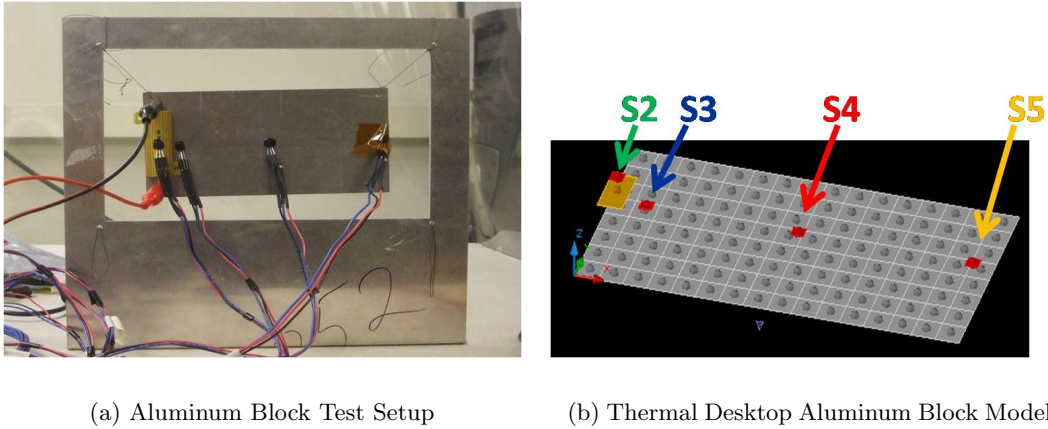


Figure 3-12: Aluminum Block Test Setup and Thermal Desktop Model

Four LM19 thermal sensors ($S2 \rightarrow S5$) and a 50 ohm resistor were attached to the aluminum block using Loctite thermal adhesive. A one hour test was performed at room temperature (~ 295 K) in a class 100,000 clean room environment. Figure 3-13 shows the baseline model-data comparison for sensors 3 and 4.

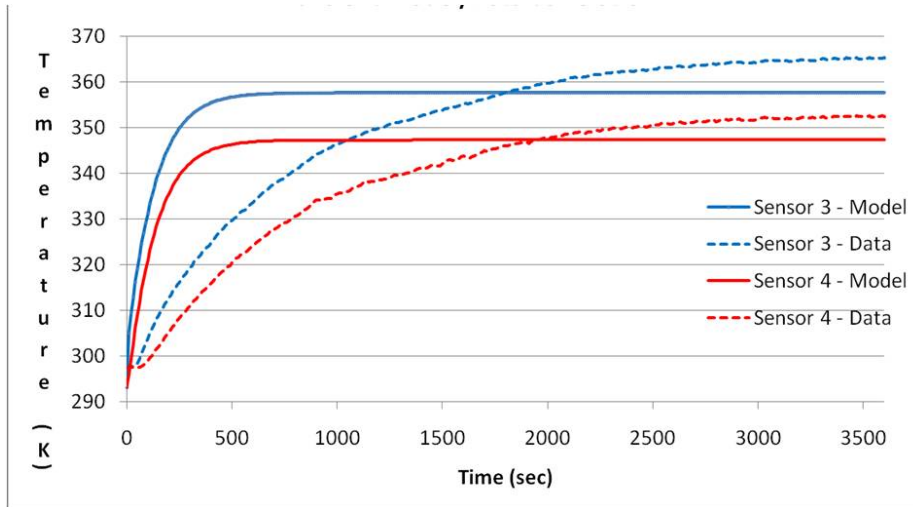


Figure 3-13: Baseline Model-Data Correlation

Clearly the baseline TD model does not match the component test data. The most noticeable discrepancies are in the time constants and the steady state temperatures. The TD model's time constant is ~ 6.7 times faster than the actual data while the steady state temperature averages 6.5 K cooler. Table A.3 of Appendix A shows a summary of the baseline model-data correlation results.

The next step was to determine the sources of error and to vary the appropriate parameters in order to correlate the model to the data. Three sources of error were identified and corrected within the model.

Error Sources

1. The thermal resistance of the plastic LM19 thermal sensors was not accounted for in the initial model. The sensors have a very low conductivity (0.2 W/mK) which significantly delayed the response time of the system. The sensors were added to the model and the temperatures were sampled at the sensor nodes.
2. Contact bonds between the sensors and the aluminum surface were assumed to be perfect. A resistance was added to the model to account for the thermal adhesive and surface properties.
3. Ambient temperature was assumed constant throughout the simulation. The actual ambient environmental temperatures were added to the model.

These modifications significantly improved the performance of the model but were insufficient to complete the model-data correlation. At this point several parameters were varied individually from the baseline design. The conductance between the resistor and sensors to the aluminum block was reduced from 5 W/K to 1 W/K to reflect the loose bond provided by the adhesive and the surface finishes for the copper resistor and aluminum block was altered to increase the emissivity to absorptivity ratio. Figure 3-14 shows the final model-data correlation. The model's new time constant is only ~ 1.1 times faster than the data with a steady state temperature averaging only 1.5 K cooler. Table A.4 of Appendix A shows a summary of the final model-data correlation results.

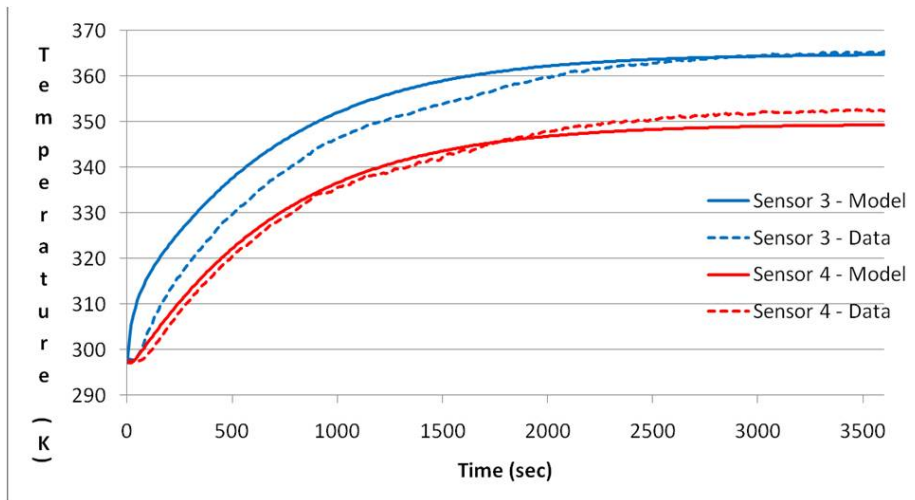
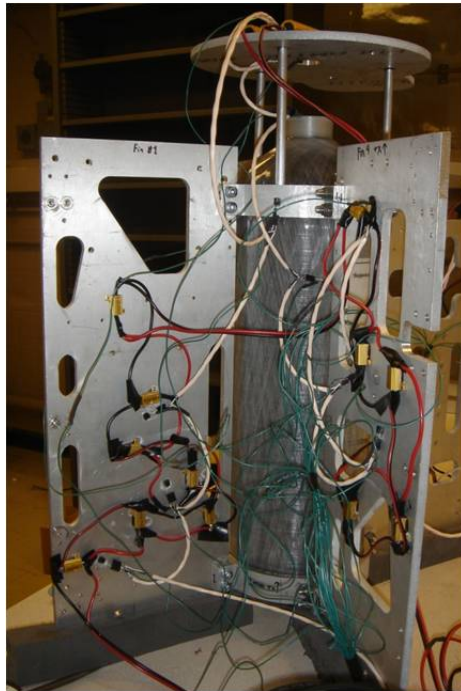


Figure 3-14: Final Model-Data Correlation

3.2.2 CASTOR Thermal Vacuum Test

The CASTOR thermal vacuum (TVac) test is a good example of model-data correlation at the assembly level. In the spring of 2009, a structural engineering model (SEM) of the CASTOR satellite was brought to MIT Lincoln Laboratory for vibration and TVac testing. The SEM consisted of the central xenon tank, tank clamps, four radiator fins, and the Divergent Cusped Field Thruster (DCFT) mounting plate. A series of 24 resistors were added to the structure to simulate the heat load for each power producing component. Table A-1 in Appendix A shows a complete list of the resistors and their locations on the SEM. Additionally, 16 LM19 thermal sensors and 6 T type thermocouples were distributed

across the satellite to monitor temperatures during testing [19, 13]. Figure 3-15 shows the CASTOR SEM and TVac test setup.



(a) CASTOR Structural Engineering Model



(b) TVac Chamber and Test Setup

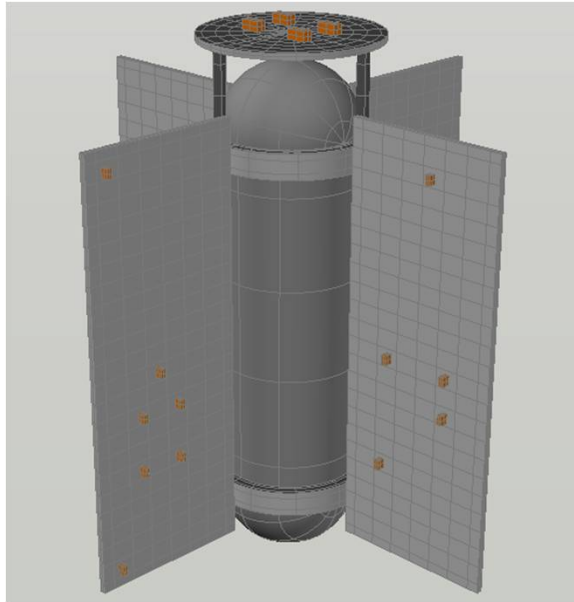
Figure 3-15: CASTOR Structural Engineering Model and TVac Test Setup

The objective of the TVac test was to demonstrate CASTOR thermal model-data correlation; proving that the TD thermal model could be used to accurately predict the thermal response of the structure under a given set of thermal loading parameters. Three tests were performed over an 11 hour time period. The tests are briefly summarized below.

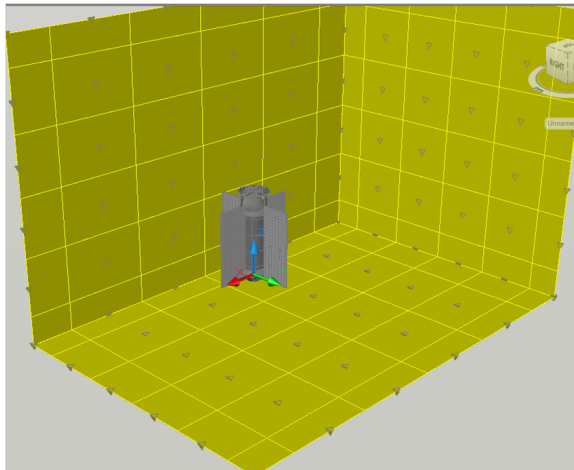
1. **Test 1** - From $T = 0$ to $T = 1200$ sec, the TVac chamber is brought down to vacuum (10^{-8} Torr) while the engine resistors output 180 W and the bus resistors output 84.5 W. At $T = 1200$ sec, the liquid nitrogen shroud begins to cool the TVac chamber from 298 K to ~ 83 K. The engine and bus power remain constant and the test runs until temperatures stabilize.
2. **Test 2** - The TVac chamber is held at ~ 83 K. Engine power is reduced to 0 W, bus power remains constant, and the temperatures are allowed to stabilize.

3. **Test 3** - The TVac chamber is held at ~ 83 K. Both the engine and bus power levels are brought to 0 W and temperatures are allowed to stabilize.

A TD model of the CASTOR TVac test is shown in Figure 3-16. The TVac test model incorporates both the CASTOR structure and the TVac chamber. Modeling the transient temperature of the TVac chamber walls was especially important. Prior to the TVac test, 7 thermocouples were placed inside the TVac chamber to monitor the thermal environment for use in the TVac model. Figure B-1 of Appendix B shows the chamber temperatures over the course of the 11 hour TVac test.



(a) Thermal Desktop Model of the CASTOR Structure



(b) Thermal Desktop Model of the TVac Chamber

Figure 3-16: Thermal Desktop Model of CASTOR Structure and TVac Test Setup

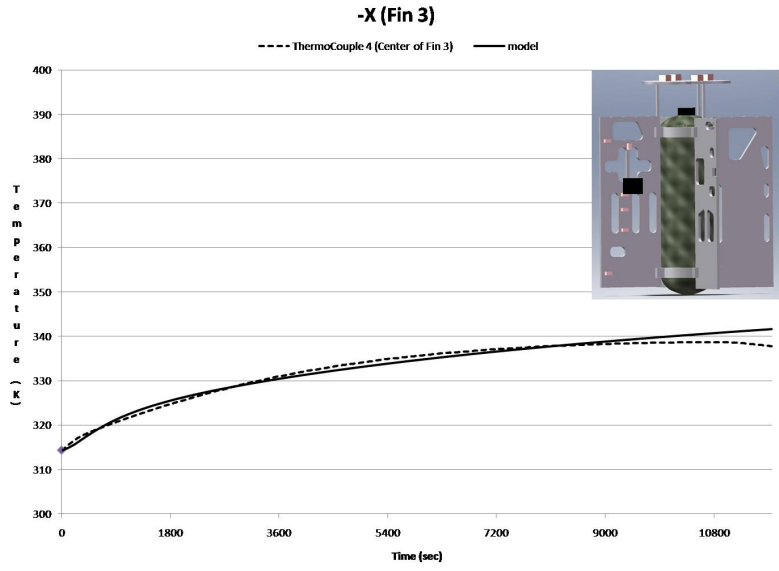
The TD model was used to simulate the post-depressurization segment of Test 1 from $T = 1200$ until temperatures first reached equilibrium at $T = 13,290$ seconds. Unfortunately the LM19 thermal sensors were unable to monitor the full temperature range of the test so the model was only correlated to the 6 thermocouples that were mounted to the structure. Figure 3-17 shows the model-data correlation for the -X and +X thermocouples. The model-data correlation for the xenon tank and +Y thermocouples is shown in Figures B-2 and B-3 of Appendix B.

The model-data correlation for the -X and +X fins show different performance trends. The -X fin model closely approximates the thermocouple data varying at most by ± 3 K while the +X fin model varies by as much as 7 K from the data. Furthermore, the +X model's time constant is significantly slower than the data and does not reach steady state by the end of the simulation. Many model parameters were investigated and tuned in an attempt to improve the model-data correlation. However, there are numerous unknowns and potential error sources that limited the accuracy of the model. The CASTOR TVac test clearly demonstrates the complexity and ambiguity of the model-data correlation process.

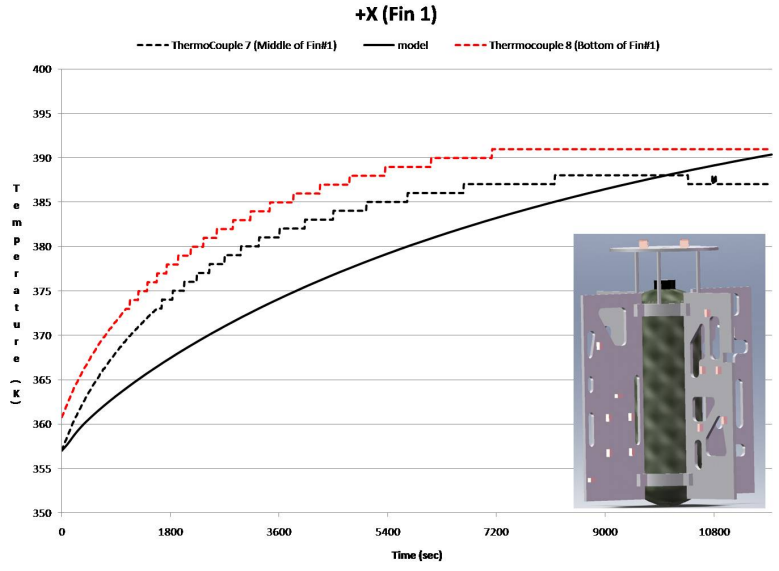
3.2.3 Summary

Validating a thermal model is a challenging endeavor that requires time and experience. Industry standard tools undergo years of testing and validation before they are brought to market and for this reason they come at a significant price. Even still, many commercially available tools remind the user that the model results are not always guaranteed and should be correlated to test data.

The ATMA model-to-model and model-data correlation techniques are two approaches that may be used for model validation. The ATMA model-data correlation examples showed that model accuracy goes inversely with model complexity. This inverse relationship is also the case for the ATMT model-to-model correlation examples presented in Chapter 4. While accuracy is obviously important, the most important takeaway from any validation is an understanding of the degree of accuracy that may be expected for a given model. Understanding a model's accuracy allows the user to build margin into the design ensuring that a system will meet its requirements.



(a) Model-Data Correlation for the -X Fin



(b) Model-Data Correlation for the +X Fin

Figure 3-17: TVac Test Model-Data Correlation

Chapter 4

Adaptive Thermal Modeling Tool (ATMT)

The Adaptive Thermal Modeling Tool (ATMT) is a MATLAB-based thermal modeling program tailored to the preliminary design phases of a smallsat program. ATMT was developed at the Massachusetts Institute of Technology Space Systems Laboratory by 2LT John Richmond with the support of the U.S. Air Force Academy and the University of Colorado Colorado Springs. The ATMT program is the central focus of the Adaptive Thermal Modeling Architecture (ATMA) and of this thesis. Designed to bridge the gap between the industry standard thermal modeling tools and the satellite specific thermal algorithms, ATMT attempts to combine the positive features of each in order to meet the thermal modeling requirements of a smallsat program.

Several tasks were required in order to meet this goal. First, the ATMT program needed to achieve a balance between simplicity and functionality. To accomplish this task a set of simplifying assumptions were used were applied to the ATMT design in order to reduce the math to the basic thermal processes required for on-orbit thermal modeling. The assumptions described in Section 4.1 form the scope of the ATMT model. Similar to any thermal modeling program, ATMT requires several initial inputs from the user. Section 4.2 identifies the inputs required to build a thermal model and to create an on-orbit simulation. The results of the thermal model described in Section 4.3 include all of the

outputs a thermal engineer should need to properly evaluate a satellite's thermal control system (TCS). Several examples of the ATMT's functionality are presented in Section 4.4 along with a Thermal Desktop validation in Section 4.5. Finally, the ATMT performance parameters (accuracy and usability) are evaluated and summarized in Section 4.6.

4.1 Modeling Assumptions

The thermal processes and algorithms presented in Chapters 2 and 3 rely on a set of simplifying assumptions. Many of these assumptions are implicit but others require a more formal introduction. Simplifying assumptions are used to solve a variety of problems that would otherwise lack a closed-form solution or would require unnecessary computational power. The fourth-order radiation heat transfer term is one common example where a simplifying assumption is used in a thermal model. Assuming that the fourth order component can be reduced to a first-order radiation resistance term, similar to the conduction resistance in Section 2.5, allows a system of higher order non-linear equations to be reduced and solved using linear approximations.

A second example is the assumption that all surfaces are smooth and flat. This assumption eliminates the need to model the radiation being absorbed and emitted from irregular geometries. In reality, microscopic surface imperfections such as waviness or roughness reduce the contact area between two surfaces. The effect of poor contact is often difficult to gauge and can lead to modeling errors in the contact resistance used for conduction calculations. Reductions in the model error due to unknown contact resistance can be achieved by using an interstitial fluid such as a thermal adhesive to bond the surfaces and fill the gaps. The thermal resistance of the adhesive allows the surfaces to be modeled as smooth flat planes with known contact properties [20]. These two examples along with the following list of key assumptions from Section 2.7.2, should be understood and considered when developing a model or performing any thermal analysis using the ATMT program.

Key Assumptions [20]:

- **Diffuse-gray Surfaces** - The spectral emissivity (ϵ), absorptivity (α), and reflectivity (ρ) are independent of wavelength and direction and are solely a function of

temperature such that $\epsilon_{(T)} = \alpha_{(T)} = 1 - \rho_{(T)}$.

- The solar and IR absorptivities of a surface are not equal ($\alpha_{solar} \neq \alpha_{IR}$), however the IR emissivity and absorptivity are equal ($\epsilon_{IR} = \alpha_{IR}$).
- **Transparent Space** - The medium between physical surfaces in space is perfectly transparent and has no effect on the radiation interchange between surfaces.
- **Opaque Surfaces** - The radiation heat transfer to and from a physical body occur only at the surface and not within the body.
- **Smooth Planar Surfaces** - Allows heat conduction to occur only through the contact surface area and radiation view factors may be calculated using Equation 2.19.
- **Isothermal Surfaces** - Individual surfaces (represented by a single node at the centroid) are considered uniform in temperature allowing the intensity, direction, and wavelength of the emitted energy to remain uniform over the surface area.
 - **Constant Conductivity** - The conductivity (k) of a material remains constant in the IR temperature range.
 - **Constant Emissivity and Absorptivity** - The emissivity (ϵ) and absorptivity (α) of a surface finish remain constant in the IR temperature range.

4.2 Thermal Model Inputs

One of the most useful features of ATMT is the implementation of a semi-autonomous architecture. This implies is that the user is removed from having to manually calculate the more time consuming and ambiguous parameters such as the VFs and the conductances between objects. At the same time, ATMT offers the flexibility to manipulate certain parameters in order to customize a model and simulation. Some user inputs are required to create a model, determine the material properties, and define an on-orbit simulation. This section will briefly introduce the required user inputs. A complete ATMT User's Manual is presented in Chapter 5.

The first step in creating a new model is to configure the geometric orientation, select the material properties, and initialize various parameters required for thermal analysis. The ATMT Geometry Tab shown in Figure 4-1 has 15 user-defined input categories. A short description of each input is provided below with greater detail available in Section 5.3 of the ATMT User’s Manual.

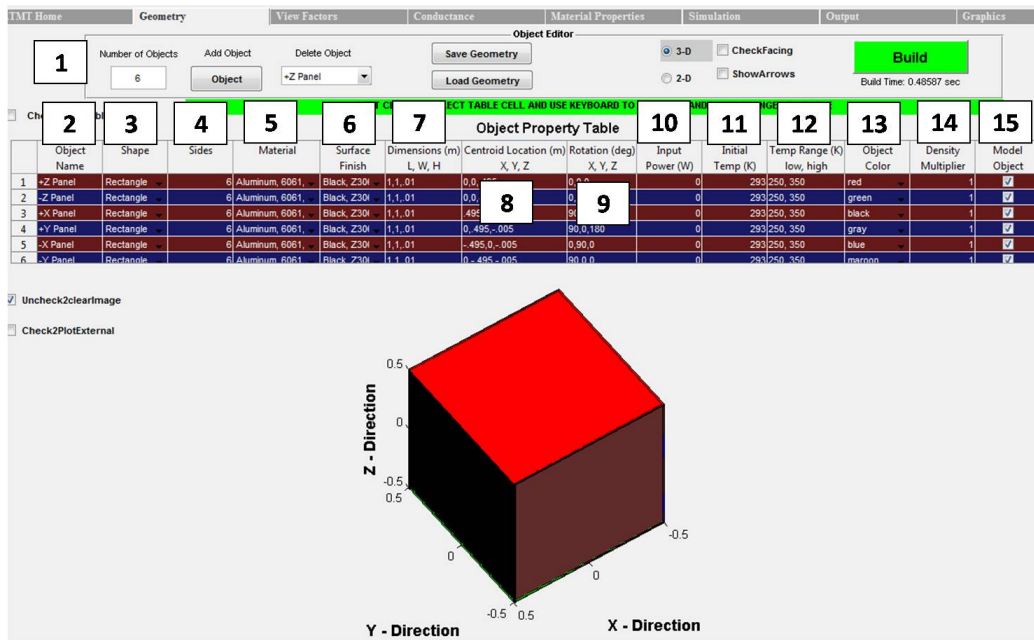


Figure 4-1: ATMT Geometry Tab: 15 User Defined Input Categories

1. **Number of Objects** - The ‘Number of Objects’ is used initially to add a desired number of components to the model. A minimum of 1 object is required with a maximum value limited only by MATLAB memory storage. ATMT is designed for smallsat applications and it is recommended that the user keep the number of components under 50.
2. **Object Name** - An ‘Object Name’ is written using any combination of letters, numbers, and keyboard symbols. For example, ‘+Z panel’ or ‘Obj. 1’ would be acceptable inputs.
3. **Shape** - The ‘Shape’ of an object may be either a rectangle or a cylinder.
4. **Sides** - A 3D rectangle (box) has a default of 6 sides while a cylinder may have

(5 → 362) sides, where two of the total number of sides are the top and bottom of the cylinder. For 2D objects, both the rectangle and cylinder are represented with one side.

5. **Material** - The “Material” option allows the user to apply a set of predefined material thermal properties to each object. The thermal properties include: density (kg/m^3), specific heat (J/kgK), and conductivity (W/mK). The material property options are discussed in Section 2.7 and are cataloged on the Material Properties Tab (See Section 5.6).
6. **Surface Finish** - The “Surface Finish” option allows the user to apply a set of predefined surface finishes to each object. The surface finish properties include solar absorptivity (α) and IR emissivity (ϵ) which are both measured on a scale of (0 → 1). The surface finish options are discussed in Section 2.7.2 and are also cataloged on the Materials Tab (See Section 5.6).
7. **Dimensions** - The “Dimensions” option allows the user to specify each object’s length (L), width (W), and height (H) in meters (m) using comma separated values (i.e. 1,1,1). For a rectangle, the length, width, and height correspond to the body X, Y, and Z axes. For a 2D surface rectangle, the height dimension is still required as it is used in the calculation of the object’s capacitance. For a cylinder, L corresponds to the radius (X and Y directions), W is a placeholder (N/A), and H corresponds to the cylinder’s height in the Z direction. The N/A dimension is required and can be any number (i.e. 1). The default dimensions for new components are (1, 1, 0.01 m).
8. **Centroid Location** - Defining the “Centroid Location” allows the user to place the object in 3D space. The centroid is the point at the dimensional center of the object (not the same as the centroid of the model). Like the dimensions parameter, the centroid location (m) requires an X, Y, and Z value separated by commas (i.e. 1, 1, 1 m) from the origin.
9. **Rotation** - Any object can be rotated in 3D space in order to place it in a desired orientation. Rotations (deg) are entered about the X, Y, and Z axes (i.e. 90°, 0°, 0°).

The default rotation vector for new components is $(0^\circ, 0^\circ, 0^\circ)$ which aligns the +Z surface along the +Z direction axis of the model.

10. **Input Power** - The “Input Power” (W) allows the user to apply an internal heat load to any component. The heat load is dissipated evenly throughout an object according to the conduction heat transfer processes. The default input power for new components is 0 W.
11. **Initial Temperature** - The “Initial Temperature” (K) parameter requires the user to specify the initial temperature state of each object. The default initial temperature for new components is 293 K.
12. **Temperature Range** - An object’s “Temperature Range” (K) is a function of the manufacturer’s specifications. Either the operational or survival temperature range is input as the lower and upper bounds using the comma separated value technique. The default temperature range for new components is (250 K, 350 K).
13. **Object Color** - The “Object Color” parameter is used solely for visualization purposes and can be selected from the drop down list similar to the way a material property was selected. The default color for new components is gray.
14. **Density Multiplier** - The “Density Multiplier” feature allows the user to model heterogeneous or discontinuous surfaces that may have more or less active surface area than a solid surface. For instance, a honeycomb solar panel may use a carbon face sheet adhered to aluminum honeycomb. The honeycomb has less active surface area and mass than a solid aluminum sheet of the same dimensions. The density multiplier is a unitless input with a range of $(0 \rightarrow 1)$. The default value for new components is 1 indicating a solid sheet.
15. **Model Object** - The last column of the Object Property Table allows the user to determine whether or not to model a specific component. If the “Model Object” box is checked, the component will appear in the geometrical configuration and will be incorporated into the simulation analysis. However, if the checkbox is unchecked, the object will not appear and will not be included in the analysis.

The final set of user-defined inputs creates the on-orbit scenario for thermal analysis. The ATMT Simulation Tab shown in Figure 4-2 contains six panels. All inputs are assigned a default value and can be modified by the user. A short description of each input is provided below with greater detail available in Section 5.7 of the ATMT User’s Manual.



Figure 4-2: ATMT Scenario Tab: 24 User Defined Input Categories

1. **Altitude** - The “Altitude” (km) is the maximum distance of the satellite from the Earth’s surface. The ATMT model requires the altitude to be greater than 100 km¹. The model does not independently verify that the satellite remains above the Earth’s surface so the user must select an altitude and eccentricity that ensure the satellite remains in space.
2. **Eccentricity** - The “Eccentricity” of the orbit (ϵ) is unitless and can take any value on a scale from ($0 \leq \epsilon < 1$) with 0 denoting a circular orbit and all values less than 1 indicating an eccentric orbit.
3. **Inclination** - The “Inclination” (i) is the angle (deg) from the orbital plane to the body’s equatorial plane. The inclination may range from ($0^\circ \rightarrow 180^\circ$) with ($0^\circ < i \leq$

¹The ATMT model uses 100 km as the reference distance for the edge of Earth’s atmosphere and the beginning of space

90°) indicating a prograde orbit and ($90^\circ > i < 180^\circ$) indicating a retrograde orbit.

4. **Argument of Perigee** - The “Argument of Perigee” (ω) is the angle (deg) from the ascending node to perigee measured in the orbital directions. The argument of perigee may range from ($0^\circ \rightarrow 360^\circ$).
5. **Right Ascension of the Ascending Node** - The “Right Ascension of the Ascending Node” (RAAN or Ω) is the angle (deg) from the principal direction (I) to the direction of the ascending node in the reference plane. RAAN may range from ($0^\circ \rightarrow 360^\circ$).
6. **True Anomaly** - The “True Anomaly” (ν) is the angle (deg) from perigee to the position of the satellite measured in the direction of motion. The true anomaly may range from ($0^\circ \rightarrow 360^\circ$).
7. **Year** - The start and finish year must be input as an integer between 1900 and 2100 AD².
8. **Month** - The start and finish month must be input as an integer between 1 and 12.
9. **Day** - The start and finish day must be input as an integer between 1 and 31³.
10. **Hour** - The start and finish hour must be input as an integer between 0 and 23.
11. **Minute** - The start and finish minute must be input as an integer between 0 and 59.
12. **Second** - The start and finish second must be input as an integer between 0 and 59.999.
13. **Body** - The body is the central object about which the satellite orbits. The default is central body is Earth. Future versions of ATMT will include the Moon, Sun, and Mars.
14. **Radius** - The “Radius” (R) of the body is measured in kilometers (km). The default radius for Earth is $R_{Earth} = 6378.14$ km.

²The algorithm is good for dates between 1 Mar 1900 to 28 Feb 2100.

³Be aware not all months will allow a range of 1-31 days.

15. **Gravitational Parameter** - The “Gravitational Parameter” (μ) of the body is measured in (km^3/sec^2). The default gravitational parameter for Earth is $\mu_{Earth} = 398601 km^3/sec^2$.
16. **Temperature** - While the body temperature (K) will vary depending on the orbital parameters and seasonal conditions, ATMT uses a constant temperature approximation. The default temperature for Earth is $T_{Earth} = 250$ K.
17. **Albedo Factor** - The “Albedo Factor” (AF) is the fraction ($0 \rightarrow 1$) of incident solar energy that is reflected by the central body. The default albedo factor for Earth is $AF_{Earth} = 0.35$.
18. **Solar Flux** - The “Solar Flux” (G_S) is a measure of the incoming solar electromagnetic radiation per unit area (W/m^2). The default solar flux for Earth is $G_S = 1354 W/m^2$.
19. **Primary Orientation** - In order to determine the satellite’s position in inertial space, the user must choose a primary orientation for the satellite. The four basic orientations include: Sun-facing, nadir-facing, velocity-facing, and right ascension and declination (R.A. & Decl.).
20. **Body Axis** - The user must specify which body axis is to be aligned along the primary direction. The options are +X, +Y, and +Z. If +X is chosen, the +Z direction will be set as the second direction with +Y completing the right-handed coordinate system. If +Y or +Z is chosen, +X will take the second direction. For a Sun or nadir-facing orbit, the second direction is aligned in the orbital plane in the direction of velocity (only along the velocity vector for circular orbits). For a velocity-facing orientation, the second direction is set in the orbital plane in the nadir direction (only along the nadir vector for circular orbits). Finally, for a (R.A. & Decl.) specific orbit, the second direction is placed in the velocity direction.
21. **Right Ascension and Declination** - Right ascension and declination (R.A. & Decl.) relate the attitude of the satellite with respect to the Sun (deg). The R.A.&Decl. orientation is used only if it is selected as the primary orientation.

22. **Rotation Axis** - The “Rotation Axis” can be selected if the user wishes to impart an on-orbit rotation on the satellite. The options are again +X, +Y, and +Z.
23. **Rotation Rates** - The “Rotation Rate” (deg/orbit) allows the orbit to maintain a constant angular velocity throughout a simulation.
24. **Step Size** - The “Step Size” (Δt) is the time segment (seconds) used to propagate the simulation forward in time. A range of (1 → 1200 sec) has been tested but it is recommended that the user stay in the range of (100 → 500 sec)

4.3 Thermal Model Outputs

The thermal model outputs are the end goal and final product of an ATMT simulation. They allow the thermal engineer to analyze the satellite’s on-orbit thermal response and to make any required design changes necessary to maintain component temperatures within their required range. For many inexperienced thermal engineers it is often difficult to understand what data should be analyzed and how to identify possible problems in their design. This section identifies the model outputs, where they can be found, and a few of the trends that should be expected from the output data.

The ATMT outputs include: object and surface temperatures, external heat fluxes (due to the Sun, Earth Albedo and Earth IR), internal radiation, internal heat generation, and conduction. Additionally, an ATMT output file will store state data such as the position (\vec{R}_{SC}) and velocity (\vec{V}_{SC}) vectors through time as well as the relative orientations of all surfaces and objects in the IJK coordinate frame. Other time dependent parameters such as the Sun vector and stability factor are also stored for post-processing. The ATMT Output Tab shown in Figure 4-3 shows the 5 main thermal outputs with a sample temperature plot for the 2D box example described in Section 4.4.1. The other model outputs are accessible from the command window. The ATMT model outputs will be discussed briefly in this section. A detailed description of the ATMT Output Tab’s functionality is available in Section 5.8.

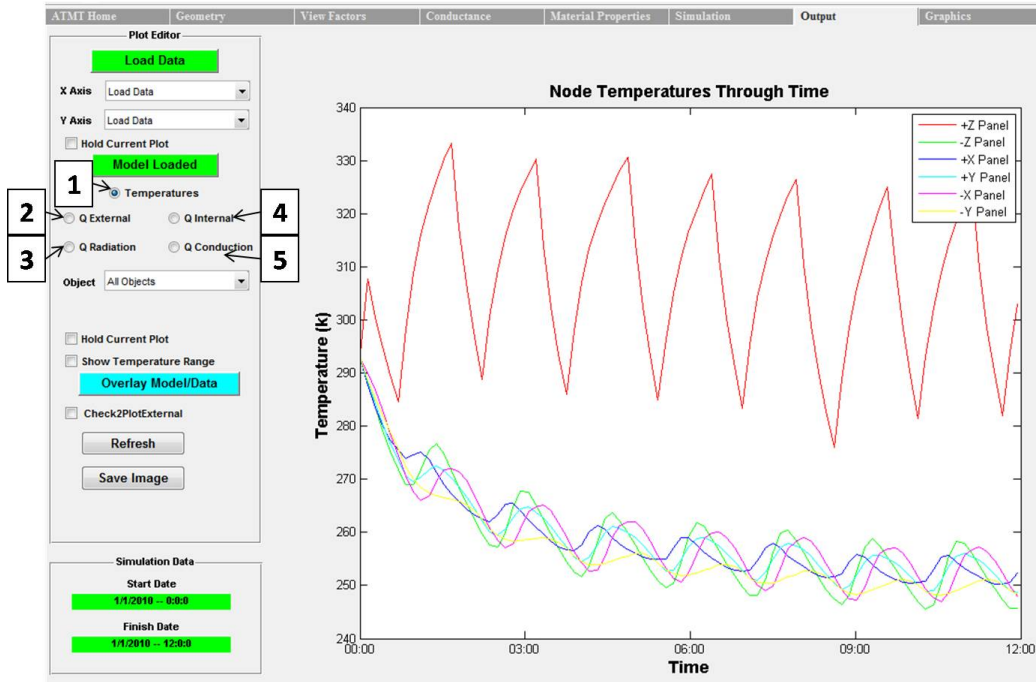


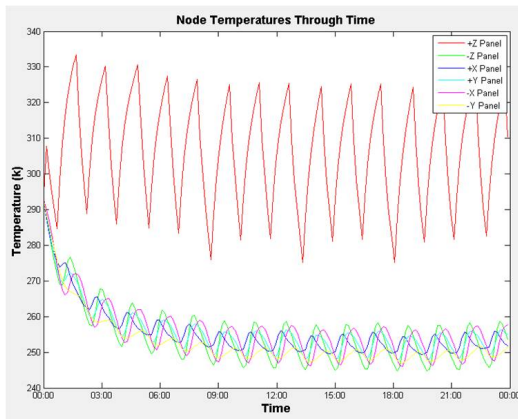
Figure 4-3: ATMT Output Tab: Thermal Model Outputs and Sample Results

1. **Temperatures** - The “Temperature” output plots the transient thermal response for the 2D box on the Y axis vs. the simulation runtime on the X axis. The MATLAB plot will automatically scale the X axis to display the time using the most appropriate units. Figure 4-13 shows an example of the on-orbit temperature response of the 2D six sided box example. Some of the key features are the amplitude and periodicity of each component’s temperature response, the thermal coupling between the non Sun-facing components, and the stabilization of the temperatures over time.
2. **External Heat Fluxes** - The “External Heat Fluxes” (EHFs) include the influence of the Sun, Earth Albedo, and Earth IR. Similar to the temperature output, the transient EHF’s are plotted on the Y axis vs. the simulation runtime on the X axis. The EHF (W/m^2) output allows the user to investigate the major sources of heat on a particular component. Figure 4-4(b) shows an example of the EHF’s on the +Z (Sun-facing) panel of the 2D box. The key feature of this plot is the relative impact of each heat source on the total heat flux.
3. **Radiation** - The “Radiation” output is the net internal radiation (W) for a particular

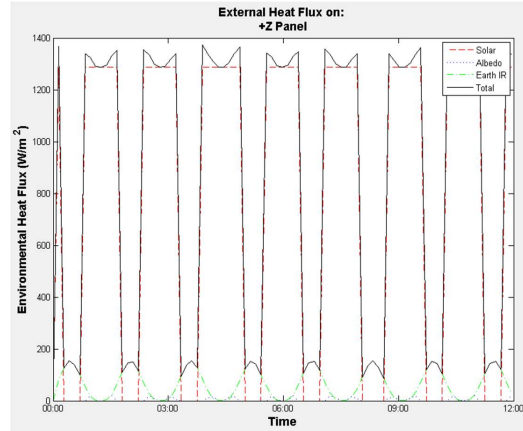
surface or object. At any given time step, the net radiation could be positive, indicating that the neighboring components are relatively warmer, or negative indicating the opposite. Figure 4-4(c) shows an example of the net radiation on each of the six side panels. The key feature is the quicky stabilization of the internal radiation which is due to the fact that the internal radiation is roughly 1/3 the magnitude of the total external flux.

4. **Internal Heat Input** - The “Internal Heat” output is simply a measure of the component operating power (W) that is converted to heat through inefficiencies. The magnitude of the internal heat comes from the input power described in Section 4.2. Currently this model output serves as a truth mechanism ensuring that the model did indeed apply the desired input power to each component. Future versions of ATMT will have the ability to model a time-dependent input power allowing the user to simulate the actual power states of the component⁴.
5. **Conduction** - Similar to the radiation output, “Conduction” is actually the net conduction (W) to and from a particular surface or object. The net conduction may be either positive indicating a gain of heat from warmer objects, or negative indicating a loss. As is evident in Figure 4-4(d), the conduction plot follows the same general trends as the radiation plot under normal circumstances.

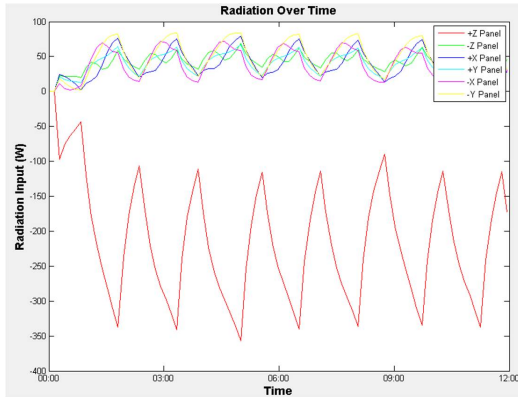
⁴The power states will vary by component and may include an off, idle, operational, and peak power level.



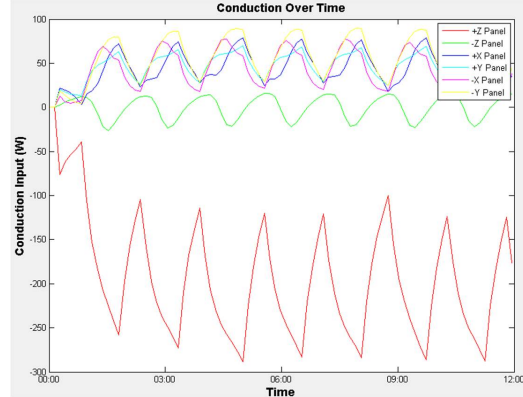
(a) 2D Box On-orbit Temperature Response



(b) External Heat Fluxes on the +Z (Sun-facing) Panel



(c) Net Radiation for 2D Box Surfaces



(d) Net Conduction for 2D Box Surfaces

Figure 4-4: ATMT Model Results for the 2D Six Sided Box Example

The secondary model outputs introduced in this section are not readily available in the ATMT Output Tab but are stored in the internal memory and simulation run-file for post-processing. The state data, orientation, Sun vector, and stability factor are some of the more frequently used model outputs that give the user the flexibility to validate simulation parameters with other software such as Thermal Desktop (TD) or Satellite Tool Kit (STK). A brief description of each secondary output is discussed below.

- **State Data** - The state data includes the IJK position (\vec{R}_{SC} km) and velocity (\vec{V}_{SC} km/s) vectors of the satellite throughout the simulation runtime. The \vec{R}_{SC} and \vec{V}_{SC} are both $N \times 3$ matrices, where N is the number of time steps and the three columns

are the X, Y, and Z values. These matrices allow the user to replicate and visualize the orbit in either TD or STK in order to check parameters such as the Sun/eclipse cycle or the β angle of the orbit.

- **Object Orientation** - An object's orientation is rotated from its individual body frame to the IJK frame at each time step. The main function of this rotation is to accurately account for the EHF's which are dependent on the angle of a surface with respect to a heat source.
- **Sun Vector** - The Sun vector (\vec{S} km) is measured from the spacecraft to the Sun in the IJK coordinate frame. The two main purposes of the Sun vector are to determine the fraction of solar energy incident on each surface and to determine whether or not the spacecraft is in sunlight or eclipse.
- **Stability Factor** - The stability factor is used to determine if the time step (Δt) is too large for a particular simulation which may lead to aliasing and in turn unstable temperature growth. The mitigation is to calculate the minimum value of Equation 4.1 for each finite-difference node or surface in the model [7].

$$\tau_n = \frac{C_n}{\sum_j G_{nj}} \quad (4.1)$$

where τ (seconds) is the stability factor, G_{nj} (W/K) is the sum of all conductances (radiation or conduction) to node n by all surrounding nodes j , and C_n (J/K) is the thermal capacitance of node n . The stability factor represents the minimum time constant of all nodes in the network at any point in time. If the stability factor is greater than the Δt value from the Simulation Tab at any point in time, Δt will be reduced by 10% each step until it meets the criteria.

4.4 Model Functionality

The functionality of ATMT is specifically geared toward the actual thermal modeling requirements and tolerances that a smallsat program will encounter throughout its early

design phases. Beginning as early as the mission statement, thermal constraints are placed on the design of the thermal control system (TCS). The mission objectives and Concept of Operations (CONOPs) will dictate the environment that the spacecraft will experience while limiting the hardware available to perform the mission. Depending on the satellite's mission, components may have requirements on thermal cycling and thermal shock or may even be required to operate at extreme hot or cold temperatures.

The early design phases (conceptualization through CDR) require a variety of thermal modeling trade studies be performed in order to design an effective TCS. Trade studies are performed at the component, assembly, and most importantly vehicle levels to identify possible configurations and orbital scenarios that will meet the mission objectives. Performing trade studies requires the flexibility to quickly develop a model and orbital scenario that give “sufficiently accurate” and “usable” results. It is imperative that the user understand these two terms as they are the evaluation metrics for the ATMT program.

The discussion in Section 3.1 explained that ATMT is geared toward simplifying the design to a point where enough details are retained to provide the user with a reliable thermal model. What this translates to is reducing the number of objects in the model to just those that are the largest, most sensitive, or greatest power producing components. Components that comprise the primary structure, power system, propulsion system or payloads are the likely target areas. The temperature ranges for these components will indicate what is to be expected for objects mounted on or near these components. The term “sufficiently accurate” is used because the results are based on a simplified design. Section 2.2.1 explained that the level of uncertainty in a satellite's thermal performance remains even as the program matures through the build and test phases. For this reason David Gilmore, author of *Spacecraft Thermal Control Handbook*, argues that the key to a thermal model is understanding its accuracy and building margin into the TCS to account for the uncertainty. The term “usable” simply means that the user can make sense of the results and use them to make design changes in future simulations. Many thermal modeling programs offer a plethora of highly accurate results, but if the user does not know how to make sense of the data then they are useless.

This following section presents some of ATMT's numerous functions and shows how

they are used to address the early design phase thermal modeling requirements. While there are an infinite number of possible combinations, four baseline test cases are used to demonstrate several of the key trade studies an engineer will encounter in smallsat thermal modeling. Section 4.5 presents a model-to-model validation of the four baseline test cases using Thermal Desktop.

4.4.1 Test Case Overview

Developing and utilizing effective test cases allows the ATMT user to understand the strengths and weaknesses of the program. Unfortunately this is a more challenging task than one would expect due to the significant differences between ATMT and other programs such as SatTherm and Thermal Desktop. Like all thermal modeling programs, ATMT is based on the physics of heat transfer and is governed by the laws of thermodynamics. However, several theoretical techniques were developed to address the more ambiguous aspects of thermal modeling. For this reason it is critical that the test cases be representative of actual scenarios that may be correlated to other models or test data. The four test cases were chosen to span a range of increasingly complex geometric configurations and realistic orbital scenarios. A baseline configuration and on-orbit scenario is used in each test case with trade studies presented for a variety of parameters.

The material properties database is shown in Figure B-4 of Appendix B, provides the material and surface finish properties referenced in all of the test cases. The database may be extended to include any desired material or surface finish. The parameters used for the baseline on-orbit simulation are shown in Figure B-5 of Appendix A. All of the test cases will use a Keplerian orbit with a 500 km altitude at 45° inclination. The other orbital elements (ϵ , Ω , ω , ν) are initially zero. The simulation starts on 1 Jan 2010 at 00:00:00 and runs until to 2 Jan 2010 (24 hour runtime) at 00:00:00. The baseline case uses the default Earth parameters discussed in Section 4.2. The baseline model orientation is in the +Z Sun-facing direction such that the +Z face points perfectly at the Sun throughout its orbit. There are no rotations for the baseline case. Finally, a 500 second step size is used to for each simulation.

Test Case 1: 2D Surface

Test Case 1 is of a 2D surface with dimensions $1 \times 1 \times 0.01$ m in length, width, height respectively⁵. While the object is represented in 2D, the height dimension is required to calculate the object's capacitance. ATMT treats the 2D surface as a single thermal node located at the centroid. All heat transfer operations are performed on the node causing the surface to remain isothermal. Figure 4-5 shows the baseline properties for the 2D surface as well as a graphical display of the surface as it is seen in the body frame.

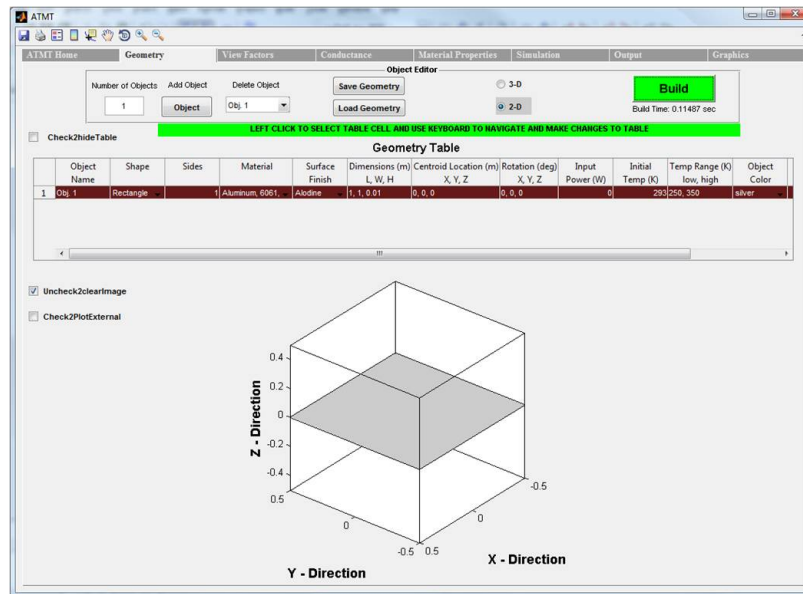
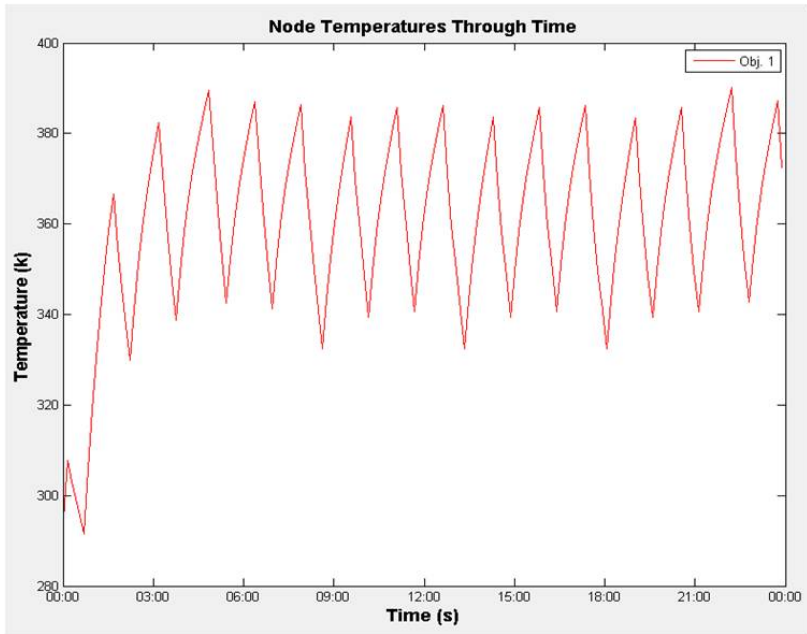


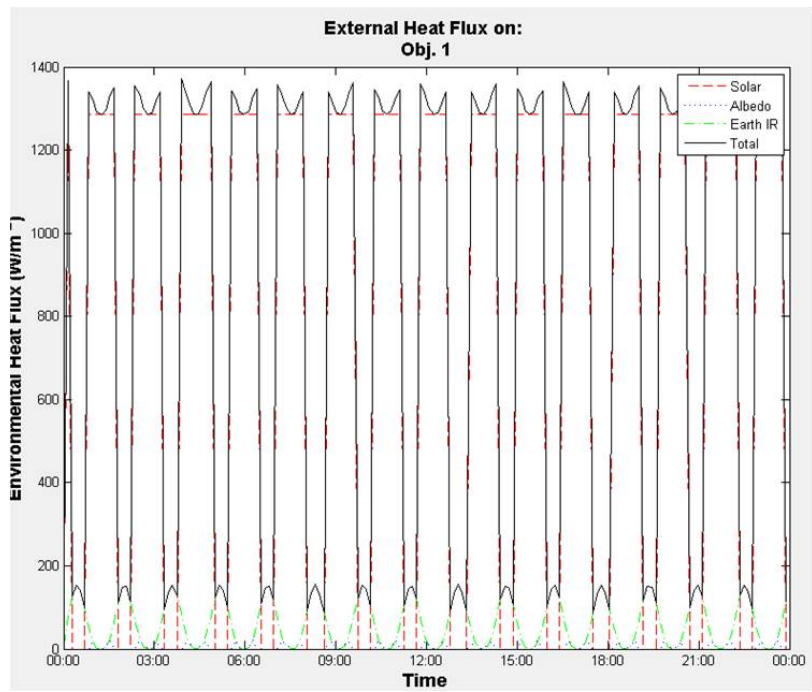
Figure 4-5: Test Case 1: 2D Surface Properties and Graphical Display

The baseline case uses Aluminum 6061 T6 as the material and 3 mil black Z306 paint as the surface finish. The object is located at the origin $(0,0,0)$ and is oriented with the height dimension in the $+Z$ direction (i.e. no rotations). There will be zero power input to the plate and the plate's initial temperature in space is 293 K with a temperature range of $250 \rightarrow 350$ K. The model is of a single planar surface represented by one node so there is no internal radiation or conduction (VF and Conductance Matrices are zero). Running the baseline simulation described in Section 4.4.1 produces the following temperature and EHF plots for the 2D surface.

⁵The length, width, and height dimensions correspond to X, Y, and Z axes where the height is always applied in the Z direction.



(a) Temperature through Time for a 2D Surface

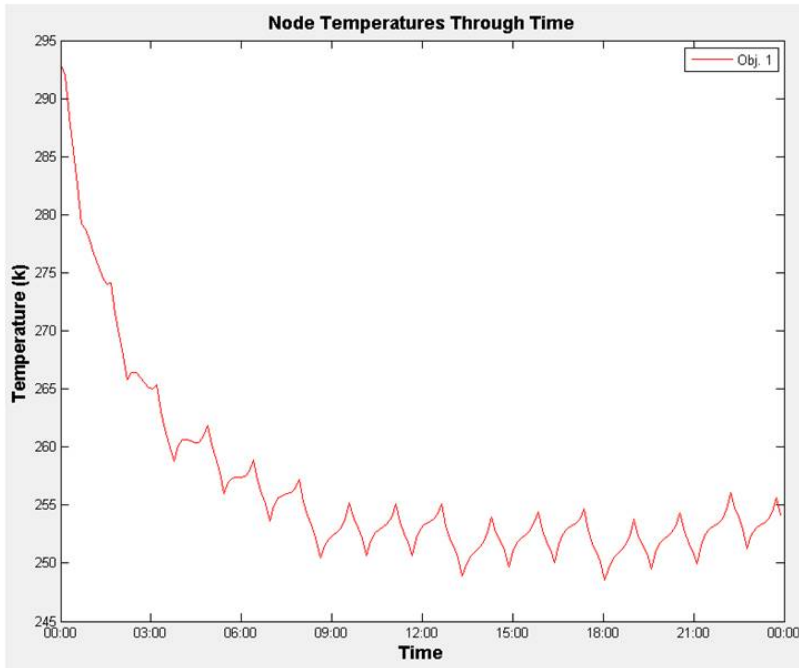


(b) External Heat Fluxes on a 2D Surface

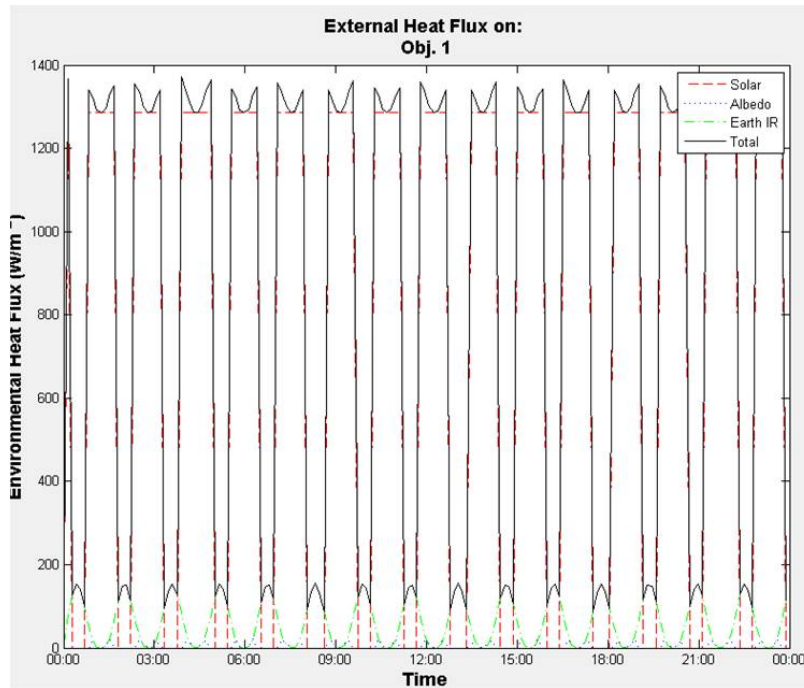
Figure 4-6: 2D Surface Temperature and External Heat Fluxes

As expected, the temperature of the surface increased from 293 K to an average of ~ 360 K. The temperature response can be attributed to two main factors: the black Z306 paint having an extremely high solar absorptivity and emissivity ($\alpha = 0.95$, $\epsilon = 0.87$) and the surface being oriented in the Sun-facing direction. Two trade studies will be performed on Test Case 1 by varying the surface finish and the height dimension.

1. The first trade study is to change the surface finish from black Z306 paint to Z93 white paint which has $\alpha = 0.17$ and $\epsilon = 0.92$. The EHF's should remain the same however; the surface should equilibrate about a considerably cooler temperature. As Figure 4-7 demonstrates, the new equilibrium temperature is ~ 252 K which confirms the expected result of using a highly emissive surface finish.
2. The second trade study for the 2D surface is to change the surface's height dimension to 1 m creating a solid $1 \times 1 \times 1$ m object. The increase in thickness should cause the model to take significantly longer to reach equilibrium. Additionally, the amplitude of each Sun/eclipse cycle will be heavily dampened. This is due to the increase in mass which makes it considerably more difficult to raise the node's temperature. Figure 4-8(a) confirms the first prediction and Figure 4-8(b) shows that it actually takes about 20 days on-orbit for an aluminum box of this size to reach equilibrium.

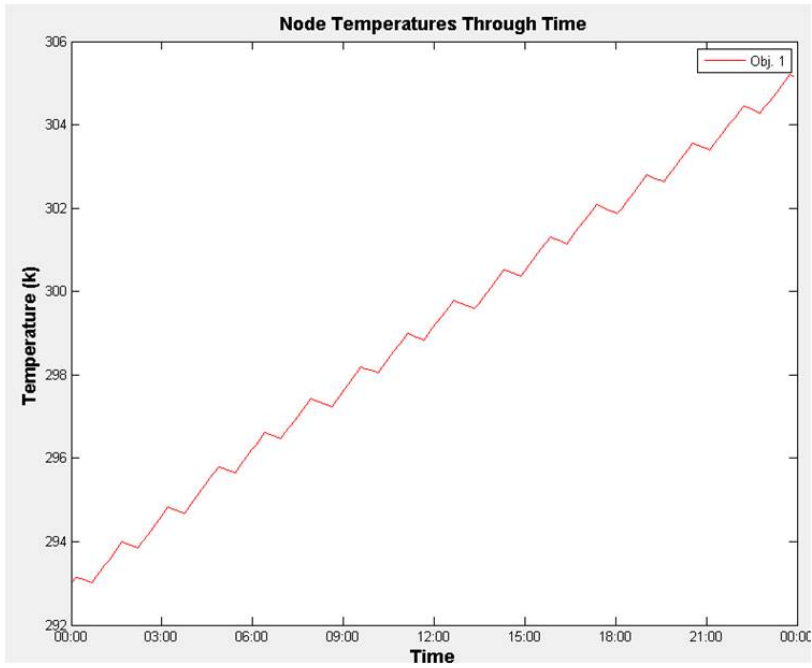


(a) Temperature through Time for a 2D Surface

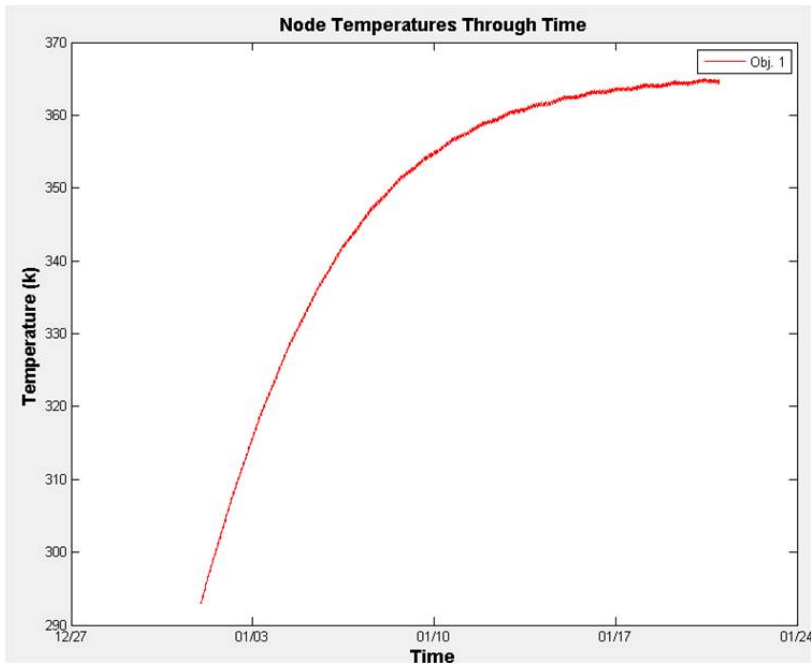


(b) External Heat Fluxes on a 2D Surface

Figure 4-7: Varying the Surface Finish for a 2D Surface



(a) Low Amplitude and Steady Rise Time for a 2D Surface



(b) Temperature Stabilization after 20 Days

Figure 4-8: Increasing the Thickness of a 2D Surface

Test Case 2: 3D Solid Rectangle

The model in Test Case 2 has the same dimensions as Test Case 1 except that it uses a 3D solid rectangle as opposed to a 2D surface. The main difference for 3D objects is the increase in the number of radiating and conducting nodes. ATMT treats 3D components as multi-surface objects applying a node to the centroid of each surface. The ATMT 3D surface-node approach requires significant adaptations from the 2D algorithms. Figure 4-9 shows that the geometric properties remain the same but the object is now modeled in 3D.

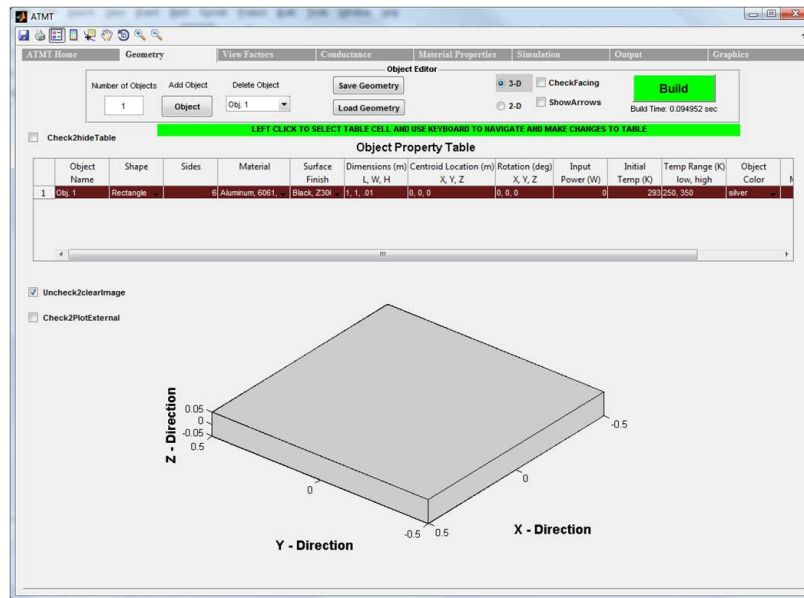


Figure 4-9: Test Case 2: 3D Solid Rectangle

While there is still only one object, there are now six actively radiating and conducting surfaces. The VF and Conductance Matrices are shown in Figure 4-10. The VF Matrix is all zeros because there are no VFs between the surfaces on a solid convex object. The Conductance Matrix shows that each surface on the 3D solid rectangle will conduct to one another.

	Obj. 1 - surf: 1	Obj. 1 - surf: 2	Obj. 1 - surf: 3	Obj. 1 - surf: 4	Obj. 1 - surf: 5	Obj. 1 - surf: 6
Obj. 1 - surf: 1	0	0	0	0	0	0
Obj. 1 - surf: 2	0	0	0	0	0	0
Obj. 1 - surf: 3	0	0	0	0	0	0
Obj. 1 - surf: 4	0	0	0	0	0	0
Obj. 1 - surf: 5	0	0	0	0	0	0
Obj. 1 - surf: 6	0	0	0	0	0	0

(a) 3D Solid Rectangle View Factor Matrix

	Obj. 1 - surf: 1	Obj. 1 - surf: 2	Obj. 1 - surf: 3	Obj. 1 - surf: 4	Obj. 1 - surf: 5	Obj. 1 - surf: 6
Obj. 1 - surf: 1	0	1	1	1	1	1
Obj. 1 - surf: 2	1	0	1	1	1	1
Obj. 1 - surf: 3	1	1	0	1	1	1
Obj. 1 - surf: 4	1	1	1	0	1	1
Obj. 1 - surf: 5	1	1	1	1	0	1
Obj. 1 - surf: 6	1	1	1	1	1	0

(b) 3D Solid Rectangle Conductance Matrix

Figure 4-10: 3D Solid Rectangle View Factor and Conductance Matrices

Test Case 3: 2D Six Surface Box

Test Case 3 simulates a 2D six surface box with geometrically equivalent surfaces to the single component modeled in Test Case 1. As in case 1, all of the components are modeled as 2D surfaces that still require the height dimension for thermal calculations. The major differences between Test Case 3 and cases 1 and 2 is the existence of non-zero VF and Conductance Matrices that relate the internal geometry and contacts between the surfaces. Figure 4-11 provides an overview of the geometric configuration and component properties. The model's centroid remains at 0, 0, 0 which requires that each surface's centroid be placed ± 0.495 m in the required direction. Notice that the centroid location takes into account half of the 0.01 m surface thickness. The +X, -X, +Y, and -Y surfaces all must be rotated by $\pm 90^\circ$ about either the X or Y axis respectively in order to place it in the desired vertical configuration. The 2D box configuration uses the baseline material properties, surface finishes, input power, and temperature ranges for all surfaces.

With a total of six surfaces in the model, both the VF and Conductance Matrices should now be 6x6 with zeros along the main diagonal to ensure that a surface does not radiate or

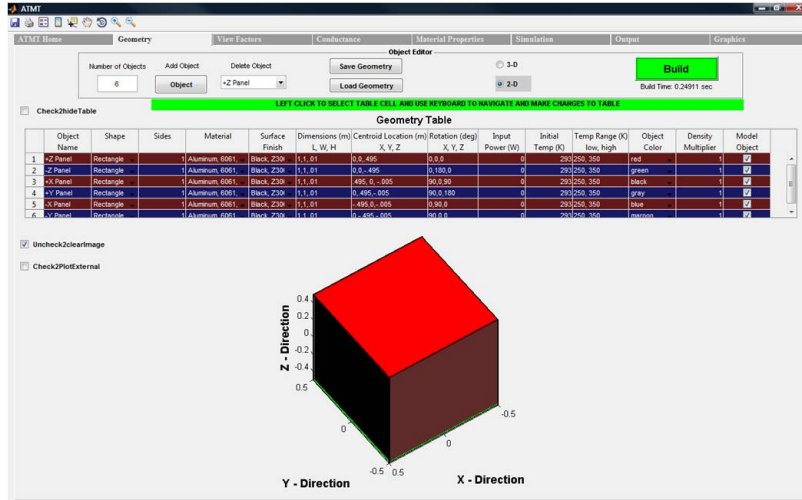


Figure 4-11: Test Case 3: 2D Six Surface Box

conduct to itself. The symmetry of the box configuration causes the VFs between surfaces to be nearly identical however, the number of significant figures used in the VF algorithm leads to the minor differences. Based on a separation tolerance of 50 mm or 0.050 m, each surface should only be in contact with four neighboring surfaces allowing the conductance between opposing surfaces to remain zero. Figure 4-12 displays these trends for the VF and Conductance Matrix that will be used in the thermal algorithm.

	+Z Panel - surf: 1	-Z Panel - surf: 1	+X Panel - surf: 1	+Y Panel - surf: 1	-X Panel - surf: 1	-Y Panel - surf: 1
+Z Panel - surf: 1	0	0.2025	0.1971	0.1971	0.1971	0.1971
-Z Panel - surf: 1	0.2025	0	0.1934	0.1934	0.1934	0.1934
+X Panel - surf: 1	0.1971	0.1934	0	0.1956	0.2025	0.1956
+Y Panel - surf: 1	0.1971	0.1934	0.1956	0	0.1956	0.2025
-X Panel - surf: 1	0.1971	0.1934	0.2025	0.1956	0	0.1956
-Y Panel - surf: 1	0.1971	0.1934	0.1956	0.2025	0.1956	0

(a) 2D Box View Factor Matrix

	+Z Panel - surf: 1	-Z Panel - surf: 1	+X Panel - surf: 1	+Y Panel - surf: 1	-X Panel - surf: 1	-Y Panel - surf: 1
+Z Panel - surf: 1	0	0	1	1	1	1
-Z Panel - surf: 1	0	0	1	1	1	1
+X Panel - surf: 1	1	1	0	1	0	1
+Y Panel - surf: 1	1	1	1	0	1	0
-X Panel - surf: 1	1	1	0	1	0	1
-Y Panel - surf: 1	1	1	1	0	1	0

(b) 2D Box Conductance Matrix

Figure 4-12: 2D Six Surface Box View Factor and Conductance Matrices

Running the baseline orbital scenario described in Section 4.4.1, the model should produce similar temperature results for the +Z Sun-facing surface with cooler equilibrium temperatures for the other surfaces due primarily to the fact that they receive no EHF's from the Sun. Figure 4-13 shows the temperature results for the baseline scenario. As expected, the +Z surface is significantly warmer than the other five surfaces although much cooler than the results from Test Cases 1 and 2 due to radiation and conduction to the other surfaces.

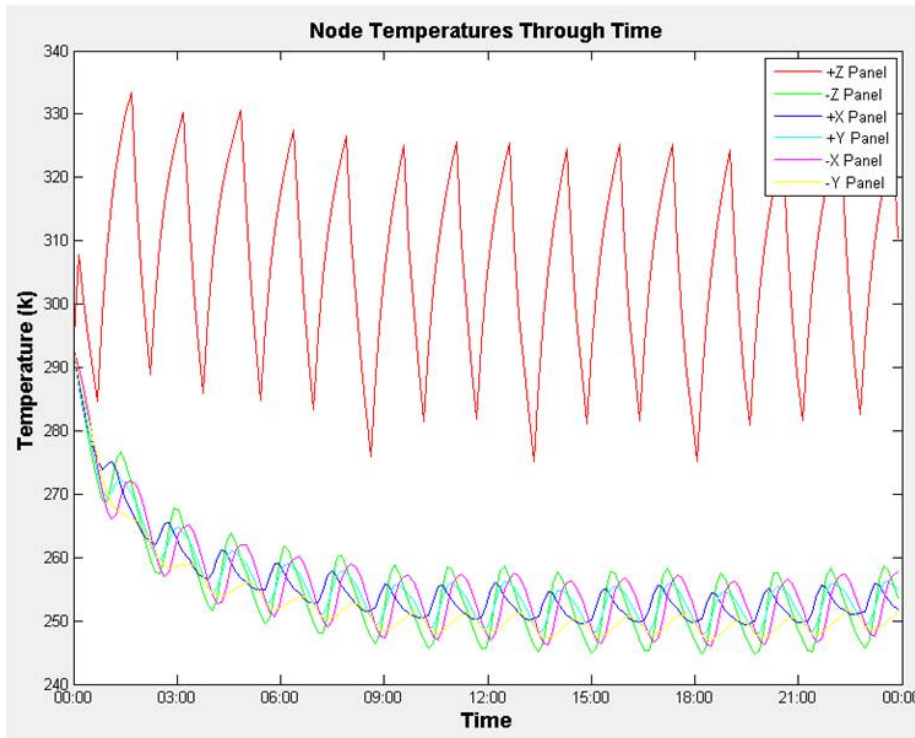
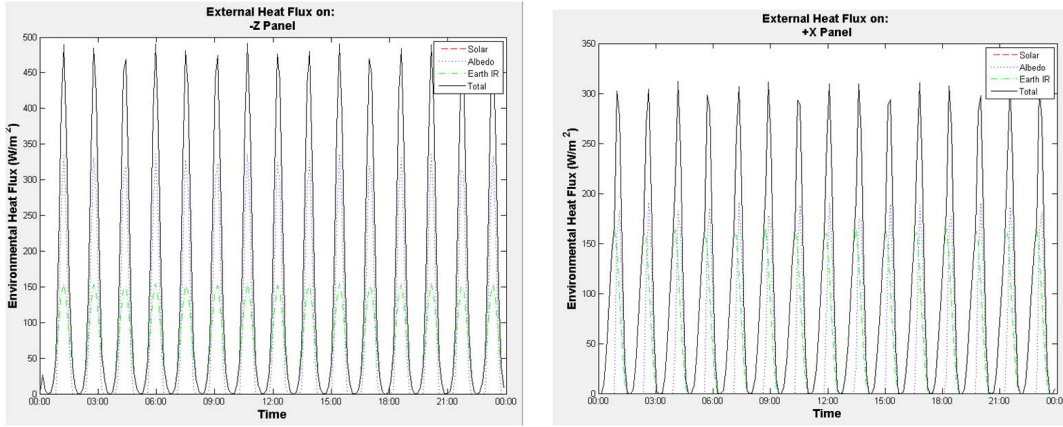


Figure 4-13: Temperatures Response for a 2D Six Surface Box

The EHF's on the +Z surface are identical to Test Cases 1 and 2 however each of the other five surfaces receive only Earth albedo and Earth IR. A comparison of the -Z and +X surfaces is shown in Figure 4-14.

Test case 3 presents a great platform for several of the most important trade studies. Three parameters will be varied in this section including the: conductance value, satellite orientation in space, and the reference orbit.

1. The first trade study for Test Case 3 is to change the conductance value from 1 to 5 W/K signifying an increase in conduction between surfaces. While 1 W/K was used



(a) External Heat Fluxes on the -Z Surface

(b) External Heat Fluxes on the +X Surface

Figure 4-14: Comparison of the External Heat Fluxes on the -Z and +X Surfaces

to represent a loose connection or poor bond between surfaces, 10 W/K could be used to simulate bolted components or thermal adhesive. Increasing the conductance value should cause the surface temperatures to become highly coupled. Even the +Z surface's equilibrium temperature will equilibrate closer to that of the other five surfaces. Figure 4-15 confirms this prediction by demonstrating how the +Z surface temperature drops because of the increase in conduction to the +X, -X, +Y, and -Y surfaces. Additionally, the equilibrium temperatures of the other five surfaces rise reflecting the net gain in heat due to conduction.

2. The second trade study will demonstrate the effect of changing the satellites on-orbit orientation from a Sun-facing orientation to having the +Z surface facing in the velocity direction. In order to lock the satellite into place, the +X axis will point toward the center of the Earth in the orbital plane. Maintaining a velocity-facing orientation should lead to a more even distribution of the temperatures due to the absorbance of solar energy by more than just the +Z surface. Figure 4-16 confirms the expected result and also shows that the overall average temperature is between that of the hot and cold surfaces from the baseline scenario.
3. The third trade study will be to alter the reference orbit. The baseline scenario placed the satellite in a 500 km 45° inclined circular orbit. For comparison, this trade

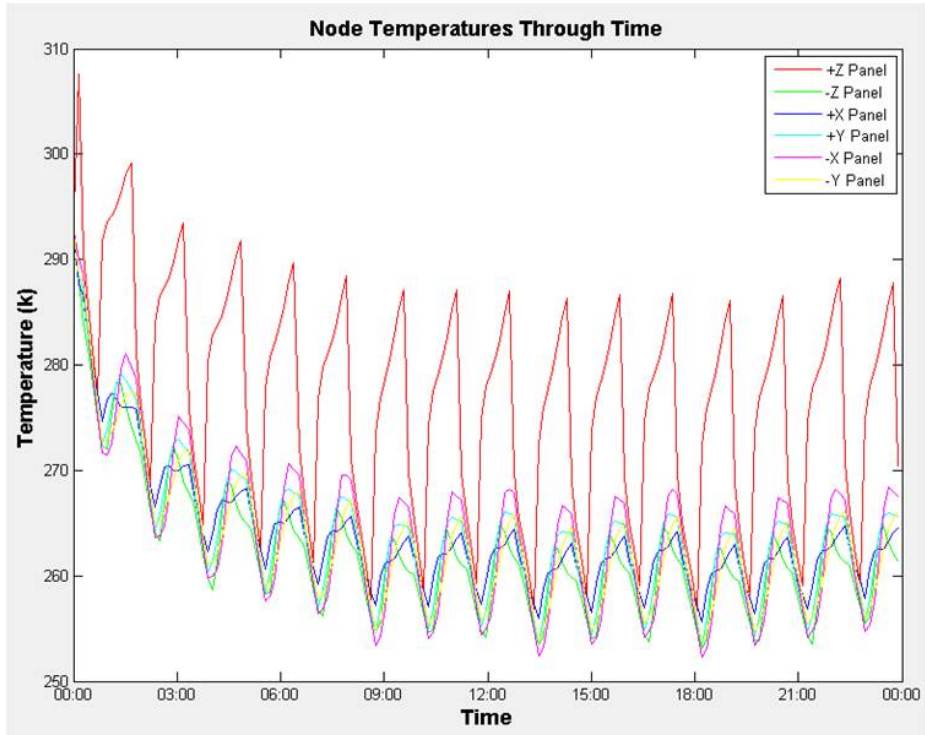


Figure 4-15: Increasing the Conductance Value Between Surfaces

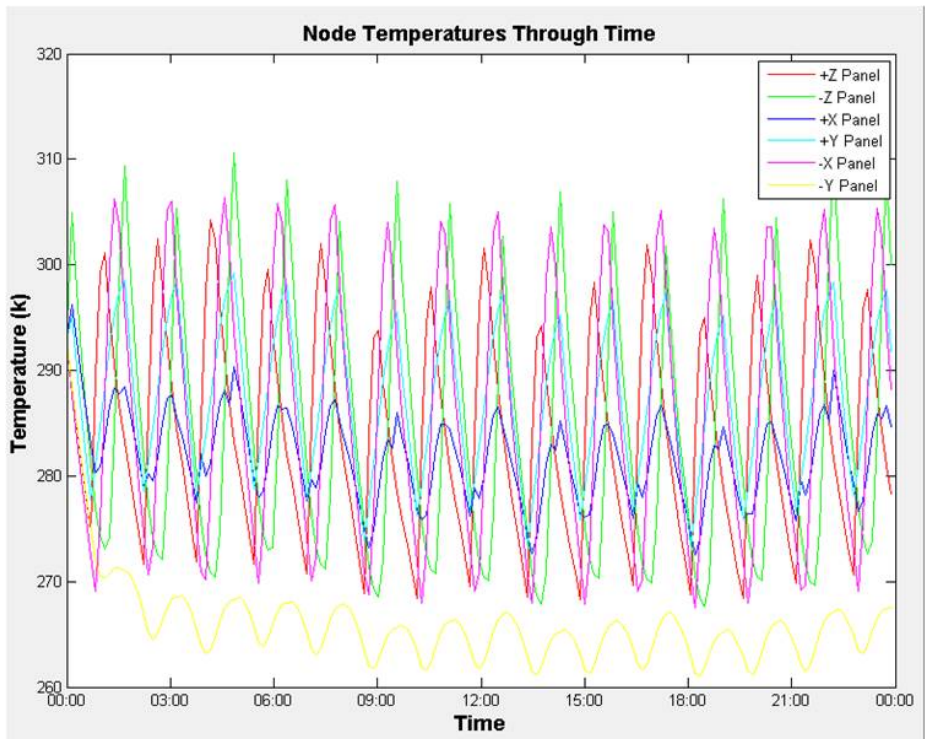


Figure 4-16: Temperature Response for a 2D Box with +Z in the Velocity Direction

study will place the satellite in a Molniya orbit using the following COEs: 26,571 km maximum altitude, $\epsilon = 0.7$, $i = 63.4^\circ$, $\Omega = 0^\circ$, $\omega = 270^\circ$, and $\nu = 0^\circ$ [18]. In the Molniya orbit the 2D box should remain in the Sun for the entire 24 hour period and should experience a smaller peak to peak amplitude range due to the reduction in Earth Albedo and Earth IR at the raised altitude. The temperature response shown in Figure 4-17 appears to be significantly different from the previous LEO test cases. The period increased to ~ 12 hours and surface temperatures respond differently with constant sunlight. The results still confirm the predictions and demonstrate that a change in the COEs will significantly impact an model's thermal response on-orbit.

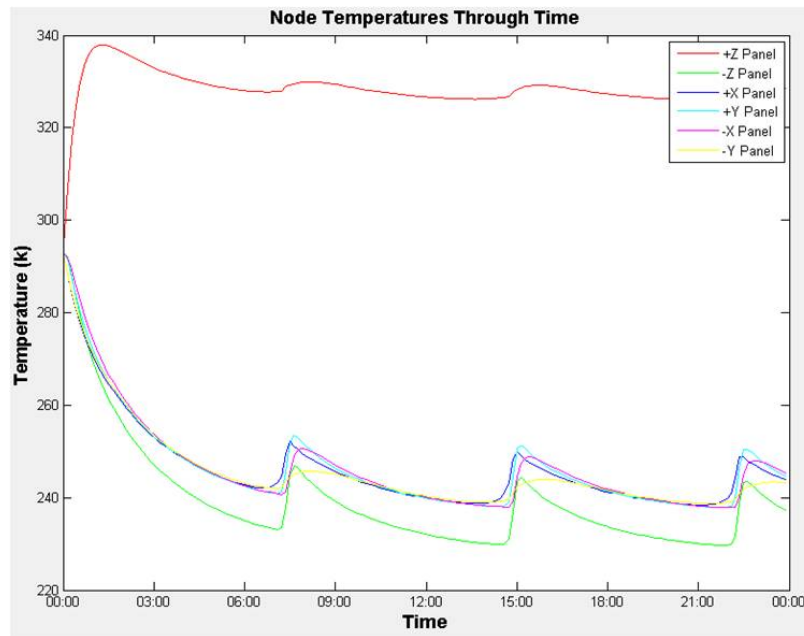


Figure 4-17: Temperature Response for a 2D Box in a Molniya Orbit

Test Case 4: 3D Six Panel Box with Internal Avionics Stack

The final test case (Test Case 4) is meant to be a simple but realistic 3D model of a flight-level satellite. Test Case 4 builds on the first three cases using the same dimensions as Test Case 1, the 3D nature of Test Case 2, and the box configuration from Test Case 3. Additionally, the model has a 3D avionics stack to simulate the effect of an internally mounted heat producing component. The avionics stack could include components such as the flight processor, communication modems, hardware control boards or even a battery

box. As Figure 4-18 shows, the avionics box is $0.3 \times 0.3 \times 0.4$ m in length, width, and height and is mounted to the $-Z$ panel of the box. The centroid is located at $0, 0, -0.995$ m which places the bottom ($-Z$ surface) of the avionics stack on the top ($+Z$ surface) of the $-Z$ panel.

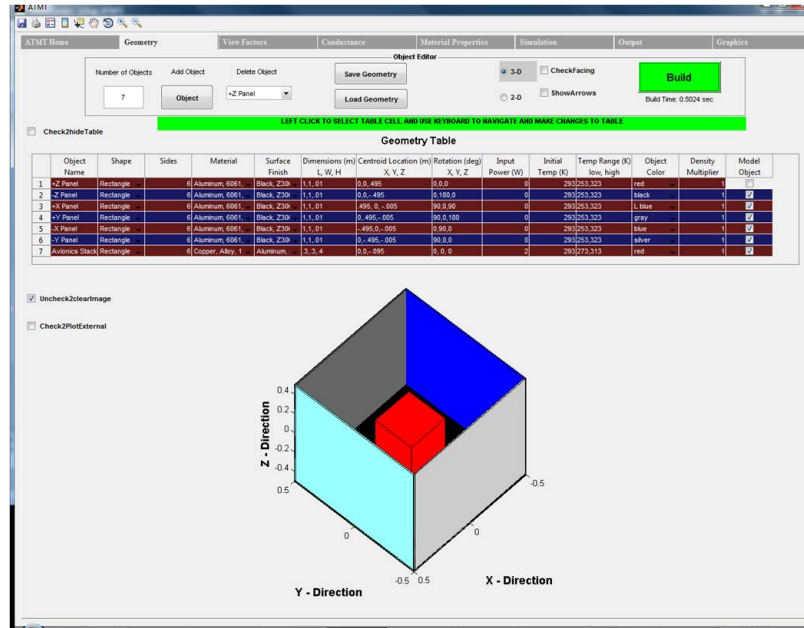


Figure 4-18: Test Case 4: 3D Six Panel Box with Internal Avionics Stack

The coordinate system and reference frame for the model are very important. The model as a whole has a body reference frame with its origin at $(0, 0, 0)$. The origin point is treated as the centroid of the vehicle for all calculations performed with respect to the IJK frame. All objects and surfaces have their own centroid and body frame separate from the body frame of the model. This allows each component to be rotated and positioned appropriately within the model. For some components it may be crucial to have the proper surface facing in the desired direction. For instance, a side panel may have solar cells on its exterior surface with a metallic or painted finish on the other five surfaces. ATMT allows the user to check that each object is rotated appropriately using the “check facing” feature that is described in Section 5.3. For consistency, the $+Z$ surface of each panel in Test Case 4 faces toward the space environment, while the $+Z$ surface of the avionics stack points in the $+Z$ direction of the model. Figure 5-7 shows how each component is rotated such that its $+Z$ surface faces in the desired direction.

In order to make Test Case 4 a realistic example of a satellite model, several additional

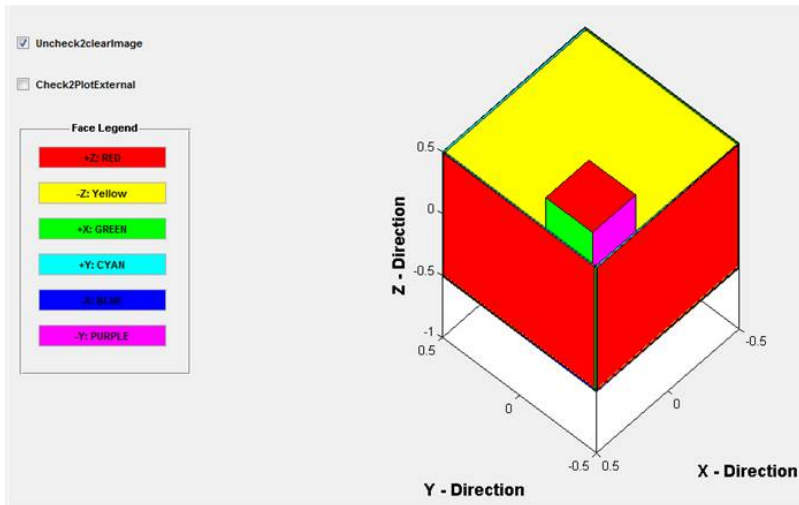


Figure 4-19: Example of Individual Component Orientation

parameters were modified from the baseline design. The avionics stack is made of a copper alloy (11000 series), has a polished aluminum surface finish, and has an input power of 20 W. The objective of Test Case 4 is to determine if the satellite model meets the component temperature constraints shown in Figure 4-18 given the baseline scenario described in Section 4.4.1. A conductance value of 5 W/K was used for the avionics stack to -Z panel to simulate a bolted or glued connection. The high mass-to-surface area ratio for the avionics stack will cause it to take significantly longer to reach equilibrium than the thin side panels so the runtime was extended to 48 hours. The 2D version of the model is run through the solver in order to minimize the runtime and provide accurate results. The 3D Temperature Averaging Algorithm used to represent a 3D object's temperature by a single node is currently under development as part of the future work in Section 6.3. Figure 4-20 shows the temperature response for the 3D box panels and the internal avionics stack.

While the temperature response of the six side panels closely mimics the results of Test Case 3, the avionics box has a far different response. Both the amplitude and equilibrium settling time are very different from those of the six side panels. The primary reason for the damped amplitude response is that the avionics box does not receive any EHF's from the Sun or the Earth. The Sun/eclipse cycle is barely evident as it is solely a function of the heat energy that is conducted from the -Z panel and radiated from the side panels. The longer settling time of the avionics stack is due to its thickness which limits the rate of heat

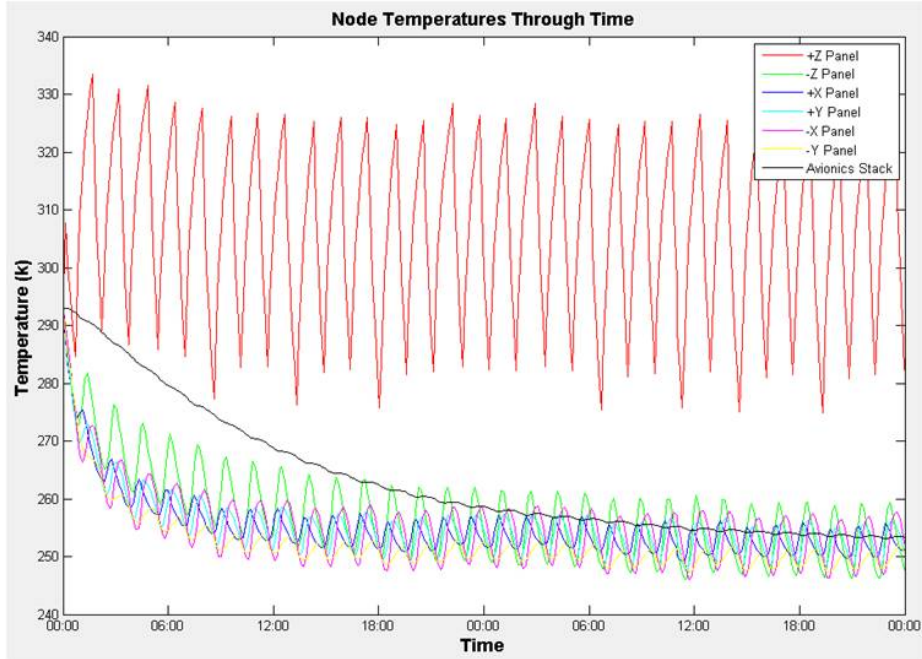


Figure 4-20: Temperature Response for a 3D Box with an Internal Avionics Stack

transfer through object. The -Z temperature is slightly higher on average than the other non Sun-facing panels due to conduction from the internally powered avionics stack. The graph alone does not answer the question of whether or not the avionics stack and the side panels remain within their temperature limits. A quick judgment can be made by adding the temperature limit lines at the values specified in Figure 4-21. The temperature limits and response are plotted for the avionics stack and +Z panel in Figure 4-21.

The temperatures of the +Z panel exceed its 323 K upper boundary while the temperatures of the avionics stack exceed its 273 K lower boundary. Given the current properties of the model as well as parameters used in the baseline scenario, both the +Z panel and avionics stack fail to meet their temperature requirements. The thermal engineer can use these results and the tools within ATMT to modify the model design until it satisfies the thermal requirements.

4.5 ATMT Validation

Validation is a challenging but invariably necessary step toward understanding ATMT's functional capabilities as well as its limitations. While the principles of heat transfer re-

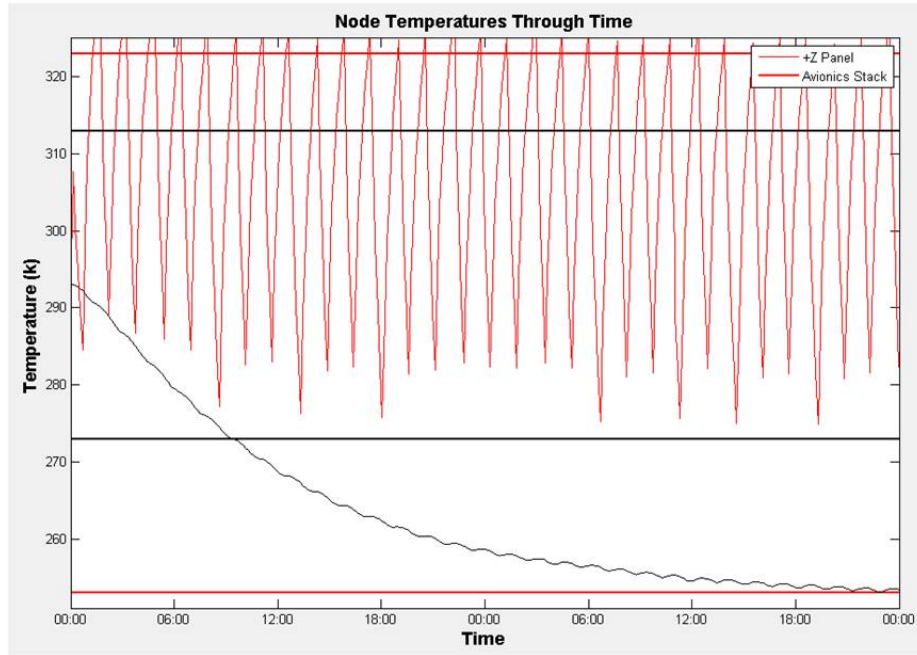


Figure 4-21: Temperature Limits for the +Z Panel and Avionics Stack

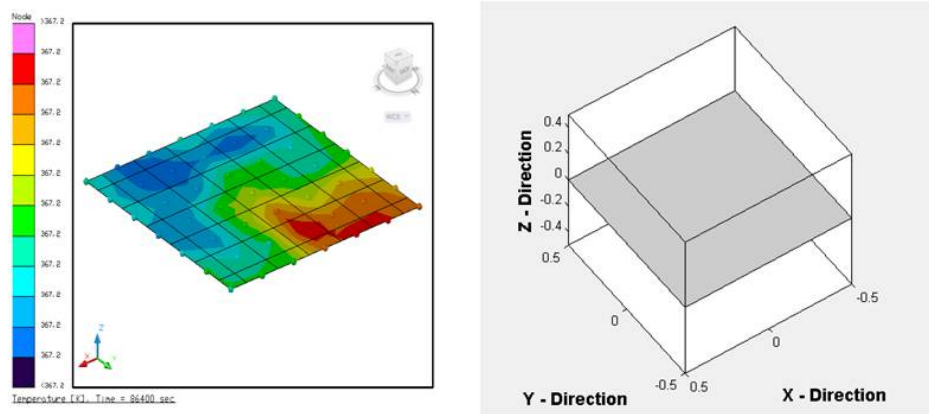
main the same for any thermal modeling program, the techniques and algorithms used to implement those principles can vary significantly. The background math provided in Chapter 2 was used to develop the theory and algorithms presented in Chapter 3, and the implementation and functionality of the ATMT program has been demonstrated up until this point in Chapter 4. However, without a source of comparison there is no telling whether the results attained from a simulation in ATMT are sufficiently accurate to make design decisions. Some error is to be expected between ATMT and other programs. The simplifying assumptions described in Section 4.1 and the unique algorithms used in the VF and conductance calculations are different from other thermal modeling programs and will contribute to some of the error. The goal is for ATMT's results to be sufficiently accurate when validated with a trusted model and especially when validated with thermal test data. However, the true objective of this section is not to prove that ATMT is extremely accurate, but to gain an understanding of how accurate ATMT is for increasingly complex models. Understanding the expected accuracy allows the thermal engineer to account for error in their design by building in margin where it is needed.

The four baseline test cases have been validated using Thermal Desktop as a truth

source. As discussed in Section 3.2, the difficulty of model-to-model validation is that there may be uncertainty in the truth source as well as in the actual model that is being validated. The Thermal Desktop validation models were created and executed by University of Colorado Colorado Springs (UCCS) senior Chris Flatley. Chris Flatley is currently the thermal modeling engineer for the U.S. Air Force Academy’s FalconSAT-5 program and is an expert with Thermal Desktop. Given the exact test cases and baseline scenario described in Section 4.4.1, Chris Flatley was able to develop and test equivalent models and simulations in Thermal Desktop. Sections 4.5.1 through 4.5.4 present the model-to-model comparison of the four baseline test cases.

4.5.1 Validation of Test Case 1: 2D Surface

Test Case 1 was modeled in Thermal Desktop (TD) exactly as it was described in Section 4.4.1. The only difference is that the TD surface has a 7×7 node mesh whereas the ATMT surface has a single node at the centroid. The TD image in Figure 4-22(a) shows the temperature gradient at the end of the 24 hour scenario. The on-orbit surface display is shown in Figure B-6 of Appendix B.



(a) 2D Surface Modeled in Thermal Desktop

(b) 2D Surface Modeled in ATMT

Figure 4-22: 2D Surface Modeled in Thermal Desktop and ATMT

The key model parameters are summarized below with the ATMT to TD correlation shown in Figure 4-23

- **Test Case 1** - 2D rectangular surface
- **Material** - Aluminum 6061 T6
- **Surface Finish** - Black Z306 paint
- **Orientation** - +Z Sun-facing
- **Orbit** - 45° inclination, 500 km maximum altitude
- **Scenario** - The simulation runs from: 1 Jan 2010 00:00:00 until 2 Jan 2010 00:00:00
- **Additional Notes** - +Z face is the only radiating surface

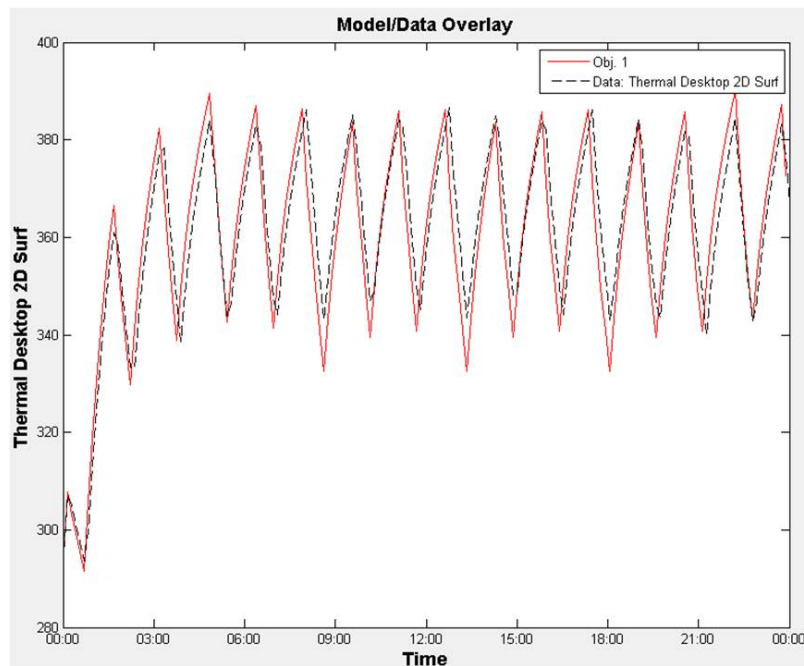


Figure 4-23: Thermal Desktop Validation of Test Case 1: Temperature at the Surface's Centroid

The ATMT temperature response appears to closely approximate the results from the TD model. Some of the key performance parameters include the peak to peak equilibrium temperature range (K), the settling time⁶ (Hr), the average percent error⁷ (%), and the temperature error⁸ (K). The performance parameters are summarized in Table 4.1. The average percent error and temperature error for Test Case 1 were 1.6% and 5.9 K respectively

⁶The settling time is defined as the time to reach 5% of equilibrium temperature range.

⁷The percent error is an absolute measurement of the error between the two models at any given time step. The average percent error considers all of the time steps for the simulation.

⁸The temperature error is an absolute measurement of the temperature difference between the two models at any given time step measured in Kelvin (K).

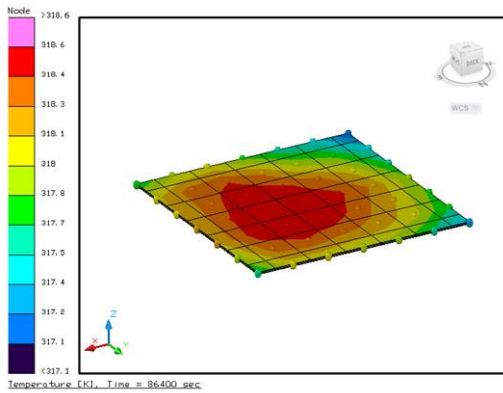
which are both well within the required ± 11 K post-TVac validation requirement discussed in Section 2.2.1.

Table 4.1: Validation Results: Test Case 1

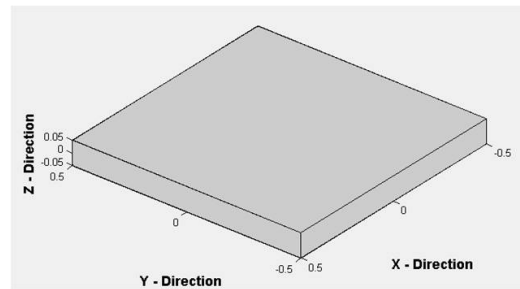
Test Case 1	Equil. Range (K)	T settle (Hr)	Avg. % Error	Temp Error (K)
ATMT	340 \rightarrow 387	5.1	1.6	5.9
TD	333 \rightarrow 390	5.1	—	—

4.5.2 Validation of Test Case 2: 3D Solid Rectangle

Test Case 2 was modeled in TD using the same parameters shown in Section 4.4.1. The TD model uses a 3D solid object with a $7 \times 7 \times 3$ node mesh whereas the ATMT 3D solid has a node at the centroid of each surface on the solid object. There was no need to add any contacts to the TD solid because the model only has one object. TD offers the choice of having the solid’s surfaces radiate internally (a feature that could be used for a transparent body), however for the solid aluminum rectangle there should be no internal radiation.



(a) 2D Surface Modeled in Thermal Desktop



(b) 2D Surface Modeled in ATMT

Figure 4-24: 3D Solid Rectangle Modeled in Thermal Desktop and ATMT

The key model parameters are summarized below with the ATMT to TD correlation shown in Figure 4-25

- **Test Case 2** - 3D solid rectangle
- **Material** - Aluminum 6061 T6

- **Surface Finish** - Black Z306 paint
- **Orientation** - +Z Sun-facing
- **Orbit** - 45° inclination, 500 km maximum altitude
- **Scenario** - The simulation runs from: 1 Jan 2010 00:00:00 until 2 Jan 2010 00:00:00
- **Additional Notes** - 5 W/K conductance value between all components

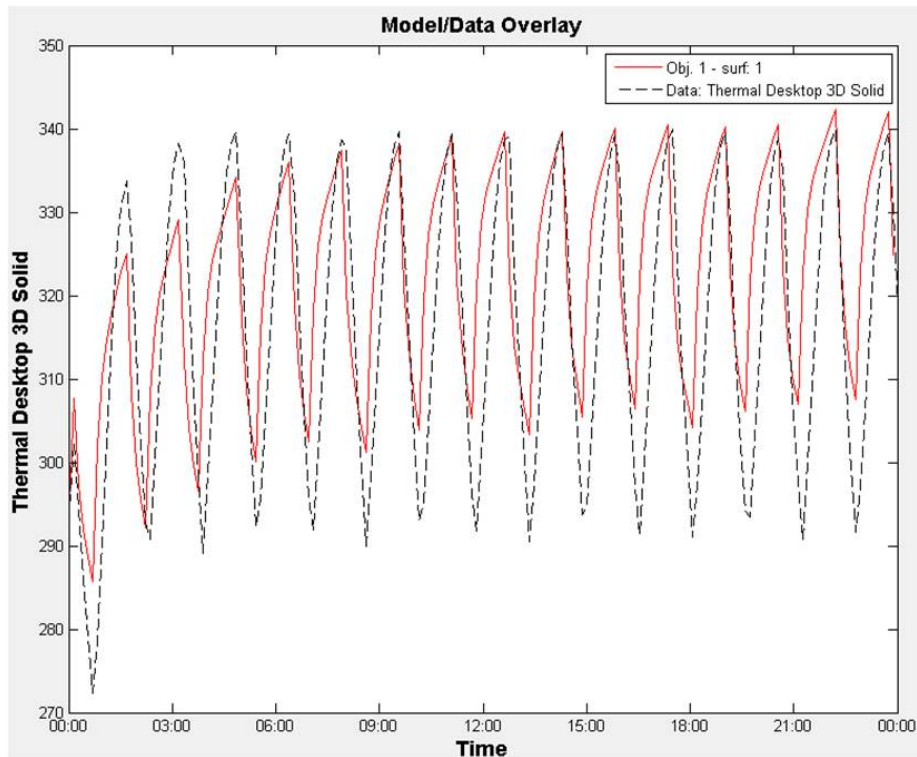


Figure 4-25: Thermal Desktop Validation of Test Case 2: Temperature at the Object’s Centroid

In this case the ATMT temperature response follows the same periodicity as the TD model however the accuracy is not as good as Test Case 1. The average percent error and temperature error for Test Case 2 were 3.3% and 10 K respectively which are still within the ± 11 K requirement. The major reason for the difference is due to the fact that ATMT uses a single node to represent each of the six surfaces as opposed to a 3D mesh. The 3D Temperature Averaging Algorithm discussed in Section 6.3 will theoretically correct this error by averaging the six surface temperatures at each time step. Preliminary analysis has shown that ATMT’s 3D Temperature Averaging Algorithm improves the performance by

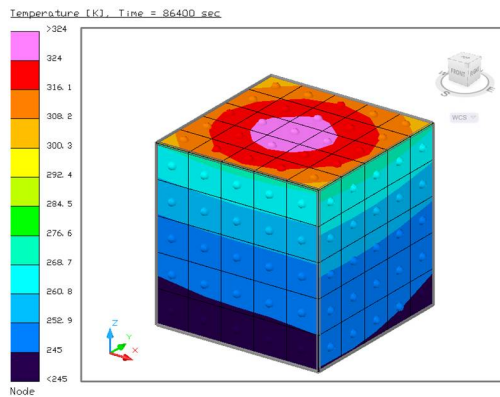
a factor of ~ 3 for a six surface rectangular object. The performance parameters for the ATMT to TD correlation are summarized in Table 4.2.

Table 4.2: Validation Results: Test Case 2

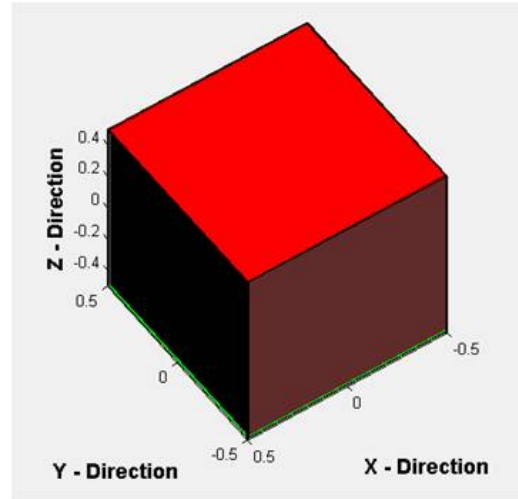
Test Case 2	Equil. Range (K)	T settle (Hr)	Avg. % Error	Temp Error (K)
ATMT	340 \rightarrow 387	5.1	3.3	10
TD	333 \rightarrow 390	5.1	—	—

4.5.3 Validation of Test Case 3: 2D Six Surface Box

Test Case 3 is significantly more complex than cases 1 and 2 and is the first place where the TD and ATMT models diverge in how they were constructed. Physically, the ATMT and TD representations of the 2D box appear similar as they were both constructed according to the parameters provided in Section 4.4.1. However, there are two main differences: the surface meshing scheme and the contacts between surfaces. The TD model uses a 5×5 node mesh for each of the six surfaces whereas the ATMT surfaces each have a single node at the centroid. Secondly, the contacts between surfaces in TD are established using surface edge contacts as opposed to an absolute conductance [1]. This difference is easily accounted for by changing the absolute conductance value in ATMT to 0.01 W/K which is the equivalent conductance used in the TD model. The TD and ATMT representations of the 2D box configuration are shown in Figure 4-26.



(a) 2D Six Surface Box Modeled in Thermal Desktop

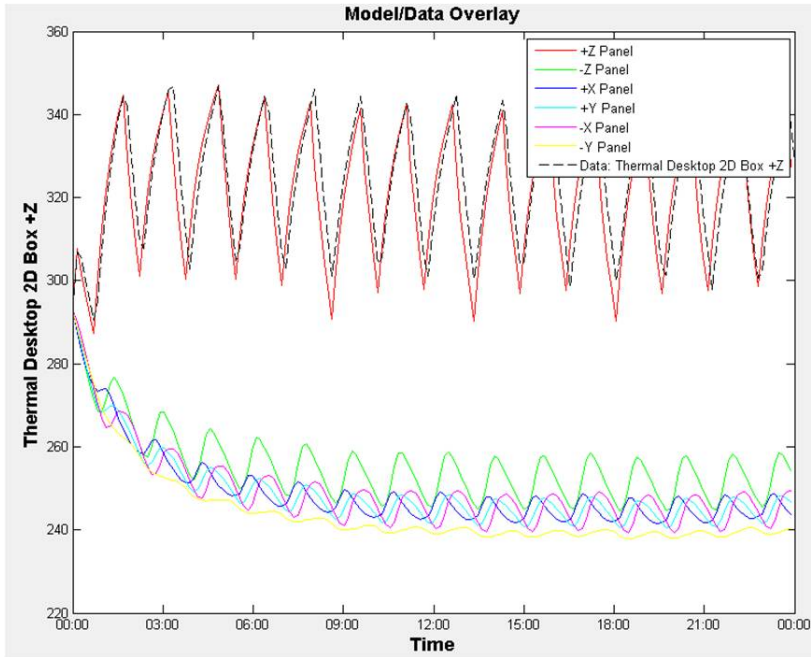


(b) 2D Six Surface Box Modeled in ATMT

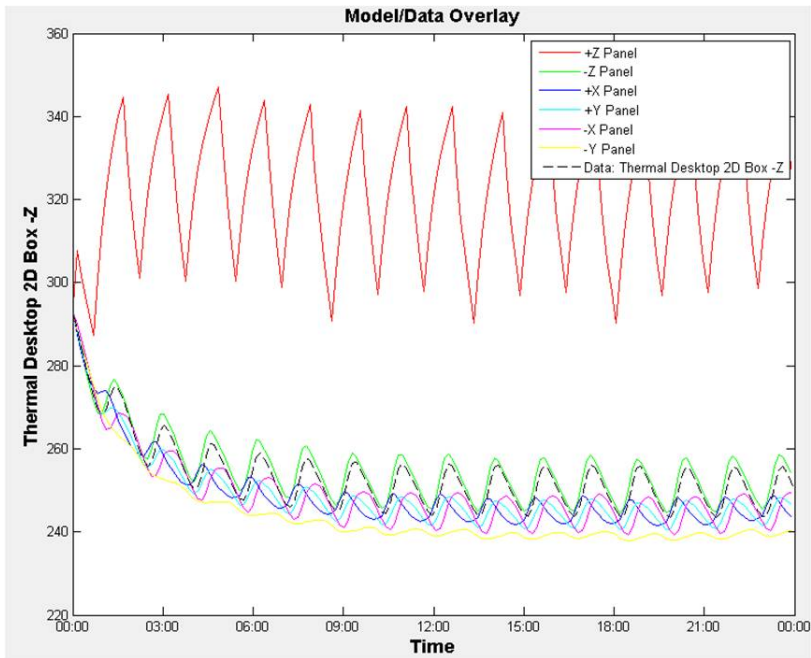
Figure 4-26: 2D Six Surface Box Modeled in Thermal Desktop and ATMT

The key model parameters are summarized below. The ATMT to TD correlation is shown for the +Z and -Z surfaces in Figure 4-27

- **Test Case 3** - 2D Six Surface Box
- **Material** - Aluminum 6061 T6
- **Surface Finish** - Black Z306 paint
- **Orientation** - +Z Sun-facing
- **Orbit** - 45° inclination, 500 km maximum altitude
- **Scenario** - The simulation runs from: 1 Jan 2010 00:00:00 until 2 Jan 2010 00:00:00



(a) 2D Six Surface Box Validation: Temperature at the +Z Surface's Centroid



(b) 2D Six Surface Box Validation: Temperature at the -Z Surface's Centroid

Figure 4-27: Thermal Desktop Validation of Test Case 3

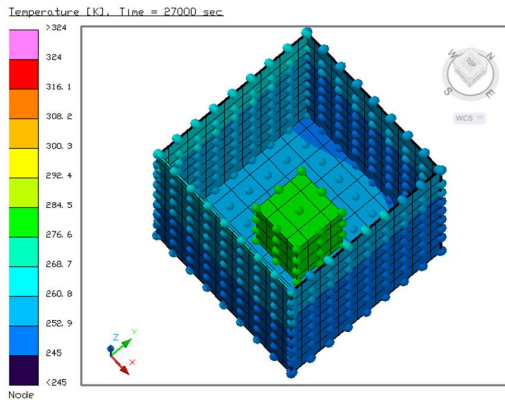
The ATMT and TD temperature responses are very similar demonstrating the ability of ATMT to accurately simulate 2D models on-orbit. The average percent error and temperature error for Test Case 3 were 1.9% and 5 K respectively which is on the same order as Test Case 1 and again well within the ± 11 K margin requirement. The performance parameters for the ATMT to TD correlation of Test Case 3 are summarized in Table 4.3.

Table 4.3: Validation Results: Test Case 3

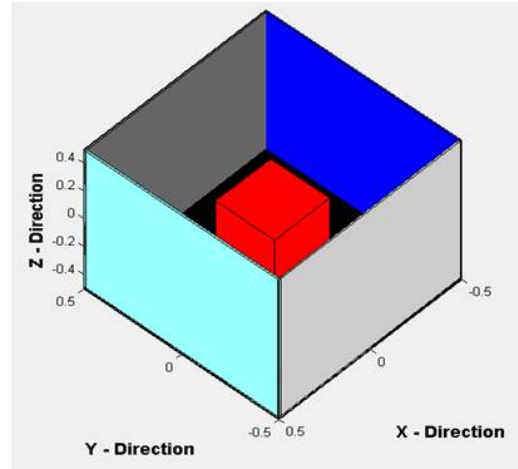
Test Case 3	Equil. Range (K)	T settle (Hr)	Avg. % Error	Temp Error (K)
ATMT	340 \rightarrow 387	5.0	1.9	5.0
TD	333 \rightarrow 390	5.0	—	—

4.5.4 Validation of Test Case 4: 3D Six Panel Box with Avionics Stack

Test Case 4 is the final case to be validated in TD. The TD and ATMT representations of the 3D box configuration are shown in Figure 4-26. Unfortunately several of the algorithms required for multi-object 3D modeling in ATMT are still being developed. The 3D Temperature Averaging Algorithm and 3D VF Shading Function discussed in the future work Section 6.3 are required to complete the validation of Test Case 4. ATMT Version 2.0 will include these new algorithms as well as several other advanced features enabling ATMT to be accurately correlated to industry standard modeling tools as well as thermal test data.



(a) 3D Six Panel Box with Avionics Stack Modeled in Thermal Desktop



(b) 3D Six Panel Box with Avionics Stack Modeled in Thermal Desktop

Figure 4-28: 3D Six Panel Box with Avionics Stack Modeled in Thermal Desktop and ATMT

4.6 Model Performance

There are many ways to judge the performance of a thermal model and choosing the right metrics is done based on a program's requirements. The previous work Section 1.4 presented a list of metrics by which ATMT should be evaluated. The two main performance metrics for ATMT are accuracy and usability. Model accuracy allows the thermal engineer to understand how well they can expect their thermal model to approximate the exact solution. The accuracy translates to the error that should be accounted for in each component's temperature margin. The usability metric identifies how well ATMT meets the thermal modeling requirements of a smallsat program. This section briefly summarizes how well ATMT performed according to the accuracy and usability metrics.

4.6.1 ATMT Accuracy

Understanding the accuracy of a thermal model is critical for designing an effective thermal control system. Section 4.4.1 presented four test cases each with increasingly complex features. The accuracy of each test case was evaluated through comparison with a much

more robust modeling program in Section 4.5. For the most part the accuracy was directly related to the complexity of the object. Test Case 1 matched the truth source with the least percent error (1.6%) while Test Case 2 was the least accurate (3.3%). A second trend in the accuracy was seen for 2D vs. 3D models. Test Case 3, a 2D model, actually outperformed Test Case 2. According to the ATMT performance metrics described in Section 1.4, $< 5\%$ error and < 11 K temperature difference were desired accuracy levels. ATMT outperformed these levels with a margin of 1.7% error and 1 K temperature difference. While the accuracy requirement was met and surpassed for the selected test cases, it is always good engineering practice to validate any new model prior to accepting it as truth.

4.6.2 ATMT Usability

The usability of ATMT is a measure of how well ATMT satisfies the thermal modeling requirements of a smallsat program. Unlike the accuracy metric, there is no uniformly defined way of measuring the usability. One of the best ways to determine a model's usability is to conduct a user program test. Four undergraduate seniors from the U.S. Air Force Academy were chosen to test ATMT based on their field of study and level of thermal modeling experience. The participants' fields of study included: Astronautics, Space Operations, Physics, and Systems Engineering. All of the participants were students in the FalconSAT-5 capstone design course who had a basic classroom-level knowledge of spacecraft thermal modeling. The students were asked to participate in two 1.5 hour studies as well as a survey at the completion of the test.

In the first study the students were given a 30 minute introduction to ATMT and asked to develop a thermal model of nanosat. The nanosat was a simple aluminum box which is used as a test bed for thermal and structural testing at the U.S. Air Force Academy. The participants were given the drawings shown in Figures B-7, B-8, and B-9 of Appendix B to reference for the dimensions and configuration of the six side panels and internal avionics stack. All of the participants were observed and timed throughout the first study. Many of the same comments, questions, and even bugs arose and were addressed on an individual basis. All of the participants finished building the nanosat model in the allotted 60 minutes with the fastest time being 35 minutes and the slowest time being 60 minutes.

The second study required the participants to run their model through the baseline scenario described in Section 4.4.1 and to analyze the results of the simulation. They were then asked to perform four trade studies of their choice using the model. All of the trade studies were discussed with the participants to ensure that they understood the implication of modifying various parameters and could predict the expected thermal response. At the end of the second study all of the students demonstrated an ability to predict the thermal response and were able to tailor their model design to meet a given set of requirements.

Finally, an ATMT user survey was administered to each participant inquiring about their experience with ATMT. The questions on the survey were tailored toward the four parts of usability (functionality, specificity, simplicity, and ease of use) described in Section 1.4. The participants' responses were positive concluding that ATMT is well suited as a preliminary thermal design tool for smallsat programs.

Chapter 5

ATMT User's Manual

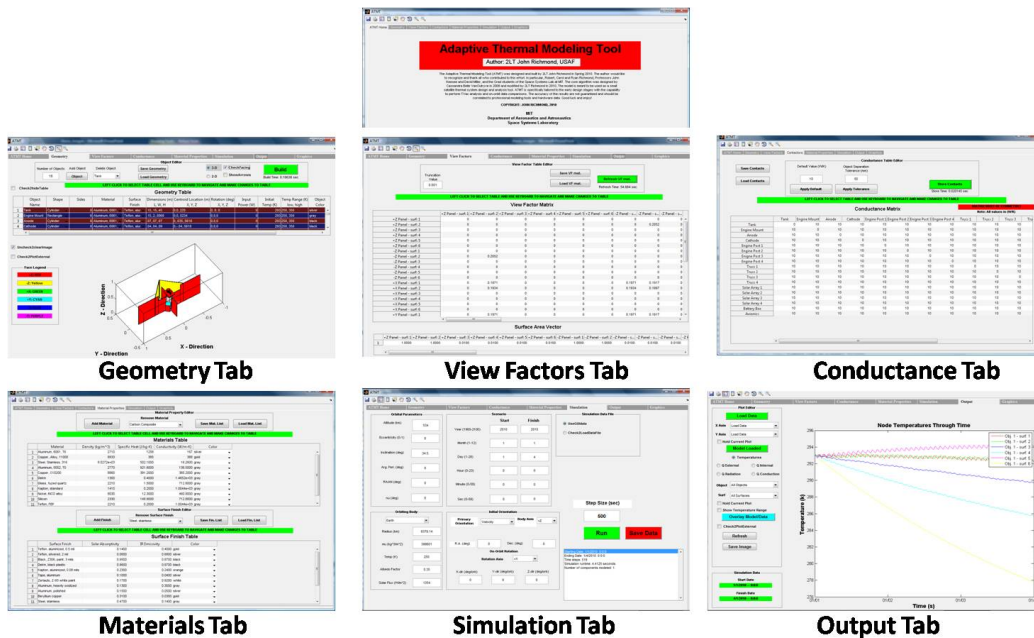


Figure 5-1: ATMT at a Glance

The Adaptive Thermal Modeling Tool (ATMT) was designed and built by 2LT John Richmond in spring 2010. As the focus of this master's thesis, ATMT was created with the aim of bridging the gap between commercially available thermal modeling software and satellite specific thermal algorithms. The core thermal solver was designed by Cassandra Belle VanOutryve in 2008 and modified by 2LT Richmond in 2010 [24]. The model is designed to perform on-orbit thermal trade studies for small satellite thermal performance modeling. ATMT is specifically tailored to the early design phases (conceptualiza-

tion through CDR) with the capability to perform TVac and on-orbit data comparisons. The results have been validated against Thermal Desktop for several but not all test cases. ATMT is accurate to within the 11 K thermal margin requirement described in Section 2.2.1. However, ATMT model results may not be exact and it is recommended that first time users follow the ATMA process in order to validate model results against test data or professional modeling tools.

5.1 Overview

The ATMT User's Manual describes how to utilize ATMT for spacecraft thermal modeling purposes. ATMT is designed to address the various thermal modeling requirements that a small satellite program will encounter throughout its acquisition lifecycle.

ATMT offers a MATLAB Graphical User Interface (GUI) for ease of use. The ATMT GUI includes six primary tabs offering the freedom and flexibility to model very simple or highly complex spacecraft geometries while minimizing the design burden on the user. The main Geometry Tab utilizes a Computer Aided Design (CAD) feature which allows the user to customize each of the model's components and to create a visual representation of the spacecraft model. The ATMT program allows the user to model the increasing levels of hardware complexity: component, assembly, and system for a range of 2D and 3D applications. After building the geometric configuration and parameterizing a model, the user is able to define and run an orbital simulation. The post processing features offer the ability to correlate the ATMT thermal model results with other software models as well as actual test data. Finally, the model results are used to make design modifications and scenario iterations for future test cases.

The current version (version 1.0) of ATMT is specifically tailored toward on-orbit thermal modeling. Future versions will allow the user to model two additional phases of thermal testing and analysis: hardware acceptance and thermal vacuum (TVac). Each sections of this User's Manual is associated with a tabs of the ATMT GUI. The ATMT Tabs are presented in the same order that they should be applied in practice. The functionality of each tab is discussed in detail in the following sections.

5.2 ATMT Home Tab

The ATMT Home Tab is the title tab that provides a short description of the author's thoughts and intentions. While the Home tab does not currently serve a functional purpose, it will be upgraded in the future to serve as a central routing point for miscellaneous items. Items such as a help menu and frequently asked questions (FAQ) will be stored on the ATMT Home Tab as well as revision titles made by future contributors. Figure 5-2 shows the ATMT Home Tab. Some of the important features include the MATLAB Tool Bar, ATMT Tab Menu, the figure expansion options and the GUI Tabpanel Editor.

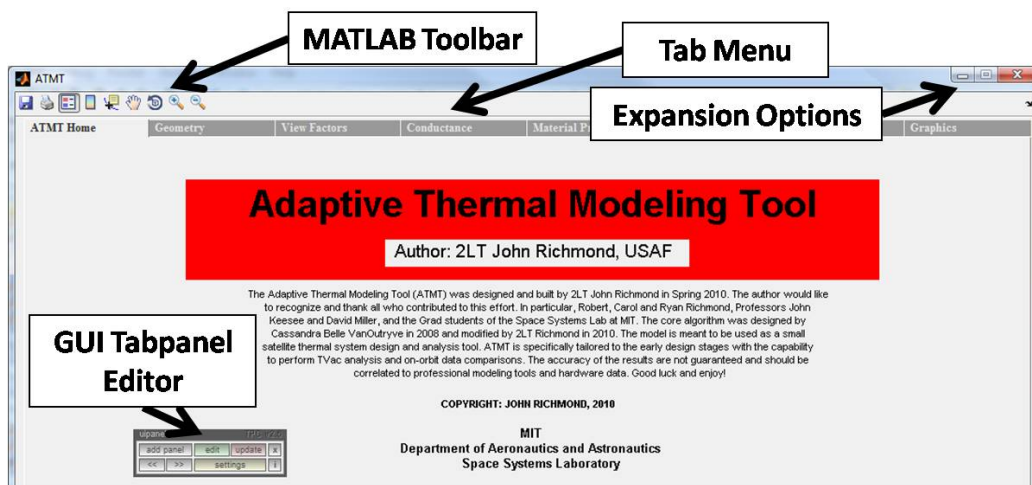


Figure 5-2: ATMT Home Tab

- **MATLAB Tool Bar** - The MATLAB Tool Bar has been specifically tailored to fit ATMT. Moving from right to left, the ATMT Tool Bar has the zoom options (zoom-out and zoom-in), rotate, pan, data cursor, insert color bar, insert legend, print, and save. Left clicking on any option will activate the tool and left clicking a second time will deactivate the tool. The save tool does not perform the same function as a MATLAB user is normally accustomed to seeing. In ATMT, the save icon will save the GUI as ATMT.m writing over the previous version and making all current values the new default options. Instead, each tab offers a Save button that can be used to store a “.mat” data file in the Saved Items subfolder within the ATMT folder.
- **Tab Menu** - The ATMT Tab Menu is the primary navigation method for the program.

Left clicking any tab will bring the user directly to the desired tab. All of ATMT's tabs are included in the menu bar. The tab menu is organized from left to right beginning with the ATMT Home Tab and ending with the Output Tab. The modeling process should follow the same left to right order under normal circumstances.

- **Expansion Options** - The expansion options allow the user to maximize, minimize, resize, or dock the figure window. The GUI contents will naturally resize to fit a docked or maximized figure window. However, resizing can cause the GUI screen to no longer remain centered in the figure window.
- **GUI Tabpanel Editor** - The GUI Tabpanel Editor is a feature that allows the user to reconfigure or edit a particular ATMT GUI tab. The editor is not available directly from the ATMT GUI but can be accessed by typing `“tabpanel('ATMT.fig','uipanel1')”` in the MATLAB workspace. Editing the GUI can lead to errors in the GUI M file. All data should be saved prior to making changes to `“ATMT.fig”`.

5.3 Geometry Tab

The Geometry Tab is the starting point for all thermal designs created within ATMT. The user has the flexibility to control many component parameters without requiring an extensive background in thermal engineering or satellite hardware. New geometric configurations can be created from the Object Editor Panel and edited within the Object Property Table. Completed geometries may be saved as a `“.mat”` file in the Saved Items subfolder for future use. The geometric configuration may be built or modified at any point before or after running a simulation. The Geometric Configuration Plot allows the user to visualize the 3D spatial orientation of all objects in the model. Figure 5-3 shows the five main facets of the Geometry Tab to include the: Object Editor, Object Property Table, Load and Save feature, Build button and the Geometric Configuration. This section describes the purpose and for each of the five items.

- **Object Editor** - The Object Editor Panel shown at the top of Figure 5-3, controls new geometric configurations and allows for modifications to preloaded geometries. There

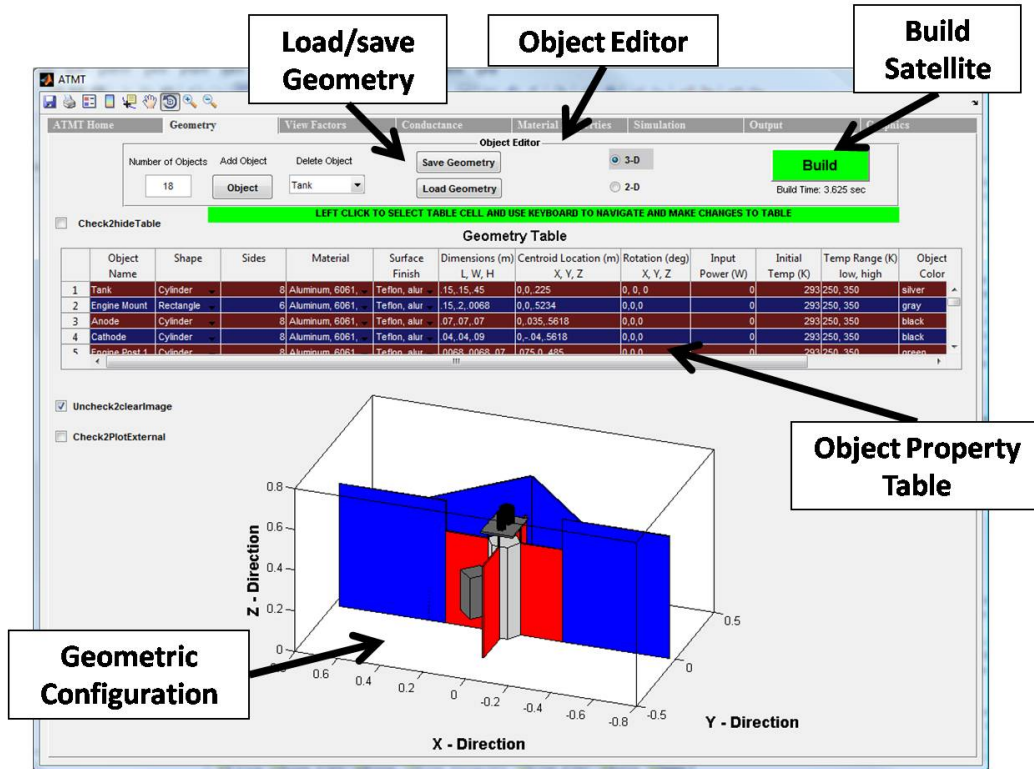


Figure 5-3: ATMT Geometry Tab

are three main components within the Object Editor Panel. From left to right in they are the: Number of Objects textbox, the Add Object button, and the Delete Object list. The Build button and corresponding 2D, 3D, and check facing options reside within the Object Editor Panel however; they are addressed in the Build Satellite feature section. Figure 5-4 shows the three options used to modify the number of objects in the model.

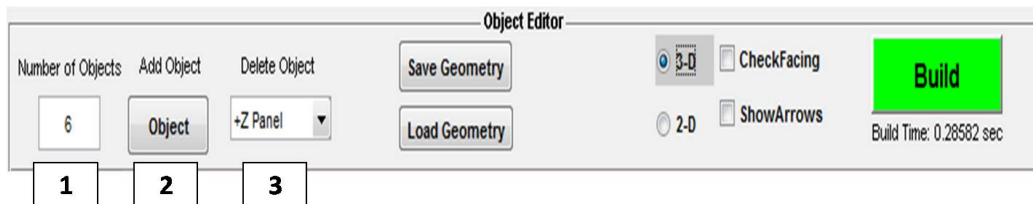


Figure 5-4: Geometry Tab: Object Editor Panel

1. **Number of Objects** - The “Number of Objects” textbox is used initially to add a desired number of components to the model. A minimum of 1 component is required with a maximum value limited only by MATLAB memory storage

($\sim 10,000$ components). ATMT is designed for small satellite applications and it is recommended that the user keep the number of components under 50. Left clicking in the Number of Objects textbox will activate the feature allowing the user to type an integer number. Pressing "enter" or clicking anywhere outside of the textbox will freeze the option and load the default parameter values for each object into the Object Property Table. If there are any components already in the Object Property Table when the command is executed, they will be erased and replaced with the new objects. Be sure to save all geometry data prior to creating a new model.

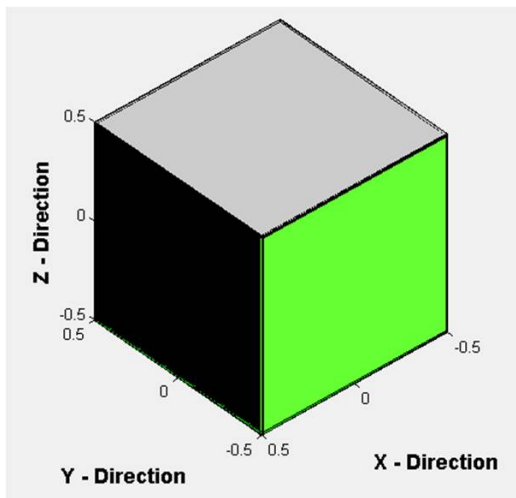
2. **Add Object** - The "Add Object" button allows the user to add one object per left button click. New objects appear in the last row of the Object Property Table with their default values and the default name "NEW OBJECT".
 3. **Delete Object** - The "Delete Object" list contains all of the objects currently in the Object Property Table. The down arrow allows the user to view the complete list. Left clicking an item will automatically delete the selected item along with its associated data. While there is no way to un-delete an object, any cell may be copied and pasted allowing objects to be replicated with ease. Deleting an object will also remove it from use in the View Factors and Conductance Tabs.
- **Object Property Table** - The Object Property Table contains the parameters for all objects in the model. All columns are editable allowing the user to modify various features according to component specifications. The Object Property Table's columns are each associated with a particular component parameter. The 14 different parameters are in detail below. It should be noted that versions of MATLAB up to and including R2009a show some anomalies when editing a GUI User Interface Table (uitable). The green caution sign in Figure 5-3, indicates that the Object Property Table should be edited with care. The proper way to edit the Object Property Table is to left click in the desired cell and use the keyboard to make edits and navigate throughout the table. Once a desired value is entered or a pull down menu selection is made, pressing the "enter" key will lock the entry in place. Using the left, right,

up, and down keyboard arrow keys is the best way to move to various cells within the table. Rapidly clicking cells and making quick changes may cause a MATLAB “Segmentation Violation” error, which is routed internal to the MATLAB GUI architecture. Newer versions of MATLAB (R2009b and greater) have been demonstrated to circumvent this error.

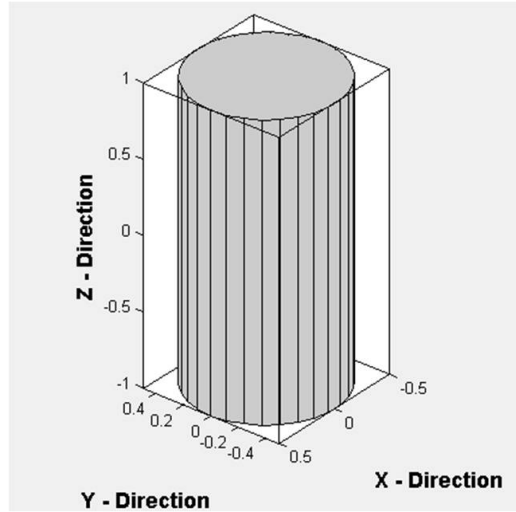
Object Property Table															
1	2	3	4	5	6				9	10	11	12	13	14	
Object Name	Shape	Sides	Material	Surface Finish	Dimensions (m) L, W, H	Centroid Location (m) X, Y, Z		Rotation (deg) X, Y, Z	Input Power (W)	Initial Temp (K)	Temp Range (K) low, high	Object Color	Density Multiplier	Model Object	
1	+Z Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	0,0,	0,	0	0	293	250, 350	red	1	<input checked="" type="checkbox"/>	
2	-Z Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	0,0,	7	0	8	0	293	250, 350	green	1	<input checked="" type="checkbox"/>
3	+X Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	495, 0,	-005	90,0,90	0	293	250, 350	black	1	<input checked="" type="checkbox"/>	
4	+Y Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	0, 495,	-005	90,0,180	0	293	250, 350	gray	1	<input checked="" type="checkbox"/>	
5	-X Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	-495,0,	-005	0,90,0	0	293	250, 350	blue	1	<input checked="" type="checkbox"/>	
6	-Y Panel	Rectangle	6 Aluminum, 6061	Black Z30I	1,1,01	0,-495,	-005	90,0,0	0	293	250, 350	maroon	1	<input checked="" type="checkbox"/>	

Figure 5-5: Geometry Tab: Object Property Table

1. **Object Name** - As the column heading indicates, the “Object Name” is the name used to identify the component throughout the View Factors, Conductance, and Output Tabs. The user can input any characters or numbers desired to represent a particular component. The default value for new components is “Obj. 1.”
2. **Shape** - In the current version of ATMT, the “Shape” of a component can either be a rectangle or a cylinder. The majority of hardware components used by small satellite programs are represented with great accuracy using these two shapes. Components such as structural panels and avionics trays are generally rectangular, while fuel tanks and comm. antennas are usually cylindrical. Future versions of ATMT will include shapes such as prisms, spheres and various user defined polygons. The default shape for new components is a rectangle. Figure 5-6 shows an ATMT representation of a rectangular avionics tray and a cylindrical tank.



(a) ATMT Rectangular Avionics Tray



(b) ATMT Cylindrical Tank

Figure 5-6: Example of a Rectangular and Cylindrical Object

3. **Sides** - The “Sides” parameter is the number of sides on a particular component. For a 3D rectangle the default is 6 sides, one per face. The sides parameter is particularly important for 3D cylinders which are represented by a multi-faceted cylindrical block. The minimum number of sides for a cylinder is 5 and the maximum can be as large as the user desires. It is recommended to use a value of ≤ 12 , 2 sides for the top and bottom and 10 along the cylinder’s height dimension. The number of sides directly affects the view factor (VF) computation time as well as the simulation run time. Figure 5-3 shows an image of the CASTOR geometric configuration featuring a 12 sided xenon tank as the central structural component. The 2D vs. 3D display option will affect the number of sides. Selecting the 2D option will automatically change the number of sides for a rectangle and cylinder to 1, which is located at the XY (length/width) mid-plane. The default value for new components is 6 sides.
4. **Material** - The “Material” parameter allows the user to select a predefined material for each component. The material database is discussed in Section 5.6. A pull down menu shows all of the available materials for each object. Left clicking on the desired material will automatically apply the selected material property

to the object. The material properties for each object are stored internally and accessed during each simulation. The default material for new components is the first material property in the list.

5. **Surface Finish** - The “Surface Finish” parameter controls the optical properties used for radiation calculations in the simulation. The surface finish pull down menu includes a list of all predefined surface finishes from the surface finish database. Upon selection, the surface finish is applied to all external faces of the component and the properties are stored for use in the simulation. The default surface finish for new components is the first surface finish in the list.
6. **Dimensions** - The “Dimensions” parameter allows the user to specify the objects Length (L), Width (W), and Height (H). Rectangle and cylinder dimension cells look similar but should be treated differently. All dimensions (m) should be entered using comma separated values (i.e. 1, 1, 1), much the same as one would create a row vector except without the brackets. For a rectangle, the length, width, and height correspond to the body X, Y, and Z axes. For a 2D surface rectangle, the height dimension is still required as it is used in the calculation of the object’s capacitance. For a cylinder the dimensions are radius, N/A, and height corresponding to the radius in the X and Y directions and the height along the Z direction. The N/A dimension is required and can be any number (i.e. 1). The default dimensions for new components are (1, 1, .01).
7. **Centroid Location** - The “Centroid Location” is used to place the object in 3D space. As is evident by the name, the centroid is the point at the center of the entire model (not necessarily the same as the cg). Similar to the dimensions parameter, the centroid location (m) requires an X, Y, and Z distance from the model origin (i.e. 1, 1, 1). The default centroid location for new components is (0, 0, 0).
8. **Rotation** - Any object can be rotated in 3D space in order to achieve a desired orientation. Rotations (deg) are entered about the X, Y, and Z axes (i.e. 90°, 0°, 0°). The “Rotation” feature is a onetime parameter used only in the model

design phase and is not to be associated with the rotation of the spacecraft on-orbit that is used in the Simulation Tab (See Section 5.7). The default rotation vector for new components is $(0^\circ, 0^\circ, 0^\circ)$.

9. **Input Power** - The “Input Power” allows the user to apply an internal heat load (W) to any component. The heat load is dissipated evenly through all sides through conduction heat transfer. Future versions of ATMT will allow the user to enter a time dependent power input used to model operational power states such as off, idle, safety mode, nominal, and peak power. The default input power for new components is 0 W.
10. **Initial Temperature** - The “Initial Temperature” parameter requires the user to specify the initial temperature state (K) for each component. The initial temperature input has a lower limit of 0 K (absolute zero). The default initial temperature for new components is 293 K.
11. **Temperature Range** - The component temperature range (K) is a function of the component’s operating temperature that may be found in a specification document (See section Section 2.2 for some typical component temperature ranges). The range is input as a lower and upper bound using the comma separated value technique. The component temperature range is plotted in the temperature output plots and out-of-bounds instances are identified. The default temperature range for new components is (250, 350) K.
12. **Object Color** - The “Object Color” parameter is used solely for visualization purposes. Component colors do not relate to surface finish properties. The object color can be selected from the drop-down menu similar to the way a material property was selected. The default color for new components is gray.
13. **Density Multiplier** - The “Density Multiplier” feature is used to model heterogeneous or discontinuous surfaces. For instance, a honeycomb solar panel may use a carbon face sheet adhered to aluminum honeycomb. Clearly the honeycomb has less surface area and mass than a solid aluminum sheet of the same dimensions. The “Density Multiplier” is a unitless input with a range of $(0 \rightarrow 1)$.

The default value for new components is 1 indicating a solid sheet.

14. **Model Object** - The last column of the Object Property Table allows the user to determine whether or not to model a specific component. If the “Model Object” box is checked, the component will appear in the geometrical configuration and will be incorporated into the simulation analysis. However, if the checkbox is unchecked, the object will not appear and will not be included in the analysis. This feature is useful for analyzing the geometric configuration of internal objects that may otherwise be shield by exterior objects or for neglecting the thermal affects of a particular component. Figure 4-18 shows an example of a 3D six panel box with the +Z panel removed showing the internal avionics.

- **Load and Save Feature** - Loading and saving the Object Property Table is very easy. Simply left click on either button and follow the prompts. When loading the Object Property Table, the user is asked to select a “.mat” file of their choice. Selecting a “.mat” file that was not saved from the Geometry Tab will result in an error. Upon loading an appropriate file, the geometry will automatically be displayed in the Geometric Configuration axes. The load and save feature is an incredibly valuable tool that allows the user to store all of their design work for future analysis. Design modifications may be made to an existing model before and after saving or loading a file.
- **Build** - The “Build” button takes in all of the user defined inputs from the Object Property Table and uses them to construct the geometric configuration. After changes are made within the Object Property Table the user must press the Build button to secure the inputs and rebuild the figure.

The 2D and 3D options located to the right of the build button in the Object Editor panel are also associated with the Build command. Only one of the buttons should be selected at any point in time. The selected button will be indicated with both the central dot and the darker background color. Selecting the 2D radio button will automatically rebuild the shape in its 2D configuration. The 2D option is especially

useful for reducing the VF calculation and simulation run times. Selecting the 3D button will rebuild the configuration using 3D objects.

When the 3D radio button is selected, the “Check Facing” and “Show Arrows” checkboxes appear within the Object Editor Panel. The Check Facing feature allows the user to visualize the orientation of each component within the model using HSV colors for each surface. All 3D components have their own body axis and reference frame, thus it is critical that the component be oriented in the desired configuration. The Face Color Legend appears next to the Geometric Configuration plot when the Check Facing box is selected. One example of when the check facing feature becomes especially important is for directional orientations in space. If the user wishes that the +Z axis face the sun initially, the +Z panel should have its own +Z direction face in the direction of the sun otherwise the model will assume that the user intends to have the sun-facing component remain shaded and opposite the direction of the sun. Figure 5-7 shows an example of the Check Facing feature for a 3D six panel box with an internal avionics stack.

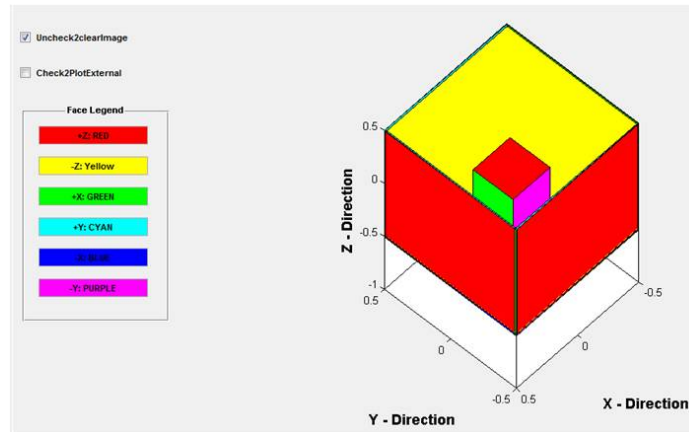


Figure 5-7: Geometry Tab: Check Facing Example

The Show Arrows feature is similar to the Check Facing option. Checking the Show Arrows checkbox will extend a ray from the centroid of each object’s +Z surface. This allows the user to visualize how the +Z surface is oriented and to determine which surfaces should share a view factor.

- **Geometric Configuration** - The “Geometric Configuration” is simply the graphical

representation of the geometry within the GUI. The plot can be fully manipulated using MATLAB Tool Bar options such as zoom, pan, rotate, or data cursor. Several additional features have been added to allow the user increased freedom to visualize the geometric configuration they have created. Figure 5-8 highlights the main features of the geometric configuration.

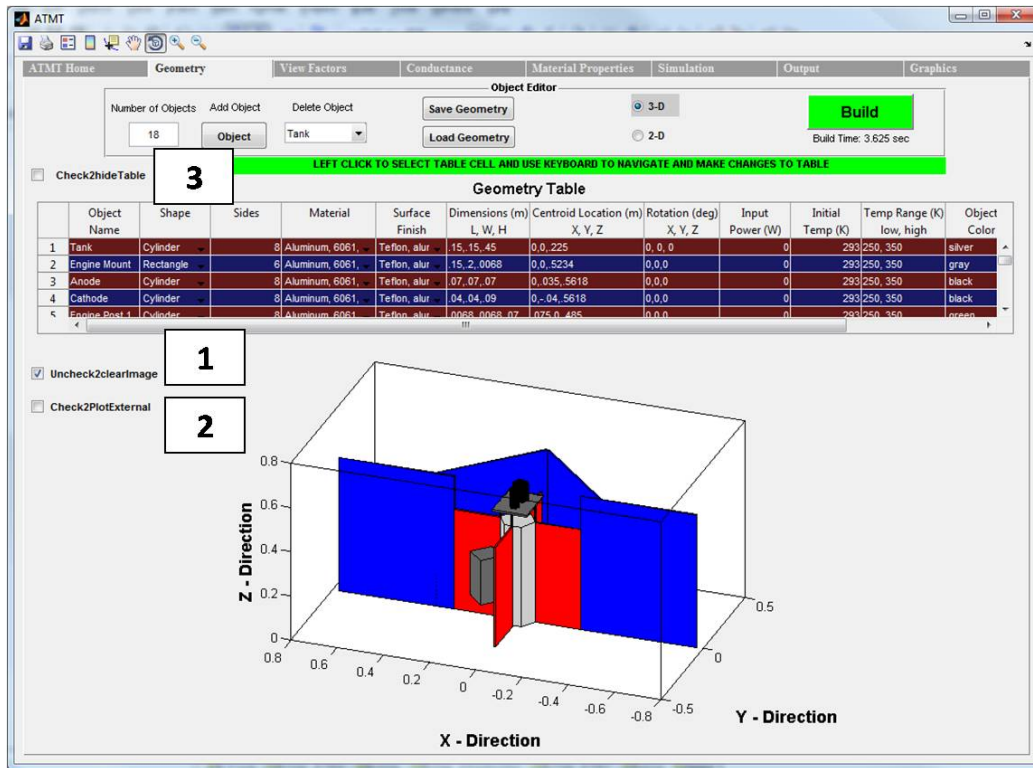


Figure 5-8: Geometry Tab: Geometric Configuration

1. **Show/Hide Image** - The “Show/Hide Image” feature allows the user to toggle the image on or off. Toggling the image off is the equivalent of clearing the image. Toggling the image on rebuilds the model based on the current model stored in the Object Property Table.
2. **Plot External** - Plotting a MATLAB figure external to the GUI is useful for many reasons. The user retains the ability to use MATLAB Tool Bar options while being afforded the additional ability to edit the plot in the same manner as any normal plot. The title, axis labels, legends etc. may be reformatted when the figure is plotted external to the GUI. When the “Plot External” checkbox is the

selected, the featured model will be plotted in a new figure window with the same plot azimuth and elevation as the GUI-based plot. Plotting the figure external to the GUI is the best way to save and manipulate the Geometric Configuration. Figure 5-9 shows an example of the CASTOR geometric configuration plotted external to the ATMT GUI. To close the new figure the user should uncheck the plot external checkbox. Closing the figure manually will result in an error later when the plot external checkbox is unchecked; however the error is negligible and can be ignored.

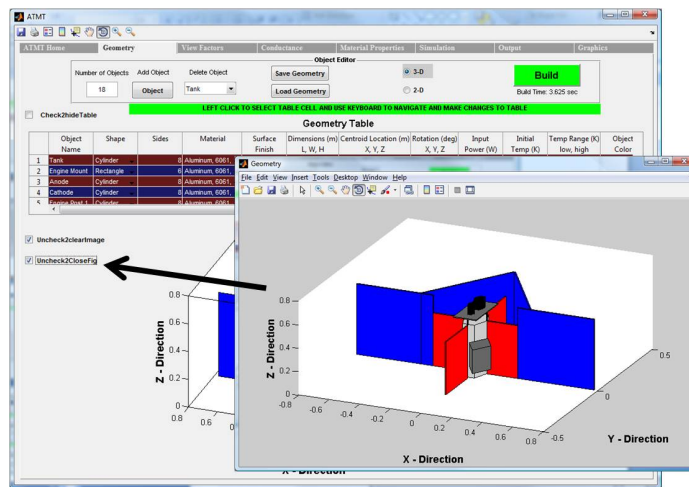


Figure 5-9: Geometry Tab: Plot External Example

- Hide Table** - The “Hide Table” feature allows the user to expand the Geometrical Configuration within the GUI. The Object Property Table will disappear but its data is stored. A simulation may be run as normal when the Hide Table checkbox is selected. Unchecking the Hide Table checkbox will restore the Geometric Configuration to its default size and position and will make the Object Property Table visible once again. Figure 5-10 shows an example of the CASTOR Geometric Configuration plotted in the ATMT GUI with the Object Property Table hidden and the axes expanded. Notice that the Hide Image checkbox becomes invisible but the Plot External feature is still available.

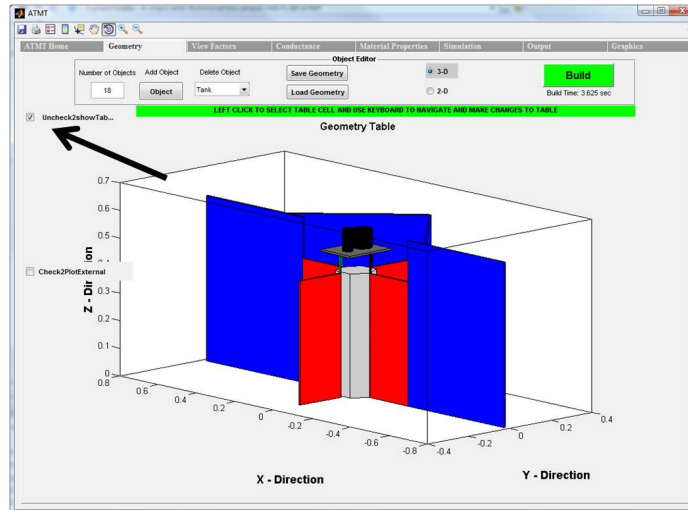


Figure 5-10: Geometry Tab: Hide Table Example

5.4 View Factors Tab

After fully defining the model in the Geometry Tab the user should proceed to the View Factors Tab. The View Factors Tab is one of the most important features of ATMT. ATMT automatically calculates the model's VFs using the viewfactor.m algorithm developed by Nicolas Lauzier and Daniel Rouse at Univeristé Laval in 2003 [12]. The Lauzier Rouse algorithm allows for VF_{ij} and VF_{ji} to be calculated between any two planar surfaces i and j defined by a finite number of vertices (must be at least 3) in 3D space. Furthermore, the algorithm calculates the surface area A_i and A_j for the two surfaces. These features significantly reduce the burden on the user of having to calculate by hand the view factors for what may be complex surfaces involving shading, blockage, and non-textbook orientations. The viewfactor.m algorithm has been modified by 2LT John Richmond for use in the overarching VFmatrix.m matrix formulation algorithm.

The View Factors Tab is automatically influenced by the Geometry Tab causing the row and column headers to match the current state of the design. The actual VF Matrix and Surface Area Vectors are not refreshed until the "Refresh VF mat." button is pressed. Similar to the Geometry Tab, the View Factor Tab offers the user a Load and Save feature, which saves time for large VF matrices. The View Factors Tab is comprised of the View Factor Table Editor, the VF Matrix and the Surface Area Vector. The image shown in

Figure 5-11 is an example VF calculation for a 3D solid box with an internal avionics stack.

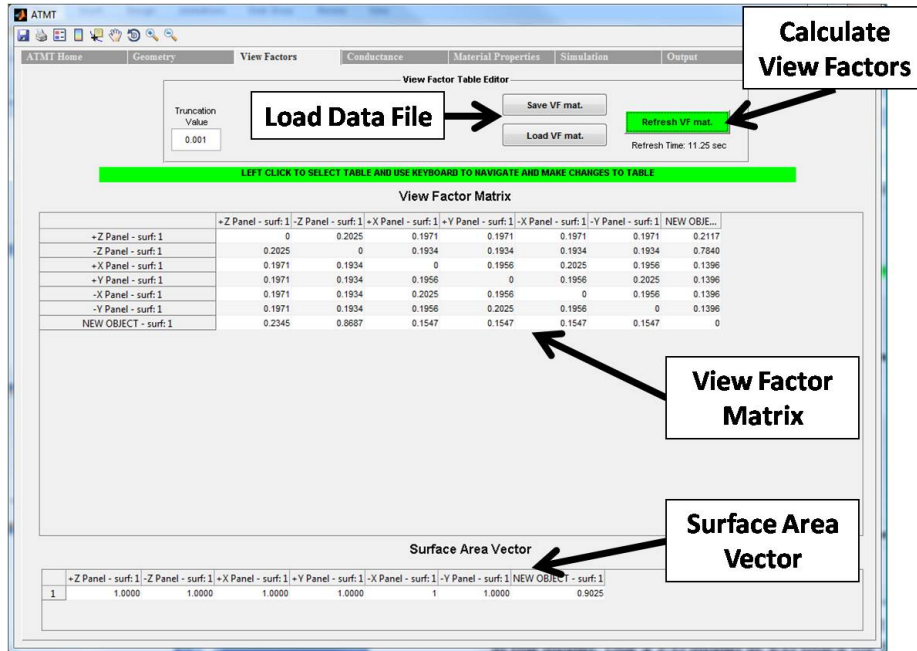


Figure 5-11: ATMT View Factors Tab

- View Factor Table Editor** - The View Factor Table Editor allows the user to decide between computing a new VF Matrix and Surface Area Vector or loading a previously calculated matrix. As is evident by Figure 5-12, space is available in the View Factor Table Editor for future additions. One potential addition is a shading tolerance that would be used to minimize the VF Matrix computation time by keeping the algorithm from calculating shaded or blocked VFs that are below a particular value.

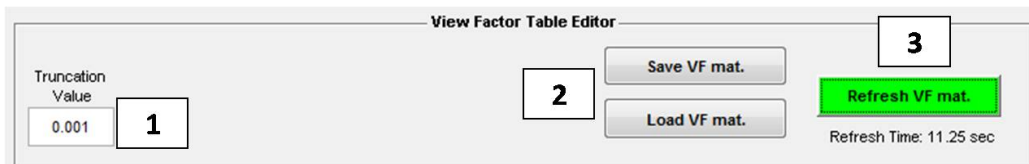


Figure 5-12: View Factors Tab: Matrix Editor Panel

- Truncation Value** - The “Truncation Value” ($0 \rightarrow 1$) is used to quickly eliminate any VFs that are below a particular threshold. This reduces the computational burden on the core solver by eliminating the need to perform radiation calculations between surface pairs without a view factor. Small VFs ≤ 0.001

have a negligible effect on the model accuracy and may be neglected to reduce the simulation run time.

2. **Load and Save Feature** - Identical to the Load and Save feature in the Geometry Tab, the View Factors Tab also allows the user to either load or save the VF Matrix and Surface Area Vector. This option becomes especially important with complex 3D geometries because the VF computation time increases linearly with the total number of surfaces in the model. Even with small geometric changes to an existing model, the user may opt to load a predefined VF Matrix to reduce computation time.
 3. **Refresh VF Mat.** - The “Refresh VF Mat.” button uses the model geometry from the Object Property Table to run the VFmatrix.m algorithm. The algorithm takes into account several key concepts to produce the correct VF Matrix in the minimum time. For instance, surfaces on the same object will not have VFs to one another because each surface’s 180° outward normal field of view (FOV) is obstructed by the 3D nature of the object. Currently, a shading detection algorithm is being developed and its future use is described in Section 6.3. The user may manually change a VF to a value ($0 \rightarrow 1$) depending on the geometry. VF calculations can be the most computationally intensive aspect of the entire ATMT program and it is suggested that the user stick to the recommended number of objects and use the 2D option whenever possible. For reference, a box consisting of six independent two dimensional surfaces will have a (6×6) VF Matrix requiring 36 iterations and taking on average 7 seconds to run the algorithm. MIT’s CASTOR nanosat has 18 components with a total of 126 surfaces. The CASTOR VF Matrix is (126×126) requiring 15,876 potential iterations and takes ~ 20 minutes to run the algorithm on a Dual Core processor.
- **VF Matrix** - The VF Matrix is a fully editable GUI table that stores the active VF Matrix (i.e. the values currently displayed in the matrix). The VF Matrix is encapsulated in a MATLAB uitable. The uitable offers two scroll bars in order to scroll left to right as well as up and down. Two items of interest are the column and

row headers as well as the individual matrix values. Figure 5-13 shows an example ATMT VF Matrix for a 2D box with an internal avionics stack.

	Obj. 1 - surf: 1	Obj. 1 - surf: 2	Obj. 1 - surf: 3	Obj. 1 - surf: 4	Obj. 1 - surf: 5	Obj. 1 - surf: 6
Obj. 1 - surf: 1	0	0	0	0	0	0
Obj. 1 - surf: 2	0	0	0	0	0	0
Obj. 1 - surf: 3	0	0	0	0	0	0
Obj. 1 - surf: 4	0	0	0	0	0	0
Obj. 1 - surf: 5	0	0	0	0	0	0
Obj. 1 - surf: 6	0	0	0	0	0	0

Figure 5-13: View Factors Tab: View Factor Matrix

1. **Column and Row Headers** - The column and row headers are labeled with the component name first followed by the surface number (i.e. “obj. 1 - surf 1”). Any 3D object will have at minimum 5 sides (6 for a rectangle) all of which could theoretically have a VF with any other surface. For this reason it is important that the column and row headers indicate specifically the surface of interest.
 2. **VF Matrix Values** - The matrix values represent the VF_{ij} and VF_{ji} between any two surfaces i and j . Unlike the Conductance Matrix discussed in Section 5.5, the VF Matrix is rarely symmetric. The values should always be in the range $(0 \rightarrow 1)$ and are all unitless. For components that are shaded or have very small view factors $\ll 0.001$ the user can manually insert a value of 0 into the matrix.
- **Surface Area Vector** - The last feature on the View Factors Tab is the Surface Area Vector. The entries to the vector may be either loaded or calculated as described in the Refresh VF Mat. section above. All surface areas (m^2) are influenced by the Density Multiplier for a given component as described in Section 5.3. Surface Area Vector entries are not editable by the user in the current version of ATMT.

5.5 Conductance Tab

With a fully defined satellite model established in Section 5.3, and the VFs stored in Section 5.4, the user should proceed to the Conductance Tab. The Conductance Tab is used to define the interaction between components used in the transient conduction calculations. As

one will find in any thermal modeling program, the contacts between surfaces are probably the most arbitrary and least understood interactions within a thermal model. Conductive heat transfer between components dependent on many factors, see Section 4.1 for an explanation of the assumptions that go into determining the contact strength between two surfaces.

The ATMT conductance approach is based off of the Thermal Desktop “absolute conductance” scheme which applies the absolute conductance value (W/K) across the entire contact line or area. In order to minimize the unnecessary complexity of identifying and defining the contacts between objects, the user can simply select a conductance value and allow the conductance algorithm to autogenerate the Conductance Matrix [1]. Similar to the Geometry and View Factor Tabs, the Conductance Tab offers the user a Load and Save feature to keep from having to redefine the contacts between surfaces. The Conductance Tab is comprised of the Conductance Table Editor and the Conductance Matrix. Under normal circumstances the Conductance Tab is fully automated allowing the user to simply verify that the desired values are applied and move on to the next stage of the model. Figure 5-14 shows an example of the Conductance Tab with the CASTOR nanosat’s Conductance Matrix preloaded in the uitable.

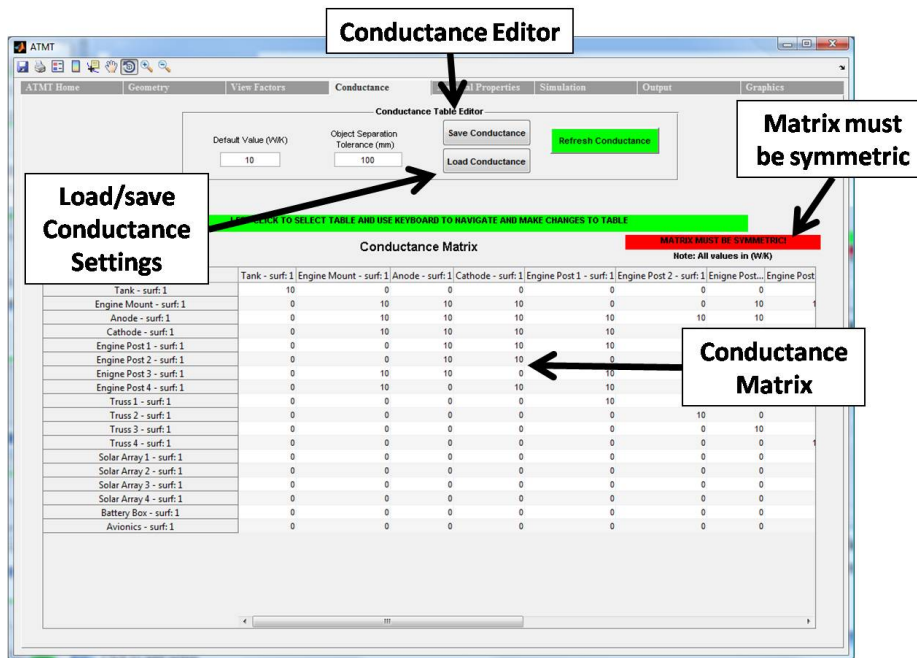


Figure 5-14: ATMT Conductance Tab

- **Conductance Table Editor** - The Conductance Table Editor allows the user to decide between using the automatically derived Conductance Matrix, which was established when ‘Build’ was pressed in the Geometry Tab, and loading a previously calculated Conductance Matrix. A default conductance value may be applied to the current Conductance Matrix which resets all entries to the selected conductance value. Figure 5-15 shows a close up display of the Conductance Table Editor.

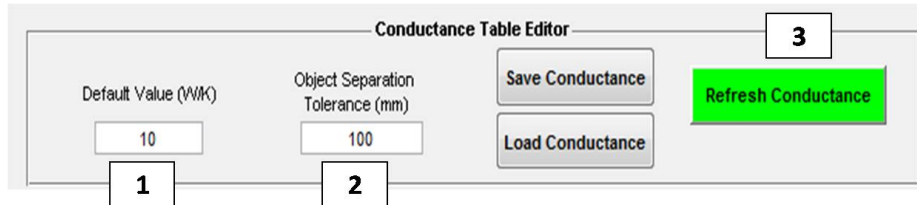


Figure 5-15: Conductance Tab: Conductance Table Editor

- **Load and Save Feature** - The Conductance Table Editor’s Load and Save features act identically to those of the View Factors Tab allowing the user to store and recall a Conductance Matrix.
 1. **Default Value** - The “Default Value” option allows the user to select an appropriate value of absolute conductance (W/K) and apply it to all of the contact points between objects. Left clicking the textbox places the blinking cursor in the box. Use the keyboard to enter a numeric value greater than 0 within the box. while negative values are currently not flagged as an error, they will produce highly inaccurate results. A larger conductance value implies greater conduction between components.
 2. **Object Separation Tolerance** - The “Object Separation Tolerance” function is used to automatically determine which components or surfaces are in contact based on a specified tolerance value (m). For example, applying a tolerance of 100 mm would cause neighboring components that have a vertex within the 100 mm range to be assigned a conductance value greater than zero. All components outside of the 100 mm tolerance range would be deemed out of contact and would receive a conductance value of 0 W/K. Applying an “Object Separation

Tolerance” will maintain the symmetry of the Conductance Matrix.

3. **Refresh Conductance** - The “Refresh Conductance” button runs the conductance algorithm to refresh the current Conductance Matrix. The button must be pressed if changes are made to an autogenerated or preloaded Conductance Matrix. If manual changes are made to the Conductance Matrix the user should not use the Refresh Conductance option because it will overwrite any updates that were made. The simulation will load and apply the current Conductance Matrix.

- **Conductance Matrix** - The Conductance Matrix is a fully editable uitable that stores the active Conductance Matrix for future conduction calculations. Similar to the VF Matrix, the Conductance Matrix is navigated by using the scroll bars to move from left to right and up and down as desired. Two items of interest are the column and row headers and the individual matrix values. Figure 5-16 shows the ATMT Conductance Matrix for a 2D rendition of the CASTOR nanosat.

	Tank - surf: 1	Engine Mount - surf: 1	Anode - surf: 1	Cathode - surf: 1	Engine Post 1 - surf: 1	Engine Post 2 - surf: 1	Engine Post 3 - surf: 1	Engine Post 4 - surf: 1	Truss 1 - surf: 1	Truss 2 - surf: 1	Truss 3 - surf: 1	Truss 4 - surf: 1	Solar Array 1 - surf: 1	Solar Array 2 - surf: 1	Solar Array 3 - surf: 1	Solar Array 4 - surf: 1	Battery Box - surf: 1	Avionics - surf: 1
Tank - surf: 1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Engine Mount - surf: 1	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Anode - surf: 1	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cathode - surf: 1	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Engine Post 1 - surf: 1	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
Engine Post 2 - surf: 1	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
Engine Post 3 - surf: 1	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
Engine Post 4 - surf: 1	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
Truss 1 - surf: 1	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
Truss 2 - surf: 1	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
Truss 3 - surf: 1	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
Truss 4 - surf: 1	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
Solar Array 1 - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
Solar Array 2 - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
Solar Array 3 - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
Solar Array 4 - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
Battery Box - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
Avionics - surf: 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Figure 5-16: Conductance Tab: Conductance Matrix

1. **Column/Row Headers** - The Conductance Matrix’s column and row headers are labeled with the object’s name. Columns and rows are automatically added or

deleted when the Add Object or Delete Object button is pressed in the Geometry Tab.

2. **Conductance Matrix Values** - The Conductance Matrix values represent the absolute conductance (W/K) between all components. The matrix should be checked for symmetry. The values on the main diagonal are automatically set to a value of 0 and should be unchanged. ATMT uses a lumped parameter nodal approach to thermal modeling as opposed to using a finite element mesh (FEM) architecture. For this reason it is assumed the each component remains at a uniform temperature. The conductance value is applied to the object's center of mass and is distribute appropriately to all contact surfaces.
3. **Symmetric Matrix** - Unlike the VF Matrix, the Conductance Matrix must be symmetric. The ATMT model assumes that the interaction between two components or surfaces remains constant throughout the simulation lifetime and is equal in magnitude and effect for both components.

5.6 Material Properties Tab

The Material Properties Tab allows the user to customize the materials and surface finish database available within the model. Two predefined databases of material and surface finish properties are loaded when ATMT is first opened and they are automatically set as the default lists within the “Material” and “Surface Finish” columns of the Object Property Editor. The user is not required to perform any functions within the Material Properties Tab in order to design and build a thermal model or run a simulation. The Material Properties Tab allows the user to modify the specific properties of an existing material or surface finish or even add new materials and finishes to the database. The material and surface finish databases were constructed from Appendices A and B of David Gilmore's “Spacecraft Thermal Control Handbook” [7]. This section discusses the functionality of the Material Properties Tab shown in Figure 5-17.

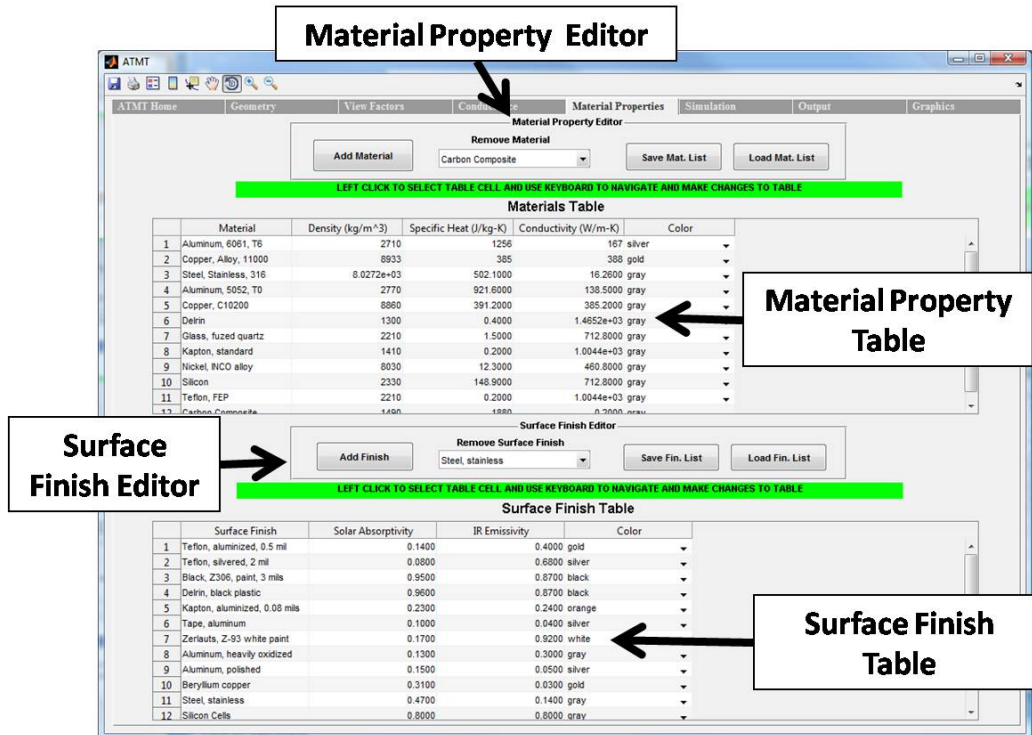


Figure 5-17: ATMT Materials Tab

- Material Property Editor** - The Material Property Editor allows the user to customize the Materials Table and the properties of each material. Similar to the Object Editor Panel discussed in Section 5.3, the Material Property Editor allows the user to add and remove properties as well as save and load material property tables. All changes to the Materials Table via the Material Property Editor are automatically reflected in the Material column of the Geometry Tab. Additional material property columns may be added to or deleted from the Material Table using the uipanel editor referenced in Section 5.2. The Materials Table is a fully editable MATLAB uitable. Figure 5-18 shows a close up display of the Material Property Editor as well as the material property columns in the Materials Table.

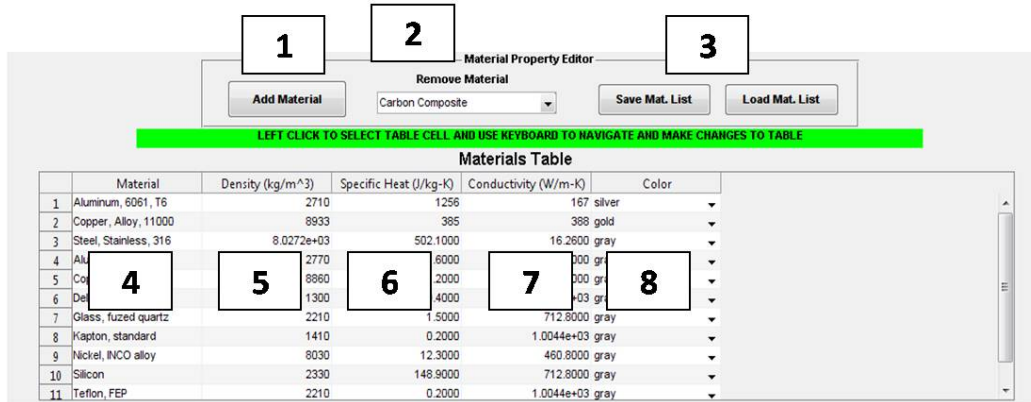


Figure 5-18: Materials Tab: Material Property Editor

1. **Add Material** - Adding a material to the Materials Table can be accomplished by left clicking on the “Add Material” button. The newly added material will appear in the last row of the Materials Table. New materials are automatically assigned the default name “NEW MATERIAL” and are given a default density (3000 kg/m^3), specific heat (1000 J/kgK), conductivity (100 W/mK), and color (gray).
2. **Remove Material** - Removing a material is done by selecting the “Remove Material” list, scrolling to the desired Table material and left clicking the material for removal. The selected material is automatically removed from the list and the subsequent row takes its place in the Materials Table.
3. **Load and Save Feature** - The Load and Save feature allow the user to store and recall a Materials database. Left clicking the Load or Save button will prompt the user to either enter a filename for the .mat file, or select a file from the current directory. Loading a previously saved Materials Table will automatically refresh the current display with the selected database file.
4. **Material** - The material column within the Materials Table is simply the name of the material. The user may edit any cell in the material column using the keyboard. Pressing “enter” or clicking outside of the highlighted cell will lock the name in place in both the Materials and Geometry Tabs.
5. **Density** - The material density (ρ) can be assigned to a cell in the second column

of the Materials Table. The density should be a positive value entered with units (kg/m^3).

6. **Specific Heat** - A material's specific heat (C_p) is a measure of the energy (joules) required to raise 1 kilogram (kg) by 1 (K). The user may adjust a material's specific heat by entering a positive value with units (J/kgK).
7. **Conductivity** - Conductivity (k) is a measure of a materials ability to conduct heat. The user may adjust a material's conductivity by entering a positive value with units (W/mK).
8. **Color** - Currently the material color feature does not serve a functional purpose. Future versions of ATMT will assign either the material or surface finish color to be the default color for a component made of the specific material.

- **Surface Finish Editor** - The Surface Finish Editor functions identically to the Material Property Editor described above and should be treated in the same manner. The selected surface finish is applied equally to all faces of an object. Future versions of ATMT will allow the user to specify different finishes for each surface. Figure 5-19 shows a close up display of the Surface Finish Editor as well as the property columns in the Surface Finish Table.

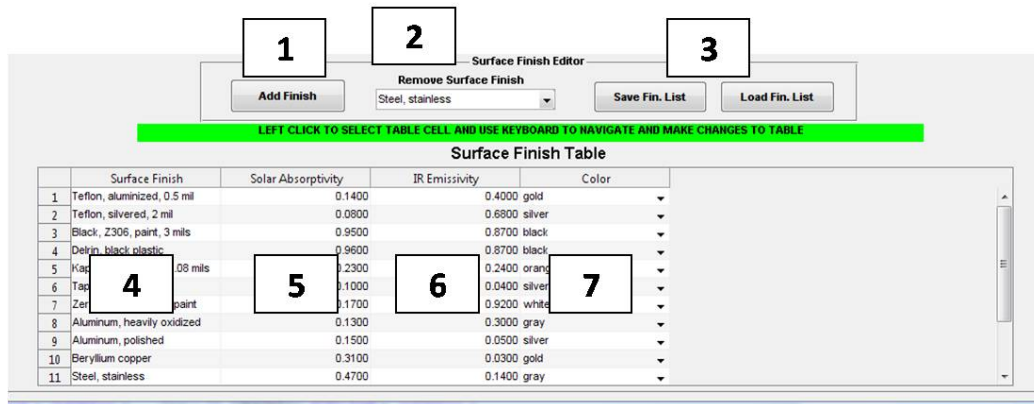


Figure 5-19: Materials Tab: Surface Finish Editor

1. **Add Finish** - Adding a surface finish to the Surface Finish Table can be accomplished by left clicking on the “Add Finish” button. The newly added surface

finish will appear as the last row of the Surface Finish Table. New surface finishes are automatically assigned the default name “NEW FINISH” and are given a default solar absorptivity (0.5), IR emissivity (0.5), and color (Hot Pink).

2. **Remove Finish** - Removing a surface finish is done by selecting the “Remove Finish” list, scrolling to the desired surface finish and left clicking the surface finish for removal. The selected surface finish is automatically removed from the list and the subsequent row takes its place in the Surface Finish Table.
3. **Load and Save Feature** - The Surface Finish Editor’s Load and Save feature functions identically to that of the Material Property Editor.
4. **Surface Finish** - The surface finish column within the Surface Finish Table contains the name of the surface finish. The user may edit any cell in the surface finish column using the keyboard. Pressing “enter” or clicking outside of the highlighted cell will lock the name in place in both the Materials and Geometry Tabs.
5. **Solar Absorptivity** - A surface finish’s solar absorptivity (α) is the fraction of incident solar energy (in the solar/visible spectrum) that is absorbed by the component coated in the surface finish. Among many factors, the solar absorptivity is a function of the surface finish color, roughness, tone, and oxidization level. While the solar absorptivity tends to vary with time and temperature, ATMT will assume a constant value for (α) over the simulation lifetime. The solar absorptivity is unitless and should always reside in the range of (0 \rightarrow 1).
6. **IR Emissivity** - A surface finish’s IR emissivity (ϵ) is the fraction of energy in the IR spectrum, that is emitted by the component coated in the surface finish. The IR emissivity is also a function of the surface finish properties but is assumed constant for temperatures in the IR spectrum. The IR emissivity is unitless and should always reside in the range of (0 \rightarrow 1).
7. **Color** - The surface finish color does not serve a functional purpose in the current version of ATMT.

5.7 Simulation Tab

The ATMT Simulation Tab is the final step of the thermal model setup. In this tab the user will define the parameters necessary for running a simulation and computing on-orbit thermal results. The Simulation Tab is fully editable and should be reviewed prior to each simulation to ensure the desired conditions have been identified. The Simulation Tab shown in Figure 5-20 contains six main panels each of which are described in detail below.

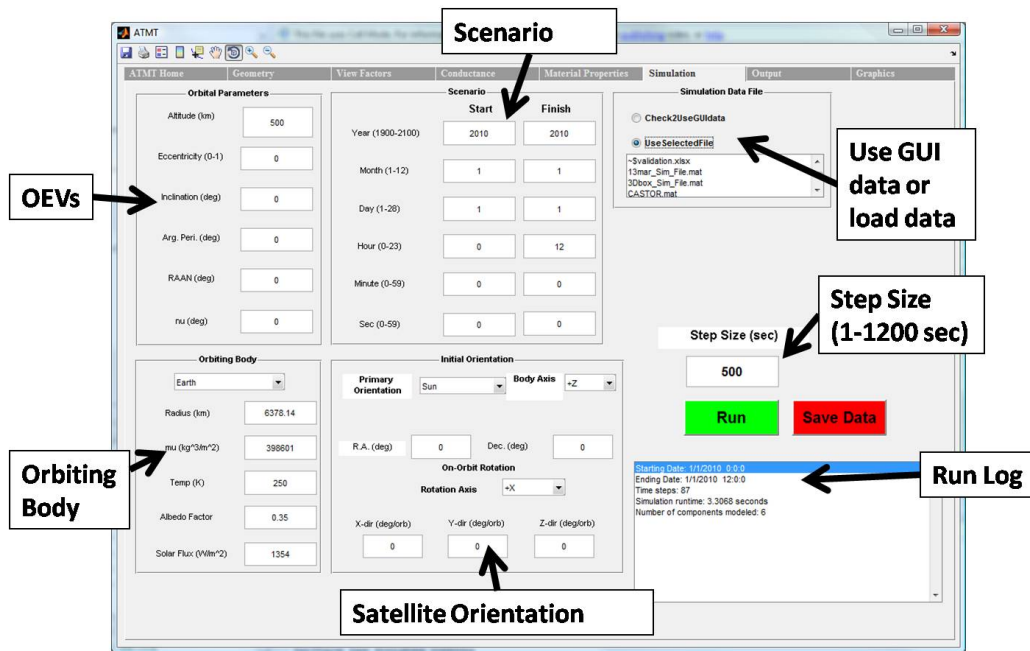


Figure 5-20: ATMT Scenario Tab

- Orbital Parameters Panel** - The Orbital Parameters Panel allows the user to define the orbit and position of the satellite along that orbit. The user first chooses the altitude, which is used to calculate the semi-major axis (a). The rest of the inputs are the 5 remaining orbital elements: eccentricity (ϵ), inclination (i), argument of perigee (ω), right ascension of the ascending node (Ω), and true anomaly (ν). Figure 5-21 shows a close up display of the Orbital Parameters Panel.

Orbital Parameters		
1	Altitude (km)	500
2	Eccentricity (0-1)	0
3	Inclination (deg)	0
4	Arg. Peri. (deg)	0
5	RAAN (deg)	0
6	nu (deg)	0

Figure 5-21: Scenario Tab: Orbital Parameters Panel

1. **Altitude** - Altitude (km) is the maximum distance of the satellite from the earth's surface (max altitude occurs at apogee for an elliptical orbit). The ATMT model requires the altitude to be greater than 100 km, which is reference distance for the edge of earth's atmosphere and the beginning of space, and may be as large as desired. The model does not independently verify that the satellite remains above the earth's surface so the user must select an altitude and eccentricity that ensure that the satellite remains in space.
2. **Eccentricity** - Eccentricity of the orbit (ϵ) is unitless and can from ($0 < \epsilon < 1$) with 0 denoting a circular orbit and all values less than 1 indicating an eccentric orbit. ATMT is not configured to analyze parabolic or hyperbolic orbits.
3. **Inclination** - Inclination (i) is the angular distance (deg) of the orbital plane from the equatorial plane of the central body (Earth). Some models use the ecliptic plane as the point of reference with orbital plane a distance beta (β) from the ecliptic. Inclination ranges from ($0^\circ \rightarrow 180^\circ$). For reference, the Sun's ecliptic plane is inclined $\sim 23^\circ$ from Earth's equatorial plane.
4. **Argument of Perigee** - Argument of perigee (ω) is the angle (deg) from the ascending node to perigee measured along the orbital plane in the direction of flight. Argument of perigee ranges from ($0^\circ \rightarrow 360^\circ$).

5. **Right Ascension of the Ascending Node** - Right ascension of the ascending node (RAAN or Ω) is the angle (deg) from the principal direction (I) to the direction of the ascending node measured along the orbital plane in the direction of flight. Right ascension of the ascending node ranges from ($0^\circ \rightarrow 360^\circ$).
 6. **True Anomaly** - True anomaly (ν) is the angle (deg) from perigee to the position of the satellite measured in the direction of flight. True anomaly ranges from ($0^\circ \rightarrow 360^\circ$).
- **Scenario Panel** - The Scenario Panel allows the user to define the starting and ending date and time for the simulation. These times are used in the Julian Date algorithm to determine the position of the sun. Furthermore, the difference between start and stop defines the simulation runtime. The date and time must be input as year, month, day, hour, minute, second for both the start and finish time. Clearly the finish date and time must occur after the start time in order for the scenario to run. Figure 5-22 shows a close up display of the Scenario Panel.

The image shows a 'Scenario' panel with two columns: 'Start' and 'Finish'. There are six rows of input fields, each with a numbered callout box (1-6) to its left. The rows are: Year (1900-2100), Month (1-12), Day (1-28), Hour (0-23), Minute (0-59), and Sec (0-59). The values entered are: Start Year: 2010, Finish Year: 2010; Start Month: 1, Finish Month: 1; Start Day: 1, Finish Day: 2; Start Hour: 0, Finish Hour: 0; Start Minute: 0, Finish Minute: 0; Start Sec: 0, Finish Sec: 0.

Scenario		Start	Finish
1	Year (1900-2100)	2010	2010
2	Month (1-12)	1	1
3	Day (1-28)	1	2
4	Hour (0-23)	0	0
5	Minute (0-59)	0	0
6	Sec (0-59)	0	0

Figure 5-22: Scenario Tab: Scenario Panel

1. **Year** - The start and finish year must be input as an integer between 1900 and 2100¹.
2. **Month** - The start and finish month must be input as an integer between 1 and 12.
3. **Day** - The start and finish day must be input as an integer between 1 and 31².

¹The algorithm is good for dates between 1 Mar 1900 to 28 Feb 2100.

²Be aware not all months will allow a range of 1 \rightarrow 31 days.

4. **Hour** - The start and finish hour must be input as an integer between 0 and 23.
 5. **Minute** - The start and finish minute must be input as an integer between 0 and 59.
 6. **Second** - The start and finish second must be input as an integer between 0 and 59.999.
- **Initial Orientation Panel** - The Initial Orientation Panel allows the user to define the satellite's initial orientation in space as well as specify any rotations the satellite will undergo throughout the simulation. Figure 5-23 shows a close up display of the Initial Orientation Panel.

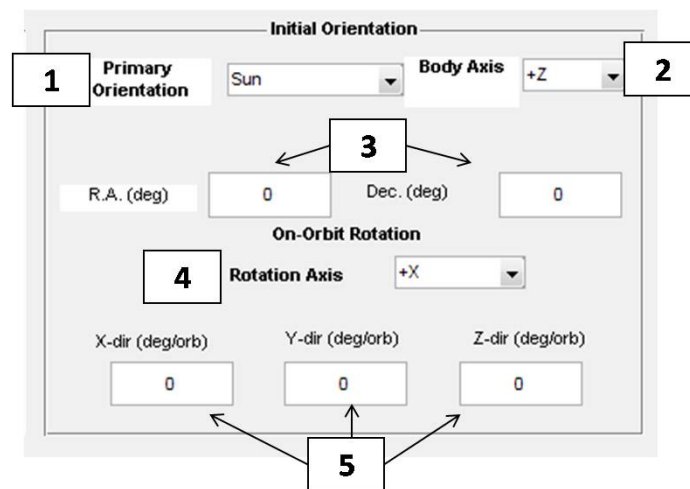


Figure 5-23: Scenario Tab: Initial Orientation Panel

1. **Primary Orientation** - In order to determine the satellite position in inertial space, the user must choose a primary orientation for the satellite. The list offers four basic orientations: sun facing, nadir facing, velocity direction, and right ascension and declination. These four orientations are described in Section 2.3.2.
2. **Body Axis** - The user must specify which body axis (+X, +Y, or +Z) is to be aligned with the selected primary orientation. If +X is chosen the +Z direction will be set as the second direction with +Y completing the right-handed coordinate system. If +Y or +Z is chosen, +X will take the second direction. For a sun or nadir facing orbit, the second direction is aligned in the orbital plane in

the direction of velocity (only along the velocity vector for circular orbits). For a velocity facing orientation, the second direction is set in the orbital plane in the nadir direction (only along the nadir vector for circular orbits). Finally, for a right ascension and declination specific orbit, the second direction is placed in the velocity direction.

3. **Right Ascension and Declination** - Right ascension and declination (R.A. & Decl.) relate the attitude of the satellite to the sun (deg). The R.A.&Decl. orientation is used only if it is selected as the primary orientation.
4. **Rotation Axis** - The rotation axis can be selected if the user wishes to impart a rotation of the satellite about either the +X, +Y, and +Z body axis.
5. **Rotation Rates** - The “Rotation Rate” feature is used if the user wishes to rotate (deg/orbit) about a particular axis.

- **Orbiting Body Panel** - The Orbiting Body Panel allows the user decide which body to use in the two-body problem. Orbiting body parameters have a significant effect on the thermal analysis and should only be tweaked with caution. Figure 5-24 shows a close up display of the Orbiting Body Panel.

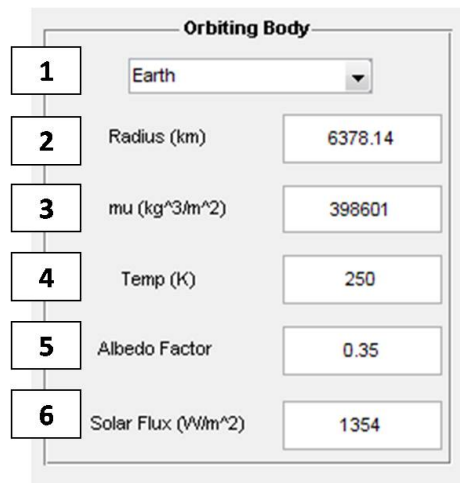


Figure 5-24: Scenario Tab: Orbiting Body Panel

1. **Body** - The “Body” is the central object about which the satellite rotates. The default is Earth and the current version of ATMT only supports Earth as the

central body. Future versions will include Earth's Moon, the sun, and Mars. The default body is Earth.

2. **Radius** - The radius of the body is measured in kilometers (km). The default Earth radius is 6378.14 km.
 3. **Gravitational Parameter** - The gravitational parameter (μ) of the body is measured in (km^3/sec^2). The default gravitational parameter for Earth is 398601 (km^3/m^2).
 4. **Temperature** - The temperature (K) of the body is the constant temperature seen by the satellite in space. The body's temperature will vary depending on the orbital parameters and seasonal conditions however; ATMT uses a constant temperature approximation. The default value for Earth is 250 K.
 5. **Albedo Factor** - The albedo factor (AF) is the fraction ($0 \rightarrow 1$) of incident solar energy that is reflected by the central body. The default AF for Earth is 0.35 and may vary from $\sim 0.4 \rightarrow \sim 0.2$ for the hot and cold cases. The hot and cold cases may vary based on the orbital parameters as well as inner and outer atmospheric weather conditions.
 6. **Solar Flux** - The solar flux (G_s) is a measure of the incoming solar electromagnetic radiation per unit area measured in (W/m^2). The solar flux is input with a default average value of 1354 W/m^2 but will vary from $\sim 1428 \rightarrow \sim 1316 W/m^2$ for the hot and cold cases. The hot and cold cases may vary based on the orbital parameters and the solar cycle.
- **Simulation Data File Panel** - The Simulation Data File Panel allows the user to choose between running the scenario from the ATMT GUI data or loading a preformatted excel file containing all of the necessary model inputs. This feature can be incredibly useful for users who prefer to work with excel or for file sharing with users who do not have the required MATLAB licenses. Figure 5-25 shows a close up display of the Simulation Data File Panel.

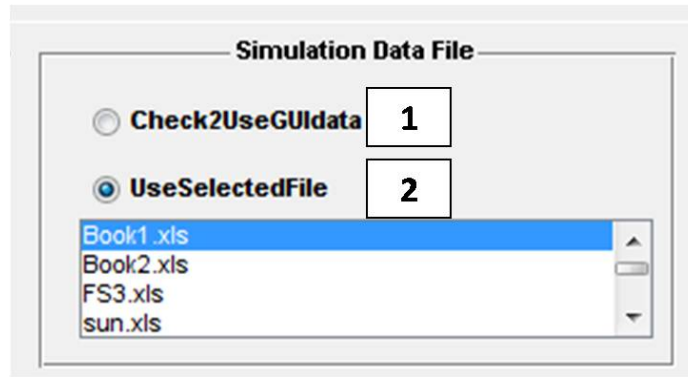


Figure 5-25: Scenario Tab: Simulation Data File Panel

1. **Use GUI Data** - The “Use GUI Data” radio button is the default option which causes ATMT to run a simulation from the current GUI data. Checking the “Use GUI Data” button will automatically uncheck the “Load Excel Data” radio button and hide the Excel file list.
 2. **Load Excel Data** - The “Load Excel Data” radio button allows the simulation to run from a preloaded Excel file that is chosen by the user. Checking the “Load Excel Data” button will make the Excel file list appear and will automatically uncheck the “Use GUI Data” button.
- **Running the ATMT GUI** - The final step of the process is to run the simulation. Several features are offered to assist the user in running the simulation and analyzing the results. Figure 5-26 shows a close up display of items involved in running the ATMT simulation.



Figure 5-26: Scenario Tab: Running the ATMT GUI

1. **Step Size** - The user must select a step size (Δt) for the simulation. The step size (seconds) is used to determine the number of steps the simulation will use for thermal data calculation. The step size has a direct effect on the runtime. Shorter step sizes will require more loops through the model resulting in a longer simulation runtime. The default step size is 100 seconds. The step size has been tested successfully for values ranging 1 \rightarrow 1200 seconds. The step size may be automatically modified during a simulation if the thermal stability criteria are not met for the selected step size. If the temperature calculations lead to a diverging situation, the step size is scaled back by 10 % per loop until the stability criteria are met. This model check ensures the simulation executes properly.
2. **Run** - The “Run” button simply executes the simulation. The button is green before and after running the simulation and will be red during the simulation. A “Percent Complete” timer refreshes as the model progresses toward completion.
3. **Save Data** - The “Save Data” button appears at the end of a successful simulation allowing the user to save the current state of the GUI with the model results for future use. After the save data command is executed the button will turn green and the caption will read “Saved”. The saved file can be viewed in the “Use Selected File” list from Figure 5-25. The saved file name is appended a “_sim_file” ending to distinguish it from the excel files in the list.
4. **Run Log** - The Run Log captures several pertinent features from the simulation. It will store the start and finish dates, runtime, and number of time steps. The previous simulation’s Run Log is automatically loaded when ATMT is first loaded.

5.8 Output Tab

The ATMT Output Tab is potentially the most important part of the ATMT program. Here the results are stored and may be visualized for analysis. The Output Tab has the capability to display multiple types thermal data outputs such as the external heat fluxes (EHFs) on a particular surface, transient temperature profile, internal radiation and conduction as well

as the input heat. Excel data files may be loaded from Saved Items subfolder and analyzed using the ATMT Output Tab. Data comparisons are done using the “Overlay Model/Data” feature which allows the user to compare the ATMT model results to models from other software packages, or to on-orbit and TVac test data. Figure 5-27 shows the Output Tab with all of its built-in features.

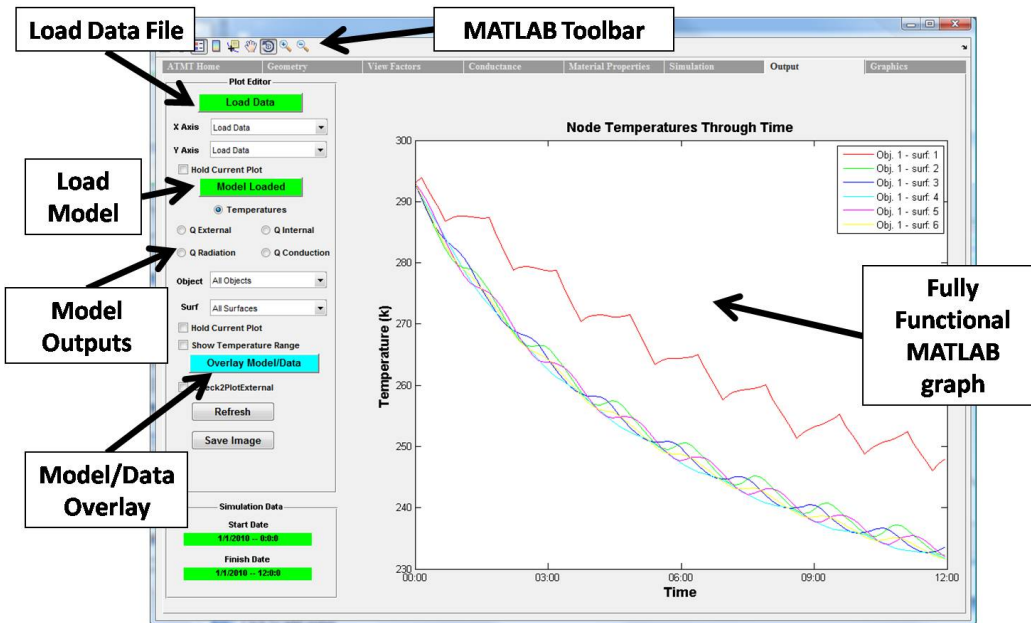


Figure 5-27: ATMT Output Tab

- **Output Plot Editor** - The Output Plot Editor contains all of the tools necessary to post process model results, analyze external data files, or overlay various sets of data for comparisons. Figure 5-28 shows the Output Plot Editor.

1. **Load Data** - The “Load Data” button prompts the user to select the desired “.xls” or “.xlsx” file to be loaded into the ATMT GUI. These Excel files must have a specific layout, namely the first row should include the desired column names and the cells below such contain numerical data. Files deviating from this format will produce an error when loading.
2. **Data Axes** - The X and Y axes drop-down menus allow the user to select which column of data should be plotted on each axis.
3. **Load Model** - The “Load Model” button updates the Output Tab with the

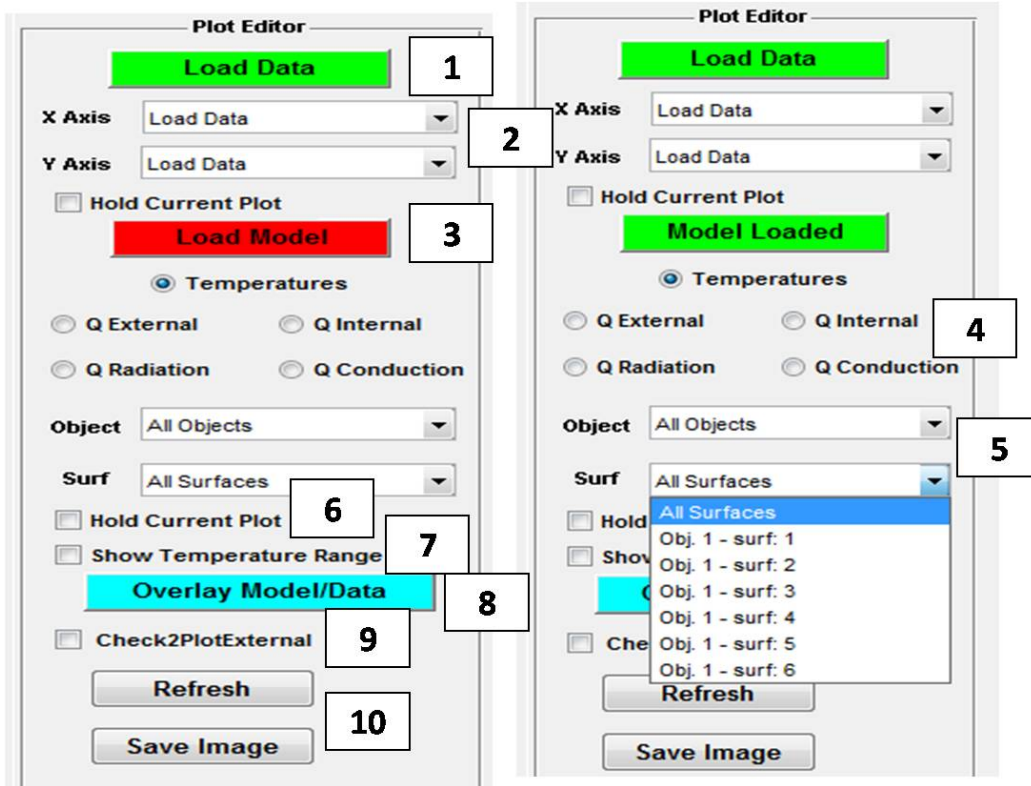


Figure 5-28: Output Tab: Output Plot Editor

most recent simulation results. After a simulation the “Load Model” button is red indicating that the button must be pressed in order to load the current simulation outputs. The button turns green with the “Model Loaded” caption once the object and surface drop-down menus have been refreshed.

4. **Model Outputs** - There are five types of outputs: temperatures, external heat fluxes, internal heat, radiation, and conduction. Selecting a radio button makes it the current choice for the output plot. Only one option may be selected at a time and the default option is temperature.
5. **Output Selection (Objects and Surfaces)** - The output selection feature allows the user to choose which object or surfaces should be analyzed. 2D simulations will have only the “Object” list because each object is also a single surface. The user may choose to plot the results for all objects simultaneously or for each object individually. For 3D models there is a “Surface” drop-down menu containing a list of all of the surfaces in the model. Once an object is selected

from the “Objects” drop-down the “Surface” drop-down will be refreshed with only the surfaces for that object. Again the user may choose to plot all of the surfaces on that object or may select each surface individually.

6. **Hold Current Plot** - The “Hold Current Plot” checkbox allows the user to overlay the model results for multiple objects or surfaces. Once the first selection is made, the user may check the “Hold Current Plot” box and change the output type if desired. The model output for the next object or surface that is selected will be plotted over the first on the same X axis time scale.
7. **Show Temperature Range** - The ultimate goal of a simulation is to determine if the results stayed within the required temperature range. When the “Temperatures” radio button is selected, a checkbox appears allowing the user to plot the temperature range for a desired object or surface. The user must first check the “Show Temperature Range” box and then choose the object or surface they wish to plot. Solid horizontal lines will appear on the graph at the object’s temperature range values which were selected previously in the Geometry Tab.
8. **Overlay Model/Data** - The “Overlay Model/Data” button will plot the ATMT selected model overlaid with the preloaded X and Y axis data. All three of the axes lists must have a selection in order for the overlay button to be active.
9. **Plot External** - The “Plot External” feature is identical to that of the Geometry Tab and allows the user to work with the plot in a figure window outside of the GUI. The figure should be closed by unchecking the plot external checkbox. This feature is especially useful for saving the plotted image with its active legend and axes labels. An example of an external output plot for a 2D six sided box is shown in Figure B-10 of Appendix B.
10. **Refresh and Save Image** - The “Refresh” feature clears the plot as well as all of the drop-down menus. The drop-down menus may be reloaded by simply pressing the “Load Data” or “Load Model” buttons without having to re-run the simulation. The “Save Image” feature saves the current Output Plot as a “.jpeg” file in the Saved Items subfolder.

Chapter 6

Conclusion and Future Work

6.1 Thesis Summary

This thesis presented a new approach to small satellite thermal modeling called the Adaptive Thermal Modeling Architecture (ATMA). ATMA was motivated by the author's experience with the U.S. Air Force Academy's FalconSAT-5 small satellite and MIT's CASTOR nanosatellite. Spacecraft thermal modeling is a critically important and complex endeavor. However, small satellite programs are often forced to forgo thermal modeling due to time, budget, and personnel constraints. ATMA offers a three phase approach to the development, validation, and implementation of satellite thermal modeling.

The Adaptive Thermal Modeling Tool (ATMT) was designed to fill the gap between the commercially available modeling tools and the satellite specific algorithms to perform effective thermal modeling at the small satellite level. Chapter 3 presented the development and verification of ATMT. Unique algorithms were designed and applied to the view factor (VF) and conductance sections of the model reducing the computation time by auto generating both the VF and Conductance Matrices. Chapter 4 addressed the validation and implementation of ATMT. The ATMT program enables on-orbit performance modeling of a satellite's thermal control system. A variety of test cases demonstrated ATMT's ability to perform on-orbit thermal trade studies as well as model-data correlation. Finally, Chapter 5 offered a complete ATMT User's Manual with examples of various modeling techniques and applications.

6.2 Contributions

The main contributions of this thesis are summarized below:

1. Performed a gap analysis on the currently available spacecraft thermal analysis tools. The gap analysis identified the two main categories of modeling tools: commercially available industry standard software such as Thermal Desktop or Abaqus, and satellite specific thermal algorithms such as the Spacecraft Control Toolbox and SatTherm. The thermal modeling requirements for a small satellite program were used to develop ATMT.
2. Created ATMA as a new approach to small satellite thermal modeling. ATMA includes the central ATMT program, model-to-model and model-data correlation using Thermal Desktop, and the implementation of ATMT to design and model an effective thermal control system.
3. Developed ATMT for use as an early design phase thermal modeling tool for small satellite programs. ATMT offers a graphical user interface for creating models, running simulations, and post processing data. ATMT has 6 main tabs that allow the user to customize their model and perform iterative trade studies.
 - (a) **Geometry Tab** - Includes all of the object parameters and CAD model visualization features.
 - (b) **View Factors Tab** - Automatically generates the VF Matrix and applies a truncation value to minimize unnecessary computations.
 - (c) **Conductance Tab** - Automatically generates the Conductance Matrix using a specified absolute conductance and separation tolerance.
 - (d) **Material Properties Tab** - Includes a database of the material properties and surface finishes for common aerospace applications.
 - (e) **Simulation Tab** - Allows the user to either manually customize or automatically load an on-orbit simulation. The Simulation Tab includes the orbital parameters,

scenario, orbiting body, initial orientation and simulation step size which are used to generate an on-orbit thermal model.

- (f) **Output Tab** - Offers data post processing and visualization tools. Model results may be overlaid with existing data or plotted against component temperature ranges to evaluate the performance of a thermal model.
4. Developed and implemented the VF and Conductance Matrix algorithms. These algorithms eliminate the user's responsibility to manually calculate the VFs and conductances between each pair of surfaces. The algorithms accurately calculate and populate the VF and Conductance Matrices for use in the thermal solver.
 5. Utilized a version of the ray-polygon intersection algorithm to determine if an object is in view, shaded, or blocked by other objects. This algorithm is being integrated into ATMT for two applications: determining if an object is shielded from external heat sources and improving the VF Matrix algorithms.

6.3 Future Work

The thermal modeling techniques and processes utilized in ATMA can be applied to a wide range of small satellite applications. However, ATMA is certainly not the only approach to thermal modeling and there is room for improvement. The future work tasks discussed in this section would strengthen ATMA and increase the functionality of the ATMT thermal modeling program.

- **Validation with On-orbit Data** - Chapter 3 discussed the importance and challenges of validating a thermal model. Model validation is done by correlating the results with actual test data as well as other trusted models. Chapter 4 presented four test cases as well as their correlation with an industry standard thermal modeling program (Thermal Desktop). The first future work task is to validate the ATMT program against actual on-orbit thermal test data. Model-data correlation will determine ATMT's true capabilities and limitations.

- **Variable Parameters** - All of the ATMT model parameters are initialized and fixed prior to running a simulation. In reality, parameters such as the input power and satellite attitude fluctuate with the operational scenario and phases of a mission. The ability to load a dynamic input power and attitude file would allow ATMT to model real-life operational scenarios.
 1. **Power Profile** - The input power to a component is rarely a fixed value. Instead most components operate at various power settings such as standby, nominal, and peak power. Two options would improve the input power feature of the model. The first option is to load and run an input power file containing the component power settings at each time step. The second option is to specify component power settings for various modes of operation (i.e. separation, commissioning, normal operations, and decommissioning).
 2. **Attitude** - The ATMT model requires the user to initialize the satellite's attitude parameters prior to a simulation. The option to load and run an input attitude file would allow ATMT to model an exact sequence of on-orbit attitudes.

- **Thermal Vacuum and Acceptance Test Modeling** - The primary objective of ATMA is to address the thermal modeling requirements for all phases (conceptualization through on-orbit operations) of a small satellite program. These phases require on-orbit, thermal vacuum, and acceptance test thermal modeling. The current version of ATMT is specifically designed for modeling the satellite's on-orbit temperature response. Two additional modules are necessary to model thermal vacuum and acceptance testing.

- **3D Temperature Averaging Algorithm** - ATMT offers the ability to model both 2D and 3D objects in space. The 2D option models each object as a single cross sectional surface with one node at the surface's centroid. The 3D option models each object as a combination of all outer surfaces with a node at the centroid of each surface. The trade off is accuracy for simulation runtime. Currently the 3D solver does not utilize all of the nodes for 3D model simulations. A future version of the

3D solver will use a weighted average temperature for all nodes keeping the object isothermal at each time step. This method improves the accuracy of the temperature measurement by incorporating the net heat transfer occurring for multiple nodes per object.

- **3D VF Shading Function** - The VF algorithm used in ATMT is designed to find and calculate the VFs between every pair of surfaces in the model. Shading and blockage presents a complex problem for the VF algorithm. Currently the algorithm simply truncates a shaded or blocked VF to zero as opposed to finding the exact value. This is a good initial assumption because the majority of VFs in a 3D model are in fact negligible and shading or blockage further reduces their impact on the model's results. However, the model accuracy is certainly limited by this assumption. Future work on the VF algorithm should be focused on calculating the exact value for shaded and blocked VFs.

Appendix A

Tables

Table A.1: View Factor Calculation Results: Version 1

Test Case	# of Obj	# of VFs	Run Time (sec)	Avg. Calc Time (sec)
2D 6 sided Box	6	36	7.8	0.217
1 3D Box	1	36	1.74	0.048
10 2D Surfaces	10	100	11.9	0.119
4 3D Boxes	4	576	51.0	0.089
5 3D Boxes	5	900	85.0	0.094
6 3D Boxes	6	1296	169.0	0.130
10 3D Boxes	10	3600	373.3	0.104
CASTOR	18	14884	1952.0	0.131

Table A.2: View Factor Calculation Results: Version 2

Test Case	# of Obj	# of VFs	Run Time (sec)	Avg. Calc Time (sec)
2D 6 sided Box	6	36	7.8	0.217
1 3D Box	1	36	1.74	0.048
10 2D Surfaces	10	100	11.9	0.119
2 3D Boxes	2	144	3.4	0.024
5 3D Boxes	5	900	21.4	0.024
6 3D Boxes	6	1296	41.7	0.032
10 3D Boxes	10	3600	85.26	0.104
CASTOR	18	14884	907	0.061

Table A.3: Initial Model Data Correlation Results

	Time Constant (sec)	Steady State Temp (K)
Model (Sensor 3)	~ 150	358
Data (Sensor 3)	~ 1000	365
Model (Sensor 4)	~ 180	347
Data (Sensor 4)	~ 1200	353

Table A.4: Final Model Data Correlation Results

	Time Constant (sec)	Steady State Temp (K)
Model (Sensor 3)	~ 900	365
Data (Sensor 3)	~ 1000	365
Model (Sensor 4)	~ 1160	347
Data (Sensor 4)	~ 1200	350

Resistor Type	Resistor #	Resistance (ohms)	Power (W)
Engine Resistor	1	5	45.0
Engine Resistor	2	5	45.0
Engine Resistor	3	5	45.0
Engine Resistor	4	5	45.0
Engine Total		20.0	180.0
Fin #1			
Battery Box Resistor	6	75	12.2
Battery Box Resistor	7	75	12.2
Battery Box Resistor	8	75	12.2
Battery Box Resistor	9	75	12.2
Comm. Antenna Resistor	5	500	1.8
Comm. Antenna Resistor	11	500	1.8
MPPT Resistor	10	500	1.8
Fin #1 Total		1800.0	54.5
Fin #3			
Anode Converter Resistor	22	1000	0.9
Voltage Converter Resistor	18	1000	0.9
Voltage Converter Resistor	19	1000	0.9
Voltage Converter Resistor	20	1000	0.9
Voltage Converter Resistor	23	1000	0.9
Voltage Converter Resistor	24	1000	0.9
Comm. Antenna Resistor	17	500	1.8
Comm. Antenna Resistor	21	500	1.8
Fin #3 Total		7000.0	9.2
Fin #4			
Avionics Box Resistor	13	250	3.7
Avionics Box Resistor	14	250	3.7
Avionics Box Resistor	15	250	3.7
Avionics Box Resistor	16	250	3.7
Reaction Wheel Resistor	12	150	6.1
Fin #4 Total		1150.0	20.8
Cummulative Total		9970.0	264.5

Figure A-1: CASTOR Structural Engineering Model Resistor Spreadsheet

Appendix B

Figures

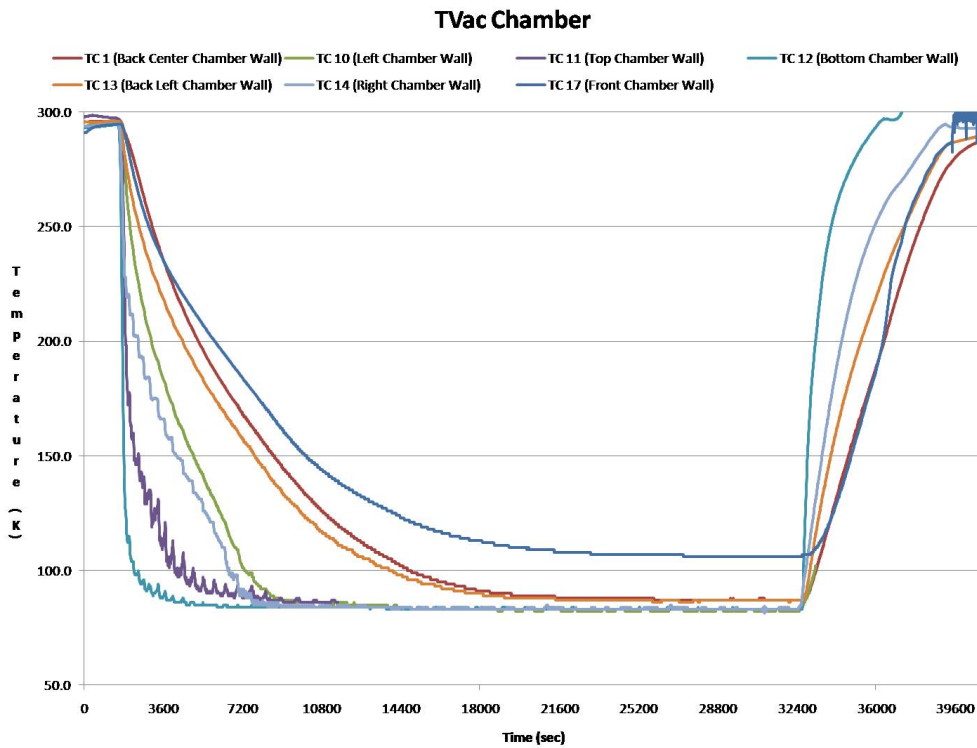


Figure B-1: TVac Chamber Internal Temperatures

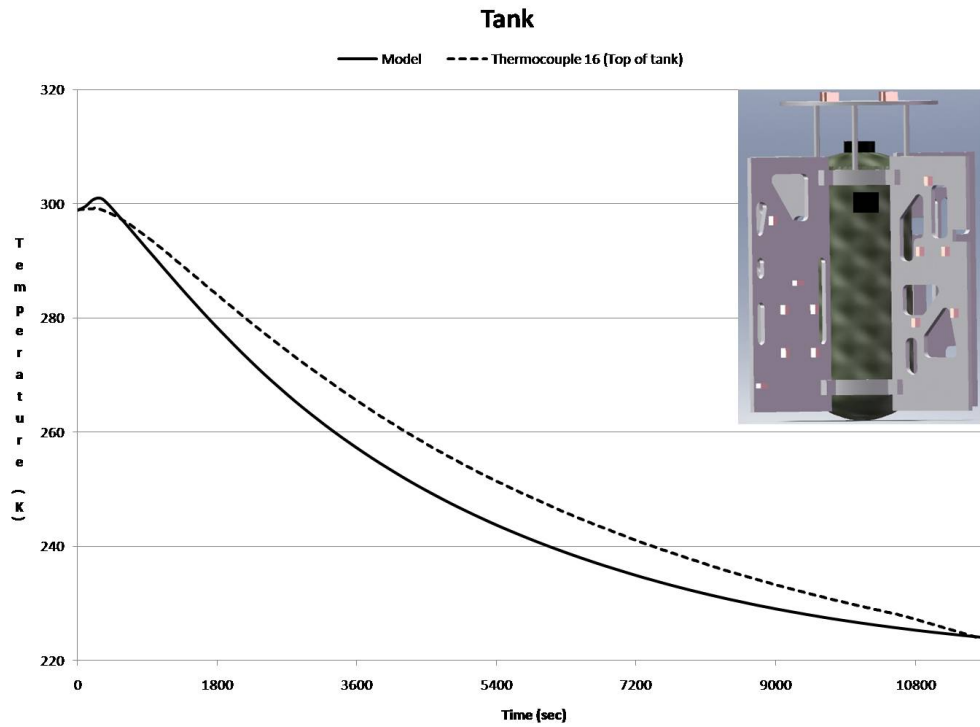


Figure B-2: Model Data Correlation for the CASTOR Xenon Tank

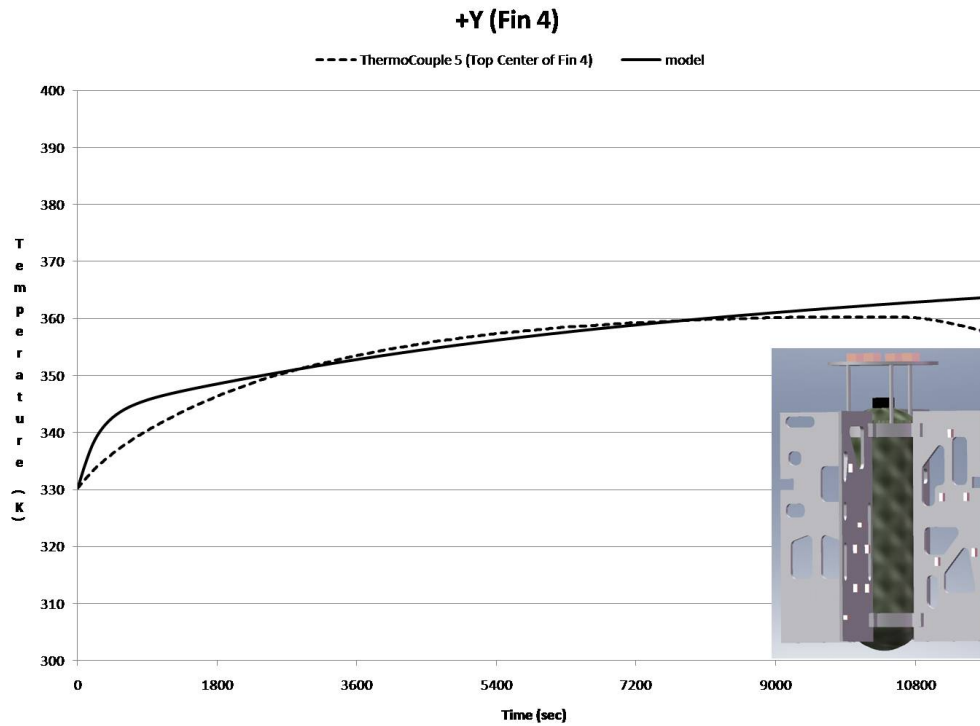


Figure B-3: Model Data Correlation for the CASTOR +Y Fin

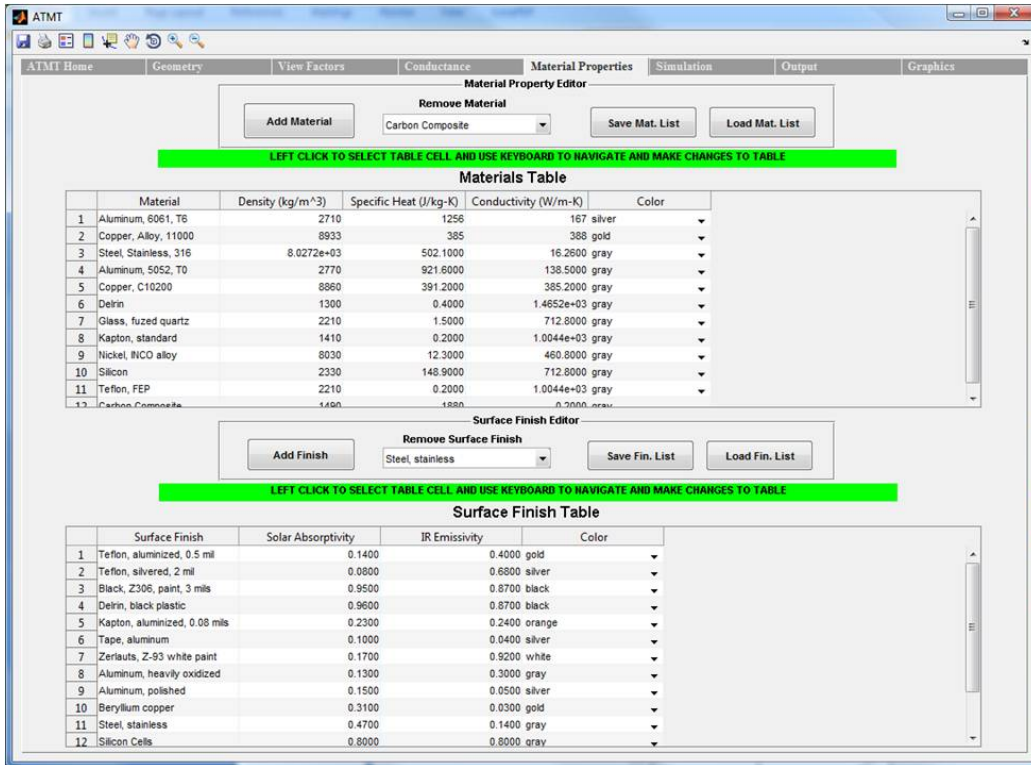


Figure B-4: ATMT Material Properties Database

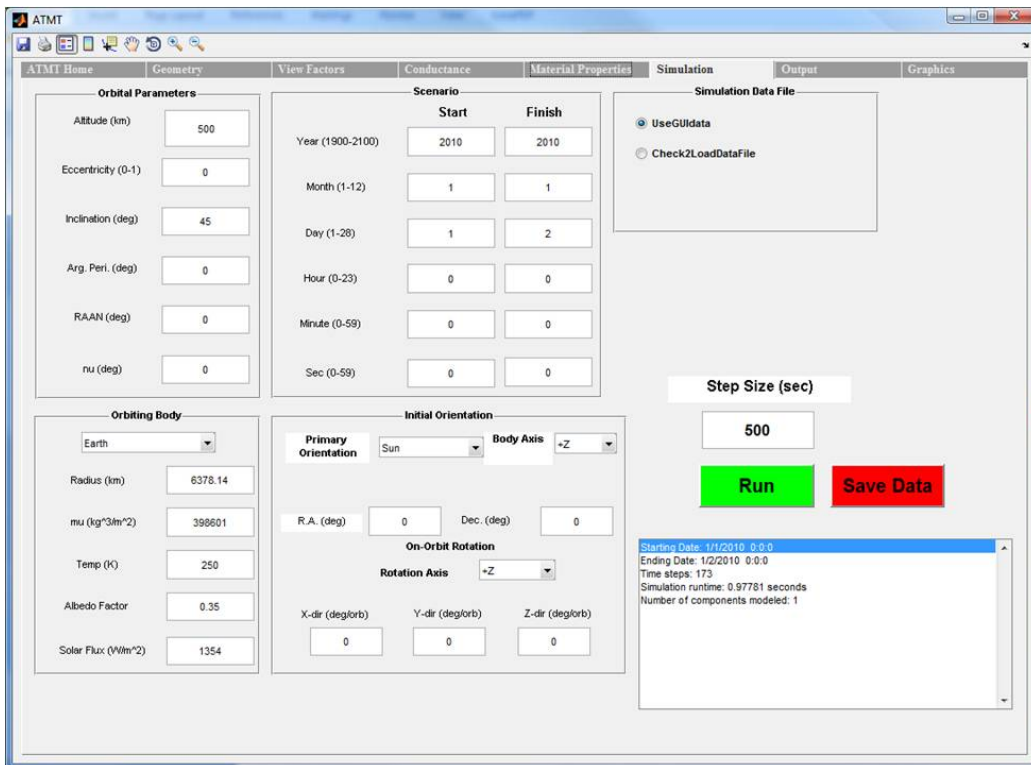


Figure B-5: ATMT Test Case Baseline Scenario

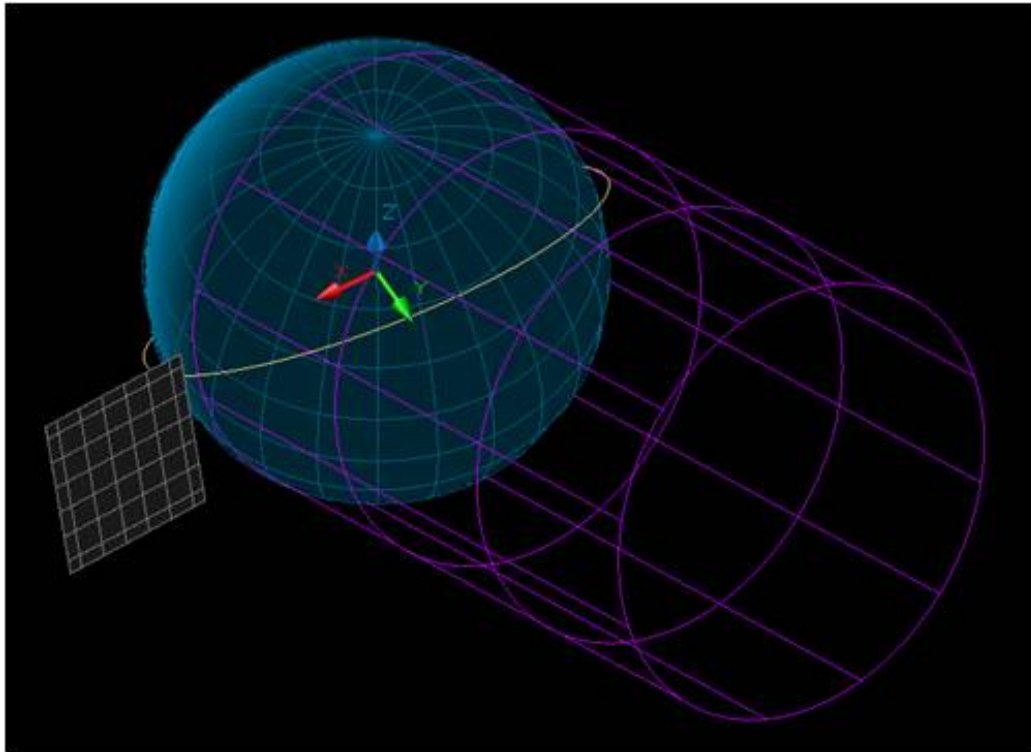


Figure B-6: Thermal Desktop On-orbit Display of a 2D Surface

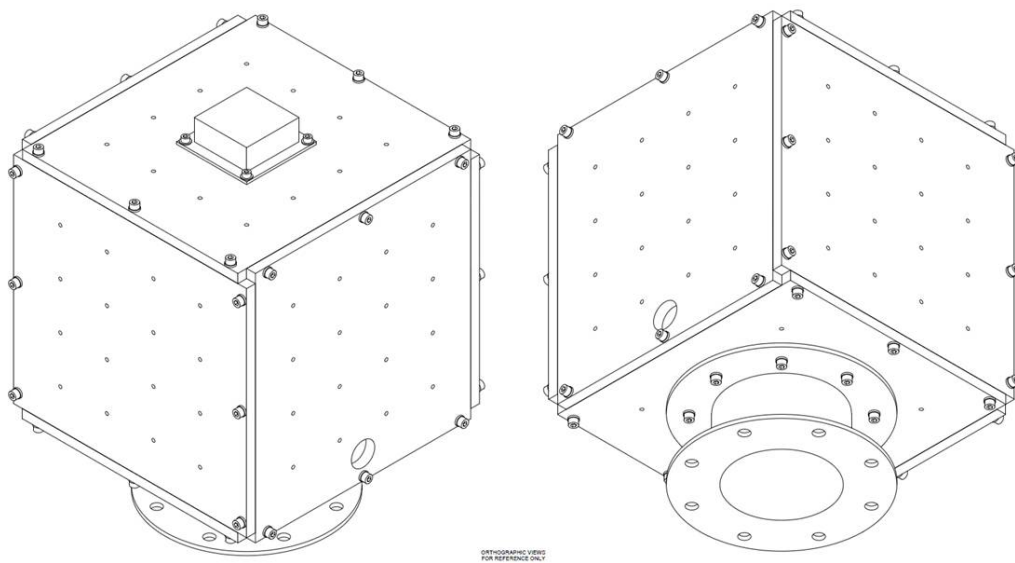


Figure B-7: Nanosatellite Thermal Test Bed

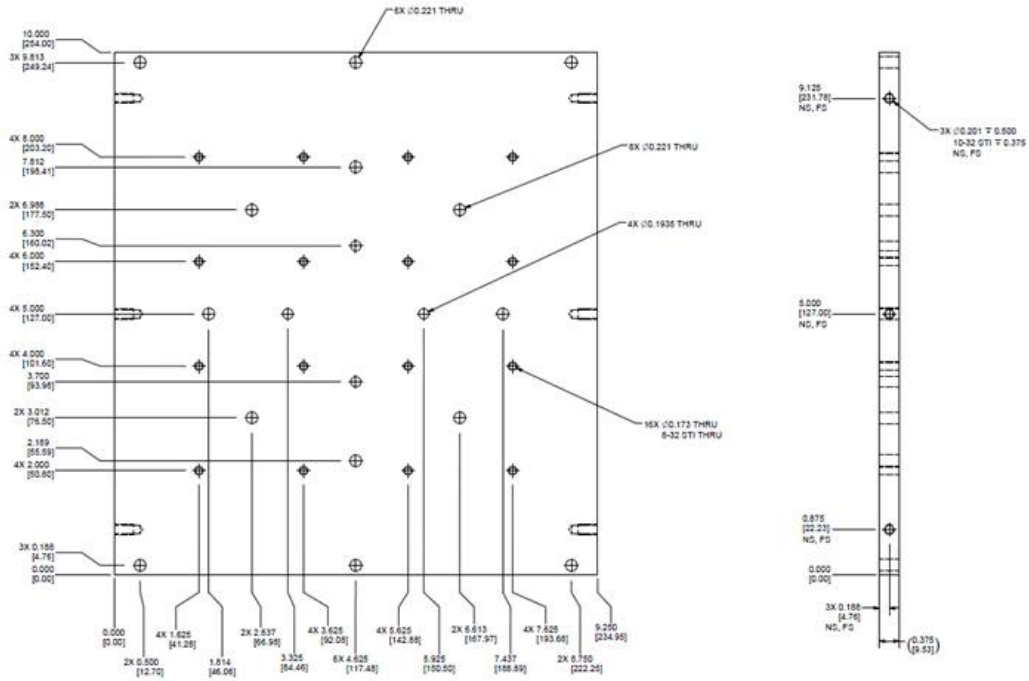


Figure B-8: Nanosatellite Thermal Test Bed: Side Panel Drawing

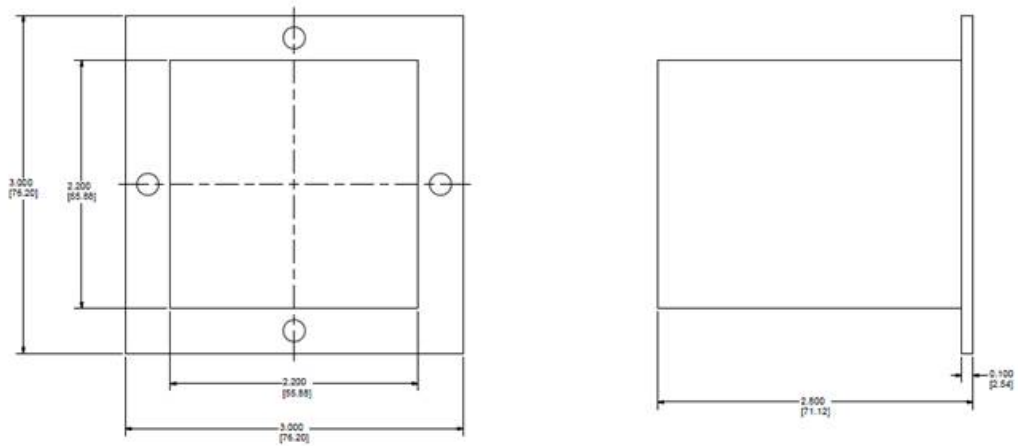


Figure B-9: Nanosatellite Thermal Test Bed: Avionics Stack Drawing

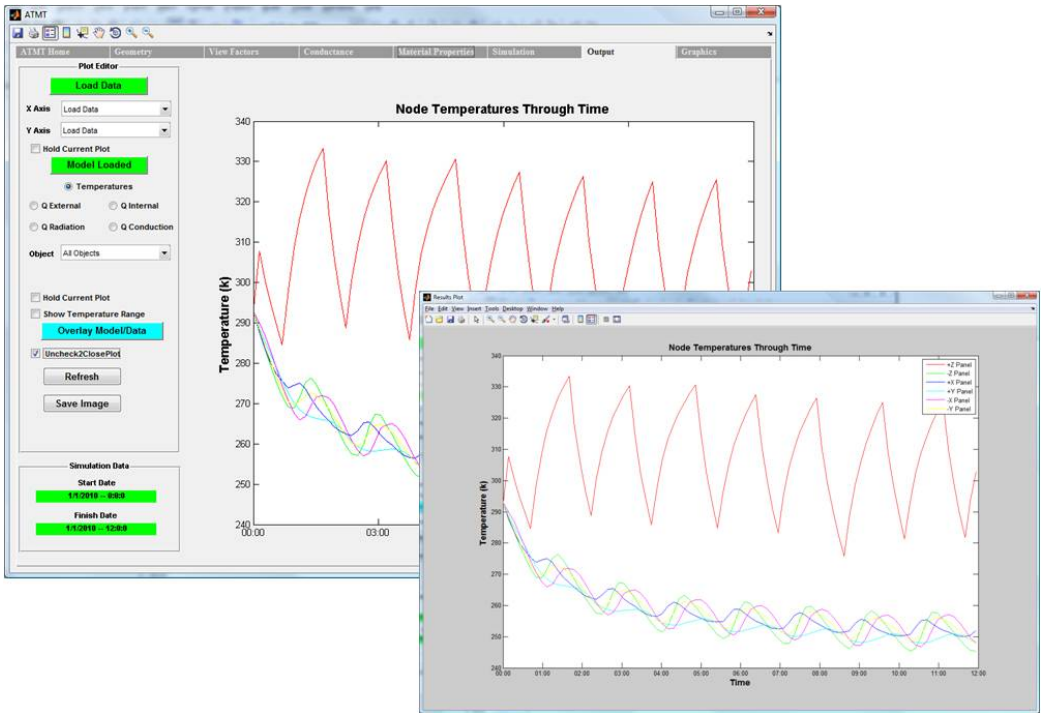


Figure B-10: ATMT Results Plotted External to the GUI

Appendix C

ATMT Source Code

C.1 ray_poly

```
function output=ray_poly(cent1,norm1,plane.eqns)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                               Function: ray_poly
%%
%% Copyright 2LT John Richmond 2010. All rights reserved.
%%
%% Purpose:  Determines whether or not a ray intersects a polygon and the
%%           point of intersection
%%
%% function  output=ray_poly(cent1,norm1,plane.eqns)
%%
%% Author/Date   : 2LT John Richmond, MIT, Spring 2010
%%
%% Inputs:
%%   cent1       - Centroid for the first surface           (x,y,z)
%%   norm1       - Normal vector to the first surface      (x,y,z)
%%   plane.eqns  - Plane equations for all surfaces
%%
%% Outputs:
%%   output      - Flag for intersection (1 = int, 0= no int)
%%
%% Locals:
%%   syms x y z t
%%
%% Constants:
%%   none
%%
%% Couplings:
%%   solve      - Internal MATLAB function to solve an equation
%%   subs       - Internal MATLAB function to substitute data
%%   double     - Internal MATLAB function to convert to double
%%
%% References:
%%   Andrew S. Glassner, editor. An Introduction to Ray Tracing. Morgan
```

```

%%      Kaufmann, 1989.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
syms x y z t real
% Define a plane (P)
P = [x,y,z];

% Load the coordinates of all vertices, the normal vectors to each surface,
% the component properties, and the number of surfaces for each component
coordinates=objects.coordinates;
normals_matrix=objects.normals_matrix;
components=objects.components;
sides_=cell2mat(objects.sides_);

% Select the plane equations for the object of interest
plane_eqns(1:sum(sides_))=x;

% Establish the plane equation for each surface
count=1;
for ind=1:components; % counter for coord1 component #
for k=1:sides_(ind)
norm1=normals_matrix(normals_matrix(:,1)==ind & normals_matrix(:,2)==k,3:5)
coord=coordinates(coordinates(:,1)==ind & coordinates(:,2)==k,3:5);
coord1=coord(1,:);
plane_eqns(ind)=dot(norm1, P-coord1);
count=count+1;
end
end;
T5 = cent1;
T6 = cent1+1000*norm1;
T7 = cent1-1000*norm1;
P = [x,y,z];
line(1) = T5 + t*(T6-T5);
line(2) = T5 + t*(T7-T5);

% Check if the normal vector from a reference surface intersects a
% comparison surface
for k=1:length(plane_eqns)
for j=1:length(line);
newfunction = subs(plane_eqn(k), P, line(j));
t0 = solve(newfunction);
point = subs(line(j), t, t0);
[value,dominant]=max(abs(norm1));
T(:,dominant)=[];
point(dominant)=[];
T(:,1)=T(:,1)-point(1);
T(:,2)=T(:,2)-point(2);
NumC=0;
t0=double(t0);
if (t0 >= 0 && t0 <= 1);
for i=1:length(T(:,1));
if i==length(T(:,1));
if (T(i,2)<0 && T(1,2)>=0) || (T(i,2)>=0 && T(1,2)<0);
if T(i,1)>0 && T(1,1)>0;
NumC=NumC+1;
elseif T(i,1)>0 || T(1,1)>0;
if T(i,1)-T(i,2)*(T(1,1)-T(i,1))>0;
NumC=NumC+1;
else

```

```

        end
    end
end
break;
else
if (T(i,2)<0 && T(i+1,2)≥0) || (T(i,2)≥0 && T(i+1,2)<0);
    if T(i,1)>0 && T(i+1,1)>0;
        NumC=NumC+1;
    elseif T(i,1)>0 || T(i+1,1)>0;
        if T(i,1)-T(i,2)*(T(i+1)-T(i,1))>0;
            NumC=NumC+1;
        else
            end
        end
    end
end
end
end
end
end
end
if mod(NumC,2);
    output=1;
    disp('point is inside polygon')
else
    output=0;
    break
    disp('no crossing')
end
end
end
end

```

C.2 Conductance

```

function conductance=Conductance(objects)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                     Function: Conductance
%%
%% Copyright 2LT John Richmond 2010. All rights reserved.
%%
%% Purpose: Determine the conductance between surfaces oriented in 3
%%           dimensional space.
%%
%% function  conductance=Conductance(objects)
%%
%% Author/Date : 2LT John Richmond, MIT, Spring 2010
%%
%% Inputs:
%%  objects    - Objects structure
%%
%% Outputs:
%%  conductance - Nsurfaces x Nsurfaces Conductance matrix containing
%%              the conductance value between each pair of surfaces (W/K)
%%
%% Locals:
%%  none
%%
%% Constants:
%%  none
%%

```

```

%% Couplings:
%%   none
%%
%% References:
%%   none
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the component properties, the number of surfaces for each component,
% the separation tolerance value, and the default conductance value
components=length(objects.name_);
sides_=cell2mat(objects.sides_);
tolerance=objects.tolerance/1000;
default=objects.default;

% Check to see if the separation tolerance is in the proper range
if tolerance<0
    error('error: tolerance value may not be less than 0 mm')
end
if default>200
    disp('Warning: Conductance value may lead to divergent temp results')
end

conductance=zeros(sum(sides_));
coordinates=objects.coordinates;

% Determine which component's sides are in contact
row=1;
for ind=1:components; %counter for coord1 component #
    for k=1:sides_(ind); %counter for coord1 surface #
        if ind==1;
            column=1;
        else
            column=sum(sides_(1:(ind-1)))+1;
        end
        for j=ind:components; %counter for coord2 component #
            for i=1:sides_(j); %counter for coord2 surface #
                if ind==j;
                    conductance(row,column)=default; conductance(column,row)=default;
                else
                    size1=size(coordinates(coordinates(:,1)==ind & coordinates(:,2)==k,3:5));
                    size2=size(coordinates(coordinates(:,1)==j & coordinates(:,2)==i,3:5));
                    test=0;
                    exit=0;
                    for l=1:size1(1);
                        for m=1:size2(1);
                            coord1=coordinates(coordinates(:,1)==ind & coordinates(:,2)==k,3:5);
                            coord2=coordinates(coordinates(:,1)==j & coordinates(:,2)==i,3:5);
                            if abs(norm(coord1(l,:)-coord2(m,:)))<tolerance;
                                conductance(row,column)=default; conductance(column,row)=default;
                                exit=1;
                                break;
                            else
                                conductance(row,column)=0; conductance(column,row)=0;
                            end
                        end
                    end
                    test=test+1;
                end
            end
            if exit==1;
                break;
            end
        end
    end
end

```

```

end
end
column=column+1;
end;
end;
row=row+1;
end;
end;

```

C.3 vf_sc_earth

```

function VF_ie = view_factor_scearth(h,gammar)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                               Function: view_factor_scearth
%%
%% Copyright 2LT John Richmond 2010. All rights reserved.
%%
%% Purpose: This function calculates the view factor VF_ie from a surface
%% to the Earth.
%%
%% function VF_ie = view_factor_scearth(h,gammar)
%%
%% Author/Date : Cassandra Belle VanOutryve, December 2008
%% modified by : 2LT John Richmond, MIT, Spring 2010
%%
%% Inputs:
%%   h      - Altitude of the surface above Earths surface (km)
%%   gammar - Angle between normal vector and nadir vector (radians)
%%
%% Outputs:
%%   VF_ie  - View factor of the surface to the Earth (0-1)
%%
%% Locals:
%%   none
%%
%% Constants:
%%   RE     - Radius of the earth (km)
%%
%% Couplings:
%%   none
%%
%% References:
%%   Vallado
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%RE=6378.14;
global RE;
rsc=RE+h; %distance of spacecraft from center of Earth (km)
H = rsc./RE;
phi_m = asin(1/H);
b = sqrt(H.^2-1);

%Case 1 - The surface sees all of Earth
if gammar <= pi./2-phi_m;
    VF_ie = cos(gammar)./H.^2;

%Case 2 - The surface sees part of Earth

```

```
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));
    VF_ie = 2./pi.*(pi./4-t1+t2);

%Case 3 - The surface does not see Earth
else
    VF_ie=0;
end
```

Bibliography

- [1] Douglas P. Bell, Brent A. Cullimore, David Johnson, Timothy D. Panczak, Steven G. Ring, and Mark J. Welch. *Thermal Desktop*. Cullimore and Ring Technologies, Inc., 5.2 edition, October 2008.
- [2] Roz Brown. Ball Aerospace Completes STPSat-2 Satellite Pre Shipment, October 2009.
- [3] Yunus A. Cengel. *Heat Transfer A Practical Approach*. McGraw Hill Higher Education, 2003.
- [4] A.J. Chapman. *Fundamentals of Heat Transfer*. Macmillan Publishing Company, 1987.
- [5] Planetary Systems Corporation. *2000785 Rev A User's Manual for Mark II Lightband*. 2008.
- [6] Space Vehicles Directorate. *Nanosat-6 User's Guide: University Nanosat-6 Program*. Air Force Research Laboratory, 1 edition, January 2009.
- [7] David G. Gilmore, editor. *Spacecraft Thermal Control Handbook*, volume I: Fundamental Technologies. The Aerospace Press, second edition.
- [8] Andrew S. Glassner, editor. *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989.
- [9] Leon R. Glicksman and John H. Lienhard. *Modelling and Approximation in Heat Transfer*. 1998-2006.
- [10] D.C. Hamilton and W.R. Morgan. Radiant interchange configuration factors. Technical report, National Advisory Committee for Aeronautics, 1952.
- [11] J.H. Henninger. *Solar Absorbance and Thermal Emittance of Some Common Spacecraft Thermal Control Coatings*. NASA/GSFC, nasa reference publication 1121 edition, 1984.
- [12] Nicolas Lauzier and Daniel Rousse. viewfactor.m. *Université Laval*, 2003.
- [13] Omega. Thermocouples - An Introduction. Omega Engineering Technical Reference.
- [14] Princeton Satellite Systems. *Spacecraft Control Toolbox v8 User's Guide*, 1997-2009.
- [15] Justin Ray. Milstar Satellite Launched in First Space Shot of 2002, January 2002.
- [16] Tatjana K. Rosev. Cube Sats Revolutionize Space Science, November 08.
- [17] Markus Schelkle. Small Satellite Design Thermal Control. Technical report, Institut Fur Raumfahrtssysteme: EADs Astrium, 2010.

- [18] Jerry Jon Sellers. *Understanding Space*. McGraw-Hill Companies, Inc., revised second edition, 2004.
- [19] National Semiconductor. LM19 2.4V, 10 μ A, TO-92 Temperature Sensor. Technical report, 2007.
- [20] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. McGraw-Hill, 1972.
- [21] Lucas Snyder. Orbital Elements, October 2007. Reference to the Classical Orbital Elements image used Chapter 3.
- [22] Space Test Program. *DoD Space Test Program Secondary Payload Planner's Guide For Use On The EELV Secondary Payload Adapter*, 1 edition, June 2001.
- [23] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Space Technology Library, 2001.
- [24] Cassandra Belle VanOutryve. A Thermal Analysis and Design Tool for Small Spacecraft. Master's thesis, San Jose State University, 2008.
- [25] Oliver De Weck. *Fundamentals of Systems Engineering: Verification and Validation*. MIT Stellar, November 2009.