Recent Improvements to the LAURA and HARA Codes

Kyle B. Thompson* Christopher O. Johnston† Brian R. Hollis‡ and Victor R. Lessard§
NASA Langley Research Center
Hampton, VA 23681, USA

This paper describes recent improvements to the LAURA and HARA codes. LAURA is a CFD code for aerothermodynamics, and HARA evaluates the shock-layer radiation that provides the radiative source term for the flowfield energy equations and radiative heating to a surface. The next release of LAURA and HARA includes a variety of new capabilities. These new capabilities include an automated uncertainty quantification workflow for radiative heat transfer, options for specifying surface roughness and turbulent transition location in the algebraic turbulence models, and improved grid and solution interpolation techniques. Additionally, the computational efficiency of both LAURA and HARA have been improved. Optimization of the MPI communication routines in LAURA are shown to improve the parallel efficiency of the primary flow when running with multiple processes per block, and recent optimization of HARA leverage graphics processing unit (GPU) acceleration in the radiation calculations. Using GPU acceleration of HARA is shown to decrease the cost of the radiation line-of-sight calculation by approximately one order of magnitude for a 10.5 km/s Earth entry simulation.

I. Nomenclature

C_H	=	heat transfer coefficient
$C_{H,FR}$	=	Fay-Riddell stagnation point heat transfer coefficient
f_{adjust}	=	radiative intensity correction factor
$J_{measure}$	=	radiative intensity measured in experiment
J_{sim}	=	predicted radiative intensity
$J_{sim,SCF}$	=	convolved radiative intensity
M_e	=	boundary-layer edge Mach number
Re_θ	=	boundary-layer momentum thickness Reynolds number
SCF	=	spatial convolution function
x'	=	spectral point
α	=	vehicle angle-of-attack

II. Introduction

This paper serves to document recent improvements to the Langley Aerothermodynamic Upwind Relaxation Algorithm [1] (LAURA) and HARA [2] codes, both of which have been used for decades as a computational fluid dynamics (CFD) tool to simulate high-energy, reacting flows. LAURA solves the Navier-Stokes equations using a cell-centered formulation on structured multi-block grid system. Formal second-order accuracy of the solution is achieved using the symmetric total variation diminishing scheme by Yee [3], and point integration for the source terms resulting from thermo-chemical non-equilibrium and several Reynolds-Averaged Navier-Stokes (RANS) turbulence models. The code includes a set of robust and state-of-the-art features for simulating phenomena occurring in high-energy flows, including shock-layer radiation, surface ablation, and species ionization. Radiation calculations are provided by the HARA radiation code, which is distributed with LAURA, and both are used in conjunction to accurately simulate

*Aerospace Engineer, Aerothermodynamics Branch, Member AIAA.

†Aerospace Engineer, Aerothermodynamics Branch, Associate Fellow AIAA.

‡Senior Technical Lead for Planetary Entry Systems, Aerothermodynamics Branch, Associate Fellow AIAA.

§Aerospace Engineer, Aerothermodynamics Branch, Senior Member AIAA.

radiative heat transfer.

Since inception, LAURA has been used in assessing the aerothermal environments experienced during entry, descent, and landing applications. During the development of the Mars Pathfinder vehicle, LAURA was used to predict two low angle-of-attack static instabilities in the vehicle's flight trajectory [4] into the Martian atmosphere, which were previously undiscovered during experimental testing. In support of the Space Shuttle Return-to-Flight effort, LAURA was used to perform computational aerothermodynamic simulations of the windside tile damage experienced by the Orbiter in flight [5]. Additionally, LAURA contributed to code verification in the FIRE-II [6] vehicle aerothermal analysis. Since then, LAURA has supported many NASA missions involving planetary entry. For earth re-entry, LAURA was used during the design and analysis of the heatshield for the Orion capsule [7]. Additionally, LAURA/HARA was recently used to demonstrate that radiative heat transfer was a significant contributor to afterbody heating for Earth entry capsules at velocities above 10 km/s [8]. For martian entry, LAURA was recently used in conjunction with the DPLR [9] code in the assessment of aerothermal environment for the Mars Interior Exploration using Seismic Investigations, Geodesy and Heat Transport (InSight) spacecraft [10]. For Jovian entry, LAURA/HARA demonstrated better predictions of the Galileo Probe entry heating as compared to previous computational models, using a coupled radiation and ablation approach [11]. Most recently, the LAURA code was also used to aid an uncertainty analysis of the stagnation-point calibration probe surface predictions for conditions that span the performance envelope of the Hypersonic Materials Environmental Test System facility located at NASA Langley Research Center [12].

Leveraging research and development obtained during the support of these applications, the next release of LAURA, Version 5.6, will include many improvements to simulating reacting flows. The first section will cover the new capabilities to the LAURA flow solver, which include better parallel performance, improved spline interpolation techniques, and new options for specifying turbulent transition location in algebraic turbulence models. The next section will cover improvements to the HARA code, which will also be distributed in the upcoming release. These HARA improvements facilitate faster radiative heating calculations, most notably that the HARA code has been made significantly more efficient when using hardware accelerators, due to the incorporation of OpenACC directives [13]. These performance gains are magnified by the implementation of a more robust and efficient ray-tracing capability that has been implemented into the LAURA code [14], along with an efficient multiband opacity binning approach [15]. Finally, a novel uncertainty quantification capability that will be delivered as part of the version 5.6 release is described. This new uncertainty quantification approach leverages the aforementioned performance gains to enable an automated approach to calculating radiative heating uncertainties that was prohibitively expensive in previous LAURA/HARA releases. All of these improvements contribute to the state of the art for simulating high-energy flows encountered in the aerothermal environments of entry, descent, and landing applications.

III. LAURA Improvements

This section covers the improvements made to the LAURA flow solver to improve parallel efficiency of the flow solver, provide more robust options for shock-aligned mesh adaptation, and provide fine-grained control of the turbulent transition location in the simulation.

A. Parallel Efficiency Improvements

The parallel efficiency of running LAURA with multiple MPI ranks per structured grid block has been improved. Figure 1 shows the relative speedup of the overall LAURA simulation for the Exomars vehicle [16] with an increasing number of MPI processes per grid block. In this case, the newly optimized implementation of the parallel communication routines in LAURA version 5.6 result resulted in a 37 percent improvement in parallel efficiency while running with eight processes per block. Additionally, optimizations applied to the LAURA start-up routines are particularly noticeable while running with two processes per block. The previous implementation saw no benefit from the additional parallelism, while the optimized implementation is 1.4 times faster than running with a single process per core.

B. Improved Convergence Monitoring

Improvements have been made to improve monitor the convergence of the point-implicit and line-implicit relaxation procedures in LAURA. While the L2 norm of the residual is typically the best metric of iterative convergence, it is often more practical to monitor the change in solution as a metric of solution convergence. As a proxy for monitoring convergence, the change in integrated convective heating between writes or the LAURA restart file is now printed to the screen output along with aerodynamic quantities in LAURA version 5.6, as shown in Figure 2. This additional screen

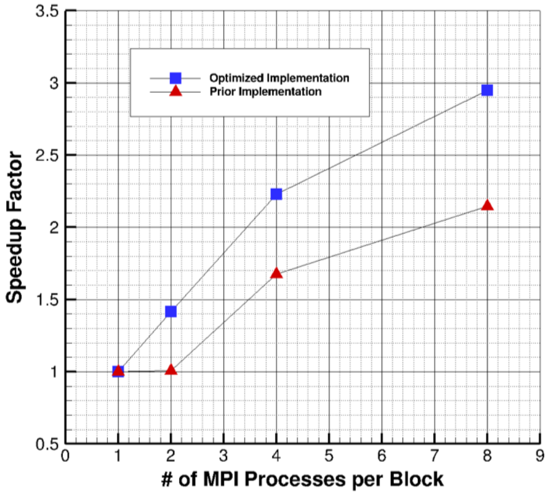


Fig. 1 LAURA parallel efficiency improvements for Exomars vehicle.

output provides users another metric of iterative convergence that can be more useful than monitoring the L2 norm of the residual, particularly in cases where the slope limiter induces a limit cycle. In this case, the L2 norm of the residual fluctuates around a constant value, typically due to cells near a shock wave constantly causing the limiter to engage and disengage, but the solution continues to converge elsewhere. In the case of a limit cycle, change in convective heating can be a much better indication of convergence than the L2 norm of the residual.

```
Saving restart and plot files.
Aerodynamic Coefficients
c_x =   1.537
c_y = -0.7741E-17
c_z = -0.5600
c_l = -0.2209E-17
c_m =  0.5480
c_n =  0.5401E-18
Non-dimensional convective heating: L-1, L-2
  0.786704708073766       3.372131789611955E-002
Change in heating since last write:
  0.196451172896500       4.505566137247407E-003
Change in heating since first write:
  0.589375607160676       1.679656788233272E-002
```

Fig. 2 Example LAURA screen output when writing restart file.

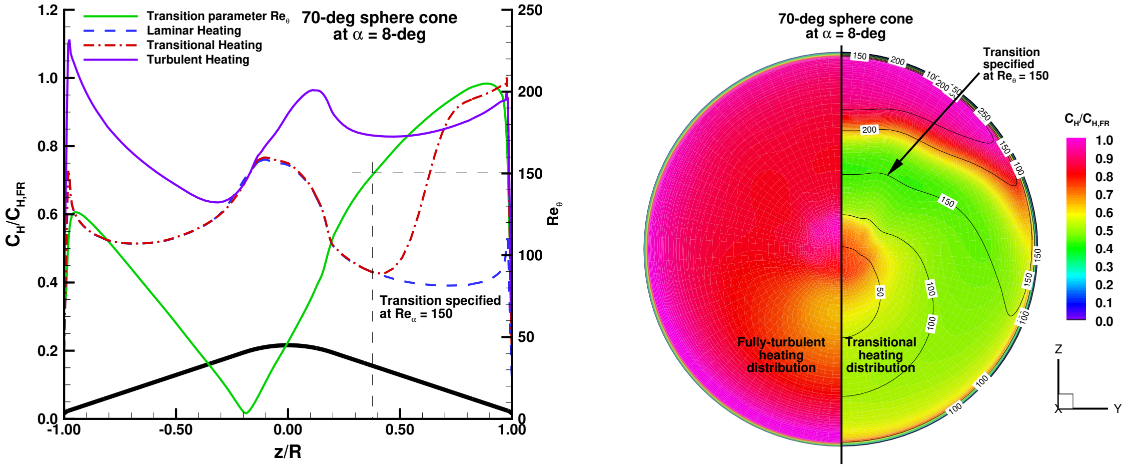
As an additional convergence monitoring criteria, LAURA now saves snapshots of the `laura_blayer` output file that includes relevant boundary layer quantities, such as the integrated momentum thickness. A utility code is included with the version 5.6 release of LAURA that computes statistics of these boundary layer quantities. These snapshot files and the associated utility code may be particularly useful for users conducting time-accurate simulations with LAURA.

C. Improved Spline Interpolation

The grid adaptation capabilities in LAURA have been improved in version 5.6. A cubic spline procedure [17] has been implemented to provide higher-order interpolation for both the solution and mesh vertex locations along k-lines in the structured grid. This spline procedure is used during the LAURA adaptation cycle. First, the mesh outer boundary is aligned with a scalar jump criterion, typically indicative of a bow shock, such that the location of the maximum k-index vertex meets a user criterion. The distribution of points along each k-line is then mapped from a computational space to the true geometric vertex locations by an interpolation procedure. In the previous versions of LAURA, this interpolation procedure was, at best, only able to exactly reproduce a linear geometric entities. This leads to issues with highly curved k-lines, where the adaptation procedure would typically fail by creating k-lines in the structured mesh that intersected, resulting in cells with negative volumes. The newly implemented cubic spline procedure is much better suited to handling curvature in lines, and is able to produce valid structured meshes where the legacy algorithm fails.

D. Boundary-Layer Transition Specification

Various boundary-layer transition specification options have been added to LAURA's Cebeci-Smith and Baldwin-Lomax algebraic turbulence models. These options include specification of the transition location by: a physical distance (radial or lengthwise); a smooth-wall boundary layer parameter (Re_θ or Re_θ/M_e); or by a distributed roughness model (based on ballistic range data, wind tunnel data, or a model that combines both ballistic range and wind tunnel data). Multiple options may be chosen in order to allow for the presence of different phenomena. Additionally, the boundary-layer transition length from laminar to turbulent flow may be controlled by specifying an intermittency factor based on an exponential or hyperbolic function, or by forcing zero-length transition due to a physical trip. An example of boundary-layer transition location based on the Re_θ criteria is shown in Figure 3 for a 70-degree sphere-cone geometry at an angle-of-attack of eight degrees. Figure 3a is a centerline heating distribution comparison between laminar, between transitional and turbulent flow. In both figures, the heat-transfer coefficient has been normalized by the Fay-Riddell stagnation point value.



(a) Centerline Heating Distribution.

(b) Heat Transfer Coefficient.

Fig. 3 Boundary-Layer transition specification for a 70-degree sphere-cone.

IV. HARA Improvements

This section reviews three of the primary algorithmic and code improvements developed to increase HARA's efficiency. These include an advanced ray-tracing capability, an opacity binning approach for non-Boltzmann molecular band systems, and a GPU-enabled implementation in HARA via the OpenACC programming model. All of these significantly decrease the HARA execution time, and are enabling factors in further improvements described in later sections of this paper.

A. Advanced Ray Tracing

To enable routine ray-tracing simulations of the radiative flux, the ray-tracing capability developed by Mazaheri et al. [18] was refined to be both more robust and usable as a subroutine. Details of the robustness improvements are discussed by Johnston and Mazaheri [14], with the primary focus on enabling the ray-tracing computation of the radiative source term. The robustness achieved in this work, along with the efficiency improvements to be discussed, has since made ray-tracing radiative heating simulations to every point on a vehicle surface a routine computation. It was previously unfeasible because of its computational expense. Restructuring the ray-tracing code into a subroutine that can be called within LAURA required a number of minor enhancements. These include the ability to automatically coarsen the surface points at which radiative heating is evaluated and then interpolate this radiative heating back to the full surface grid, and the automatic mirroring of half-body solutions or rotation of axisymmetric solutions, which is required for ray-tracing. These enhancements enabled the application of ray-tracing to the automatic uncertainty analysis discussed in Section V.

B. Opacity Binning Improvements

The line-by-line approach for computing molecular band radiation requires millions of spectral points to accurately model the hundreds of thousands of relatively thin rotational lines. Although the smeared band approach [19] avoids this computational expense by essentially analytically integrating over these lines, it is only appropriate for optically-thin molecular band systems. To develop an approach that is both computationally efficient and accurate for non-optically-thin band systems, the opacity binning concept developed previously for equilibrium radiation [20, 21] was extended to non-Boltzmann molecular band systems by Johnston et al. [15]. In addition to extending this approach to non-Boltzmann molecular band systems, the approach was implemented within HARA so that it could be used simultaneously with line-by-line computations of atomic lines, atomic continuum, or smeared band calculations of other molecular band systems. This allows for the opacity binning approach to be used only for band systems that require it. An approach to reconstruct the line-by-line spectrum from an opacity binning solution was also developed, which unpacks the opacity bins and distributes them across the spectrum. This is useful for analyzing the spectrum and comparing with measurements.

C. GPU Capability

The most computationally expensive tasks in HARA have been reimplemented with OpenACC directives to improve the efficiency of the radiation line-of-sight calculations. The directive-based approach offered by OpenACC was chosen because it was found to be the least intrusive option that allowed access to accelerator hardware without substantial modification of the existing HARA implementation. Table 1 shows a breakdown of the computational costs involved in the line-of-sight calculations for a 10.5 km/s Earth entry simulation. These timings were calculated on a GPU compute node of the NASA Advanced Supercomputer (NAS) Pleiades supercomputer. The non-GPU timings were calculated using all 16 CPU cores of an Intel Xeon E5-2670 dual-socket node, and the GPU times were calculated on a Nvidia Tesla K40m GPU. In this case, using the GPU-enabled code is shown to be almost 10 times faster than the code that does not use the GPU. Utilizing the GPU for radiation calculations is a significant breakthrough, and enables larger scale ray-tracing and tighter coupling between the HARA and LAURA codes.

Table 1 HARA line-of-sight calculation times for 10.5 km/s Earth entry.

Name	Non-GPU Time (s)	GPU Time (s)
Spectral Grid	0.62	0.29
Atomic Lines	1.08	0.22
Atomic Continuum	6.18	0.22
Molecular Bands	7.91	0.22
Other Rad. Mech.	0.22	0.05
Radiation Transport	0.15	0.11
Total	**16.8**	**1.70**

All of these improvements to HARA have enabled the simulation of more challenging reentry simulations, and a new uncertainty quantification technique in the LAURA code.

V. State-of-the-Art Uncertainty Quantification Capability

A state-of-the-art approach to uncertainty quantification of radiative heat transfer has been implemented in LAURA version 5.6. This approach includes parametric uncertainty calculations, as well as a shock tube informed (STI) bias approach [22] for leveraging shock-tube data when calculating uncertainties. This enables the user to obtain flight vehicle radiative heating simulations based on shock-tube measurements with minimal extra effort required. The authors recommend that the application of this new approach be the standard approach for LAURA/HARA radiative heating simulations that support vehicle design projects, because it captures the nominal ray-tracing radiative heating values and the corresponding parametric uncertainty and shock-tube informed bias. These represent the dominant contributions to the radiative heating margin.

A. Novel Shock-tube Informed Bias Approach

The novel STI bias approach developed by Johnston [22] allows shock-tube measurements to inform radiative heating. This approach based on a newly implemented streamline-tracing capability, where the nominal radiative heating is inferred by shock-tube comparisons. Streamline-tracing determines the change in velocity of a streamline processed by a strong shock and the Langrangian time for a fluid parcel crossing a shock to reach the vehicle surface. Both of these parameters from streamline-tracing are used in the mapping of shock-tube agreement factors to the simulation predictions. The process of applying the STI bias approach is largely based on recomputing the radiative heating with the flow field adjusted to match shock-tube measurements. This process is summarized in Algorithm 1, where f_{adjust} is a function of the upper level number density of the transition and is solved at every spectral point, x'.

Algorithm 1: STI bias algorithm

```
Compile relative shock-tube measurements
Compute simultated intensity prediction, $J_{sim}$
$J_{sim,SCF} \leftarrow \int J_{sim}(x')SCF(x'-x)dx'$
$J_{measure}(x) \leftarrow \int J_{sim}(x', f_{adjust}(x'))SCF(x'-x)dx'$
while $J_{measure} - J_{sim,SCF} > tolerance$ do
    $f_{adjust} \leftarrow J_{measure}/J_{sim,SCF}$
end
Map $f_{adjust}$ to flowfield via streamline tracing in LAURA
Perform radiation computation in with $f_{adjust}$ assigned to upper level number density
```

At the end of this process, the radiative heat flux computed by the STI bias approach can be compared with the nominal radiative heat flux to infer the change in radiative heat flux that results from disagreements between the shock-tube measurements.

Figure 4 compares the STI bias radiative heat flux to the nominal radiative heat flux on the surface of a Titan entry vehicle. In this example, the STI bias results suggest that the nominal radiative heat flux should be reduced by more than 16 percent in some surface locations. This approach has also been applied to the proposed Dragonfly mission by Johnston et al. [23] and to support aerothermal analysis of the Schiaparelli Mars entry flight data by by Brandis et al. [24].

B. User Workflow to Perform Uncertainty Quantification

Once the user has set up a LAURA run, no additional input is required other than replacing the traditional LAURA executable with the `hara_uq` executable. The `hara_uq` executable, which is automatically installed with the traditional LAURA executable, will automatically set up the case directory structure and perform separate, parallel runs of the LAURA code. The `hara_uq` automatically maps the MPI processes such that each the `laura` executable is executed with no more than a single MPI process per block. This ensures that the compute resources are allocated in the most efficient way possible when running LAURA. Maximizing the computational efficiency of a `hara_uq` is important because the computational time required by `hara_uq` can be significant.

A single `hara_uq` is roughly equivalent to 20 to 40 executions of the LAURA flow solver run for 5000 flowfield iterations with ray-tracing performed for all user-defined body points. Fortunately, these executions of the LAURA flow solver are embarrassingly parallel. Therefore, the total execution time required by `hara_uq` can be reduced to roughly the same amount of time as a single LAURA flow solver execution, provided there are sufficient computational

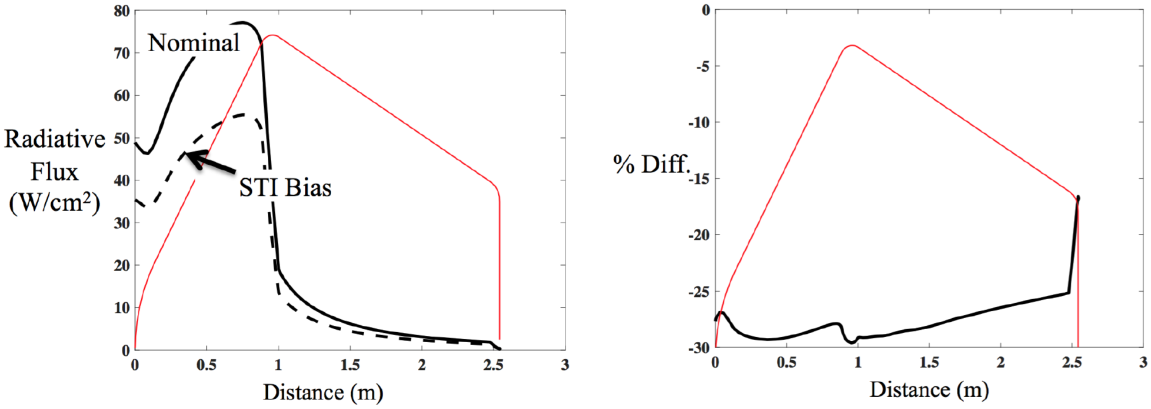


Fig. 4 STI bias values for the ray-tracing radiative flux along the vehicle surface.

resources available. However, a user is not always able to allocate 20 to 40 times the computational effort required by a normal LAURA flow solver execution, so many of the default options for the hara_uq may be altered to reduce the computational load. Changing these options can result in minimal loss of accuracy with some knowledge of the problem of interest. For example, an Earth entry case simulated with a velocity below nine kilometers per second may omit accounting for atomic radiation without significantly impacting the predicted radiative heating. Using this knowledge, the user may opt to run hara_uq without the atomic radiation modeling parameters to reduce computational cost.

VI. Concluding Remarks

The improvements to the HARA code have enabled the application of more accurate physics modeling to more complex geometry, with the addition of GPU acceleration decreasing the most computationally expensive tasks in HARA by an order of magnitude in some cases. The improvements to the LAURA code have enable more functionality in the area of turbulent transition, and better overall convergence and accuracy with the added spline functionality for grid and solution interpolation. The release of LAURA version 5.6 is the first in many years, and marks an improvement to the state of the art for computing the convective and radiative heating for vehicles designed for entry, descent, and landing applications.

VII. Acknowledgements

The authors would like to thank the Entry Systems Modeling Project within the NASA Game Changing Development Program, Space Technology Mission Directorate for their funding and support of this research. The authors would also like to thank Dr. Alireza Mazaheri for his hard work developing and maintaining LAURA since the version 5 release, along with Drs. Bill Wood, Karl Edquist, Tom West, and all those who have contributed bug fixes and improvements to LAURA and HARA over the years.

References

[1] Gnoffo, P. A., “An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows,” NASA TP 2953, 1990.

[2] Johnston, C. O., “Nonequilibrium Shock-Layer Radiative Heating for Earth and Titan Entry,” Ph.D. thesis, Virginia Polytechnic Institute and State University, 2006.

[3] Yee, H. C., “On symmetric and upwind TVD schemes,” NASA TM-1985-86842, 1985.

[4] Braun, R. D., Powell, R. W., Engelund, W. C., Gnoffo, P. A., Weilmuenster, K. J., and Mitcheltree, R. A., “Mars Pathfinder six-degree-of-freedom entry analysis,” *Journal of Spacecraft and Rockets*, Vol. 32, No. 6, 1995, pp. 993–1000.

[5] Pulsonetti, M., and Wood, W., “Computational Aerothermodynamic Assessment of Space Shuttle Orbiter Tile Damage - Open Cavities,” AIAA Paper 2005-4679, 2005. doi:10.2514/6.2005-4679.

[6] Hash, D., Olejniczak, J., Wright, M., Prabhu, D., Pulsonetti, M., Hollis, B., Gnoffo, P., Barnhardt, M., Nompelis, I., and Candler, G., “FIRE II Calculations for Hypersonic Nonequilibrium Aerothermodynamics Code Verification: DPLR, LAURA, and US3D,” AIAA Paper 2007-605, 2007.

[7] Thompson, R., Lessard, V., Jentink, T., and Zoby, V., “Analysis of Compression Pad Cavities for the Orion Heatshield,” AIAA Paper 2009-1575, 2009.

[8] Johnston, C. O., and Brandis, A. M., “Features of Afterbody Radiative Heating for Earth Entry,” *Journal of Spacecraft and Rockets*, Vol. 52, No. 1, 2015, pp. 105–119.

[9] Wright, M., “A Family of Data-Parallel Relaxation Methods for the Navier-Stokes Equations,” Ph.D. thesis, University of Minnesota, 1997.

[10] Beck, R. A. S., Songer, J. T., Szalai, C. E., Saunders, D. A., Karlgaard, C. D., and Johnson, M. A., “InSight Aerothermal Environment Assessment,” AIAA Paper 2020-1273, 2020. doi:10.2514/6.2020-1273.

[11] Erb, A. J., West, T. K., and Johnston, C. O., “Investigation of Galileo Probe Entry Heating with Coupled Radiation and Ablation,” AIAA Paper 2019-3360, 2019. doi:10.2514/6.2019-3360.

[12] Brune, A. J., West, T. K., and White, L. M., “Calibration Probe Uncertainty and Validation for the Hypersonic Material Environmental Test System,” *Journal of Thermophysics and Heat Transfer*, Vol. 34, No. 2, 2020.

[13] Chandrasekaran, S., and Juckeland, G., *OpenACC for Programmers: Concepts and Strategies*, 1st ed., Addison-Wesley Professional, 2017.

[14] Johnston, C. O., and Mazaheri, A., “Impact of Non-Tangent-Slab Radiative Transport on Flowfield-Radiation Coupling,” *Journal of Spacecraft and Rockets*, Vol. 55, 2018, pp. 899–913.

[15] Johnston, C. O., Sahai, A., and Panesi, M., “Extension of Multiband Opacity-Binning to Molecular, Non-Boltzmann Shock Layer Radiation,” *Journal of Thermophysics and Heat Transfer*, Vol. 32, No. 3, 2018, pp. 816–821.

[16] Pinaud, G., Bertrand, J., Soler, J., Tran, P., and Ritter, H., “Exomars mission 2016: A preliminary post-flight performance analysis of the heat shield during entry on Mars atmopshere,” AIAA Paper 2019-0244, 2019.

[17] Steffen, M., “A simple method for monotonic interpolation in one dimension.” *Astronomy and Astrophysics*, Vol. 239, 1990, pp. 443–450.

[18] Mazaheri, A., Johnston, C., and Sefidbakht, S., “Three-Dimensional Radiation Ray-Tracing for Shock Layer Radiative Heating Simulations,” Vol. 50, No. 3, 2013, pp. 485–493.

[19] Johnston, C. O., Hollis, B. R., and Sutton, K., “Spectrum Modeling for Air Shock-Layer Radiation at Lunar-Return Conditions,” Vol. 45, 2008, pp. 865–878.

[20] Wray, A. A., Prabhu, D. K., and Ripoll, J.-F., “Opacity Distribution Functions Applied to the CEV Reentry,” AIAA Paper 2007–4542, 2007.

[21] Scoggins, J. B., Magin, T. E., Wray, A., and Mansour, N. N., “Multi-Group Reductions of LTE Air Plasma Radiative Transfer in Cylindrical Geometries,” AIAA Paper 2013–3142, 2013.

[22] Johnston, C. O., “Evaluating Shock-Tube Informed Biases for Shock-Layer Radiative Heating Simulations,” AIAA Paper 2019-3259, 2019.

[23] Johnston, C. O., West, T. K., and Brandis, A. M., “Features of Afterbody Radiative Heating for Titan Entry,” AIAA Paper 2019-3010, 2019.

[24] Brandis, A. M., White, T. R., Saunders, D., Hill, J., and Johnston, C. O., “Simulation of the Schiaparelli Entry and Comparison to Aerothermal Flight Data,” AIAA Paper 2019-3260, 2019. doi:10.2514/6.2019-3260.