Reliable, Fast, and Flexible: A Thermal Modeling Approach for Small Satellites

Max Gulde,1 Javier Montemayor Mancías,1,2 Christopher De Genova,1,3 Martin Schimmerohn,1 Frank Schäfer1
1Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI
Ernst-Zermelo-Straße 4, 79104 Freiburg, Germany
+49 (0)761 2714 414 – max.gulde@emi.fraunhofer.de
2Monterey Institute of Technology and Higher Education, Campus Monterrey
Av. Eugenio Garza Sada 2501 Sur, 64849 Monterrey, Mexico
3University of South Carolina, Columbia Campus
301 Main Street, 29208 Columbia, SC, USA

ABSTRACT
The ongoing revolution of space access by means of cost-effective and highly performant small satellites, in particular CubeSats, drives the development of a vast host of new and increasingly complex applications. However, the use of small satellites for ambitious missions brings its own challenges with thermal breakdown as one of the key contributors to component failure. We have therefore developed a lightweight approach specifically tailored to the thermal modeling of small satellites to localize and mitigate the associated thermal risks while maintaining the flexibility and low resource footprint necessary to be applicable in the framework of small satellite mission design. At the core of the methodology, we implemented an experimental database of physical parameters as well as highly parallelized numerical analysis methods. In particular, we introduce an efficient way to determine view factors for isolation and internal radiative energy transport based on a hemicube radiosity algorithm. The results agree within 1 K with commercially available modeling software and allow us to perform highly reliable temperature predictions while conserving the flexible and cost-efficient spirit of small satellite missions.

INTRODUCTION
Out of the vast number of potential risks for spacecraft, thermal breakdown is widely considered to be one of the most dominant ones: Large thermal gradients lessen optical pointing accuracy, complicate sensor calibration, and can ultimately destroy electrical components such as batteries1,2. Therefore, we usually undertake huge efforts to precisely simulate the thermal environment for the mission and localize as well as mitigate potential risks in various operation states of the spacecraft.

With small satellites and in particular CubeSats becoming increasingly potent in terms of payload as well as mission intricacy3,4, the complexity of respective thermal analysis is continuously expanding. However, respective software tools were primarily designed for conventional, large satellite missions and require not only expert knowledge but often also huge computational as well as financial resources. The need for suitable small satellite thermal modeling software arises because such requirements do not match well with the design spirit of small satellites5.

APPROACH
We base our approach on the premise that it should mirror the spirit of small satellites, in particular CubeSat mission design: Firstly, it should be flexible enough to allow swift adaption to design changes in the satellite or modified mission parameters. Furthermore, it should be resource-efficient and not rely on costly third-party software. Lastly, the solution must be very performant to allow for fast simulations of the whole system, e.g., in view of carrying out parameter studies. In particular, we propose a graphics processing unit (GPU)-based concept to allow for extremely fast view factor determination.

Figure 1: Schematic of the thermal modeling approach dependencies: Databases: yellow, models: orange, computations: blue, results: green.

Figure 1 depicts the general methodology of the thermal modeling approach as well as the notation we employ. In principle, we rely on three data input sources (yellow): (i) The actual geometry of the satellite, (ii) the satellite orbit definition, and (iii) a library of physical material properties.

In order to enable wide compatibility, we employ Wavefront® as an open geometry definition file format to define the satellite geometry. To define an orbit, we can...
either directly provide the respective orbital elements or use an existing TLE to propagate the satellite. Both, AGI’s System Tool Kit as well as the open source java library Orekit, are suitable to generate the required input.

At the heart of our approach, we have the material database. It works as a fundamental library for physical material properties, e.g., coefficients for absorption and emission for coatings, heat capacities, contact conductance, etc. The database rests upon literature values as well as experimental results from in-house thermal vacuum tests and is constantly refined and updated.

We broadly subdivide the approach into three computational steps: Firstly, we propagate the spacecraft. For TLE propagation, we use the SPG4, while the J2 perturbation propagator is employed when providing the orbital elements directly. Computationally, this step is relatively inexpensive and delivers the satellite attitude model needed for the second computational step – the view factor determination.

A view factor $F_{ij}$ is a purely geometrical quantity between two surfaces $i$ and $j$, which represents the ratio between the energy radiated from $i$ to $j$ to the total energy radiated by $i$.

In every thermal modeling software we know of, this step is by far the computationally most laborious one. Specifically, view factor calculation comes into play at two points: For the computation of insolation and related phenomena, we need to determine the view factor between each surface and the sun, taking into account self-shadowing effects and the momentary orientation of the surface. Similarly, for internal thermal energy transfer, the view factors between each two components inside the satellite have to be computed.

Subsequently, we merge the resulting view factor matrix with the functional model of the satellite to create the thermal model. Using a lumped capacitance network approach, the solution of the latter is performed inside the MATLAB programming environment. Notably, small satellites usually fulfill the assumption of homogeneous properties well, considering their reduced geometrical extent.

In a last step, we correlate the resulting node temperatures with the geometric and functional models. Critical thermal loads can now easily be located and mitigated by modifying the satellite geometry, material selection, or functional model.

**View factor determination**

While the first (propagation) and the last (thermal coupling) computational steps are performant enough to simulate the whole mission at high temporal resolution within few minutes at most, view factor determination presents the main computational bottleneck in designing a fast thermal modeling approach.

Since analytical view factor calculation is only possible for very specific cases, conventional thermal modeling software such as ESATAN-TMS employs Monte-Carlo Ray Tracing (MCRT) algorithms to determine the view factors$^2$: For each surface pair, some ten thousand rays are propagated to determine the visible area and relative surface orientation. With an increasing number of nodes, this process becomes computationally very demanding even after application of different geometrical optimization techniques and parallelization.

Here, we instead use concepts from 3D computer graphics to efficiently determine view factors with large accuracy. In particular, we employ orthographic projection techniques for insolation and the hemicube approach for radiosity computation between individual components.

**Figure 2: Orthographic projection of ERNST$^7$ geometric model for insolation computation with 1 mm$^2$ spatial resolution.**

Generally, a view factor $F_{S\to R}$ between a sender S and a recipient R is given by

$$ F_{S\to R} = \frac{1}{A_S} \int_{\Omega \in \mathcal{S}} \int_{\Omega' \in \mathcal{R}} \frac{1}{\pi r^2} \cos \phi_u \cos \phi_v V_d A_u dA_v, $$
with $A_s$ the surface area of the sender, $\phi_u$ ($\phi_v$) the angle between surface normal and connection vector for each pair of patches $(u,v)$ of sender and recipient, respectively, $r$ the distance between them, and

$$V = V(u,v) = \begin{cases} 1, & \text{if } u \text{ sees } v \\ 0, & \text{otherwise} \end{cases}$$

For the computation of insolation view factors, the sun can be seen as an infinitely distant point source with constant energy density and parallel moving photons. In this case, the incident thermal radiation is directly proportional to the projection of the surface in direction of photon propagation and can thus be computed by a simple orthographic projection (Figure 2) by the GPU of the computer.

Internally, the software computes a color histogram for each rendered image with each color assigned to a specific surface node. Simply by counting pixel colors in between render calls and even for thousands of nodes at once, a standard GPU can easily compute several hundred view factors per second, making it orders of magnitudes faster than MCRT.

For inter-node view factors, the methodology is slightly more complex. Firstly, we cannot assume a point source, but have to take into account a finite angular spread of the source. Additionally, not only the orientation of the receiving patch but also of relative orientation of the sender have to be taken into account. Lastly, the energy density decreases with the distance squared between the patches.

We hence adapted the methodology accordingly, implementing the energy-conserving radiosity approach more commonly used in the computation of realistic light distribution within computer graphics scenes.

Specifically, for each pair of sender and recipient, we draw the whole scene as seen by the recipient with a field of view of 180 degrees using perspective projection. Perspective projection scales down objects proportional to their distance to the viewer, hence automatically incorporates the $\frac{1}{r^2}$ dependency as required. Since field of views approaching 180 degrees are mathematically not trivially implemented, we instead use five 90 degree views stitched together to a hemicube.

For inter-node view factors, the methodology is slightly more complex. Firstly, we cannot assume a point source, but have to take into account a finite angular spread of the source. Additionally, not only the orientation of the receiving patch but also of relative orientation of the sender have to be taken into account. Lastly, the energy density decreases with the distance squared between the patches.

We hence adapted the methodology accordingly, implementing the energy-conserving radiosity approach more commonly used in the computation of realistic light distribution within computer graphics scenes.

Specifically, for each pair of sender and recipient, we draw the whole scene as seen by the recipient with a field of view of 180 degrees using perspective projection. Perspective projection scales down objects proportional to their distance to the viewer, hence automatically incorporates the $\frac{1}{r^2}$ dependency as required. Since field of views approaching 180 degrees are mathematically not trivially implemented, we instead use five 90 degree views stitched together to a hemicube.

We detail the approach schematically in Figure 3: We would like to compute the view factor between a source surface (S) radiating energy to a potentially geometrically complex panel (R) while part of the radiation is blocked by an intermediate object O. The scene is perspectively rendered in Figure 3a. First, we render the scene as seen by the source (viewing direction indicated by dot matrix) in five different directions to get the binary hemicube map (Figure 3b). In order to account for the angle of incidence as well as for perspective distortion, we multiply the resulting image by a transfer map (c). To ensure energy conservation, the transfer map is normalized. We can now compute the view factor by multiplying (b) and (c) and subsequent integration over the resulting image (Figure 3d). Until this point, the algorithm was performed exclusively on the GPU. To avoid performance bottlenecks by synchronizing CPU and GPU when transferring large amounts of image data, we can simply use MIP map textures averaged down to a single pixel. The final pixel value is in the range of zero to one and represents the view factor, i.e., the radiative energy fraction transmitted from source to target.
VALIDATION

To validate the hemicube approach independently, we compare the obtained numerical results with analytically solvable geometries. Figure 4 displays the results as well as the relative error for two cases: parallel plates of the same size (left panel) and parallel discs with a radius ratio of five (right panel). In both cases, we observe an overestimation of large view factors (e.g., at small distances) and an underestimation of small view factors (at large distances). The intermediate region displays a crossover of both curves and generally the best agreement. Taking into account the assumption of ideal Lambertian emission characteristics as well as the applied simple correction of perspective distortion, these results are in good agreement.

![Figure 4: Model results (solid lines) compared to reference model (dashed lines) for three nodes: optical bench (blue), payload (orange), outer RAM panel (yellow). Left: absolute temperatures. Right: temperature differences.](image)

To corroborate the applicability of the thermal modeling process as a whole, we validated a simple spacecraft against the commercial software package ESATAN-TMS. We specifically modified the spacecraft design to include all aspect of the thermal modeling process inside a complex space environment, including:

- insolation and albedo,
- ingoing and outgoing IR radiation,
- thermal conduction across material boundaries with a contact conductance value taken from our experimental database,
- radiative heat transfer between geometry nodes,
- outer and inner structure notes as well as nodes with both space-facing and inside-facing parts, and
- thermal loads from on-board payloads.

We depict the results for three representative nodes, the RAM facing structure panel, the payload, as well as an internal component, in Figure 5 for the duration of a single orbit (sunsynchronous, 700 km altitude).

![Figure 5: Model results (solid lines) compared to reference model (dashed lines) for three nodes: optical bench (blue), payload (orange), outer RAM panel (yellow). Left: absolute temperatures. Right: temperature differences.](image)

Generally, the ESATAN-TMS results (dashed lines in left panel, labeled “ref.”) display slightly higher temperatures and less pronounced amplitudes. However, our results (solid lines, left panel) have been computed with higher temporal resolution compared to the reference, which might account for the discrepancy.

Figure 5, right panel, displays the absolute temperature differences between our model and the reference. Our results (solid lines) display agreement within 1 K of the reference with a total root-mean-square deviation of 0.63 K.

Notably, the approach is performant enough to allow not only for simulation of specific points within a mission or a single orbit but can instead compute the transient solution over the whole mission time with few-second resolution within minutes, to some extent, of course, depending on the complexity of the satellite model. This will relieve users from pre-defining mission critical orbital positions and times and hence decrease the likelihood to overlook potential thermal breakdown scenarios.

The presented methodology will furthermore be employed in the thermal simulation of the nanosatellite ERNST currently in development at Fraunhofer EMI. In course of the validation, we plan to complement our physical parameter database in particular by experimentally measured parameters such as thermal expansion coefficients and thermal contact conductance between different materials such as aluminum, Scalmalloy® and others.\[12,13\]

CONCLUSIONS

We have demonstrated a highly adaptable and very performant approach to thermal modeling specifically designed to match the design spirit of small satellite missions. The methodology makes strong use of modern GPU parallel processing capabilities. In particular, we use a hemicube radiosity approach to substantially shorten the computation time for view factor determination compared to commonly employed raytracing methods. The resulting approach is

1. flexible to allow swift adaption to design modifications,
2. in good agreement with state-of-the-art thermal modeling software to allow for a reliable identification and mitigation of thermal risks,
3. highly performant compared to conventional MCRT methods, which enables users to quickly perform simulations of the whole system over the complete mission time, hence hugely contributing to facilitating reliable thermal modeling.
Future improvements of the methodology may include diffuse light reflection to increase the accuracy of the radiative transfer solution as well as adaptive resolution scaling and a paraboloid instead of a hemicube radiosity algorithm to increase the performance of the view factor determination even further.

In our view, these capabilities represent an ideal fit to the drastically reduced development times and planning resources of small satellite missions. Moreover, due to the current lack of adequate thermal modeling software for CubeSats, we believe it to serve as an additional accelerator of the current space revolution.

References
(7) Horch, C.; Schimmerohn, M.; Schäfer, F. Integrating a Large Nanosatellite from CubeSat Components – Challenges and Solutions. In 68th International Astronautical Congress (IAC); 2017.