

Research Article

High-Fidelity Aerothermal Engineering Analysis for Planetary Probes Using DOTNET Framework and OLAP Cubes Database

Prabhakar Subrahmanyam

Department of Mechanical and Aerospace Engineering, Center of Excellence for Space Transportation & Exploration, San Jose State University, One Washington Square, San Jose, CA 95192, USA

Correspondence should be addressed to Prabhakar Subrahmanyam, prasub@gmail.com

Received 15 January 2009; Accepted 15 April 2009

Recommended by Tom Shih

This publication presents the architecture integration and implementation of various modules in *Sparta* framework. *Sparta* is a trajectory engine that is hooked to an Online Analytical Processing (OLAP) database for Multi-dimensional analysis capability. OLAP is an Online Analytical Processing database that has a comprehensive list of atmospheric entry probes and their vehicle dimensions, trajectory data, aero-thermal data and material properties like Carbon, Silicon and Carbon-Phenolic based Ablators. An approach is presented for dynamic TPS design. OLAP has the capability to run in one simulation several different trajectory conditions and the output is stored back into the database and can be queried for appropriate trajectory type. An OLAP simulation can be setup by spawning individual threads to run for three types of trajectory: *Nominal, Undershoot and Overshoot trajectory*. *Sparta* graphical user interface provides capabilities to choose from a list of flight vehicles or enter trajectory and geometry information of a vehicle in design. DOTNET framework acts as a middleware layer between the trajectory engine and the user interface and also between the web user interface and the OLAP database. Trajectory output can be obtained in TecPlot format, Excel output or in a KML (Keyhole Markup Language) format. Framework employs an API (application programming interface) to convert trajectory data into a formatted KML file that is used by Google Earth for simulating Earth-entry fly-by visualizations.

Copyright © 2009 Prabhakar Subrahmanyam. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Space exploration directorate's national vision calls for "*The human and robotic exploration of the Solar System and beyond*." Human and Robotic exploration of the solar system is to search for evidence of life, to understand the history of the solar system, and to support human exploration. But before they arrive, *planetary probes* will likely have preceded them to provide the understanding required to make further exploration possible. Advances in technologies ranging from new instrumentation, sophisticated heatshield materials, and improved nanotechnology make planetary probes a vital tool in pursuit of scientific truth and the origins.

In support of this mission, a complete computational design framework is necessary that automates the calculations for stagnation point heating, TPS sizing so that better probes can be built with this analysis. An integrated planetary probe design framework that automatically generates grids

for planetary probes given vehicle geometry and entry trajectory flight conditions is presented in this publication. The probe design framework includes access to existing probe designs and provides a mini-CAD-like design environment for the construction of new configurations based on classes consistent with existing designs. A platform-independent graphical user interface- (GUI-) based, relational database-driven tool called *Sparta* has been developed to generate customized grids (*Sparta*-Mesh) required to accurately predict aerodynamic and heating entry environments. The aeroheating environment depends on the trajectory flown, size, and shape of the vehicle. Figure 1 shows the software architecture of *Sparta*. Once the grids are generated, they can be automatically linked to CFD tools for analysis.

A comprehensive database of different ballistic re-entry vehicles has been developed using the Planetary Mission Entry Vehicles manual [1]. This includes vehicle dimensions and trajectory data for all the capsules that has been flown

in the past. The *Sparta* GUI provides the capability to choose from a list of flight vehicle geometric information and entry trajectories and generates 3D volume grids. The work presented demonstrates a geometry centered automatic grid generation capability for planetary probes. The application also analyzes the effect of the planetary probe flight trajectories on the volume grid requirements as a function of flight velocity and altitude.

An approach is presented for dynamic TPS sizing. Materials properties like Carbon-, Silicon-, and Carbon-Phenolic-based ablators have been obtained from the NASA Ames Thermal Protection Materials and System Branch TPSX Internet Database and modeled into the local OLAP database and integrated with the user interface. Any previously flown planetary probe can be chosen from the user interface and tied to an ablator from the list in the graphical interface and a trajectory case can be run for a planet, that is, available in the database. This populates the trajectory data back into the database. Two different ablators can be chosen for the same probe and the tool can run two different trajectories and the aerodynamic heating is computed for comparative data analysis. Also, two different ablator types from two different categories like Carbon ablator and Silicon-based ablators can also be chosen for TPS sizing analysis. The purpose of this research is to investigate the aerothermodynamics environment of atmospheric entry probes and to generate an automatic computational trajectory database populated by OLAP cubes and to provide a comprehensive flow field analysis for such probes.

2. Software Architecture

The overall software architecture is shown in Figure 1. The front graphical user interface allows the user to provide inputs for trajectory, geometry, and grid generation computations. It is connected to an extensive planetary probe database and a matlab/Java computational engine, which is used to generate flight trajectory data. Based on the flight conditions and probe specifications, the geometry engine constructs the vehicle configuration. It also generates the inflow outer boundary and outflow surfaces that define the computational volume for grid generation. The geometry constructed feeds directly to *GridPro* and is used to automatically set up the mesh topology and surface assignments for grid generation. The *GridPro Ggrid* solver [2] is then run in the background to generate the volume grid. The entire process results in high-quality volume grids and is transparent to the user. With the emerging Information Technology growth, heterogeneous systems are rapidly developing and this presents a need for distributed analysis system. So, with this in mind, a platform-independent tool has been developed. This architecture neutral code has the capability to execute in any platform for analysis without having to recompile.

2.1. Graphical User Interface. The GUI-based analysis tool was developed in the Java platform-independent environment. Figure 2 shows the probe design options provided in

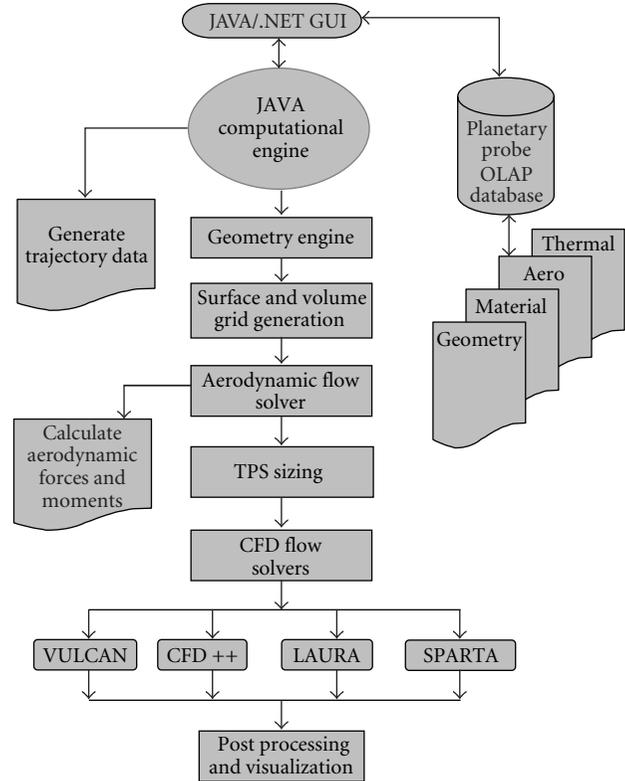


FIGURE 1: *Sparta* framework software architecture.

the *Sparta* interface. A planetary probe can be chosen from the user interface flight vehicle drop down menu. When the flight vehicle is chosen, appropriate initial trajectory and vehicle dimensions data are populated in the input boxes from the relational database. Values in the input boxes can be changed by the user or the populated values can be used to run the trajectory simulation. The user is also able to choose a stagnation point correlation, either *Fay Riddell* or *Sutton Grave* correlations from the interface.

The user can either choose a flight vehicle for trajectory analysis from the list of available probe designs in the planetary probe database as explained above or construct a new configuration. This is achieved by calling the geometry engine from within the GUI. After the vehicle is constructed, the trajectory is computed by calling the computational engine from within the GUI. The final step is to automatically generate surface and volume grids [3]. The *TopoGen* utility was developed to construct the topology, internal and external surfaces for preprocessing, and setup of the *GridPro* grid generation program.

2.2. Atmospheric Profile. *Sparta* design environment also links the trajectory code to appropriate planetary empirical atmospheric models depending on the planetary probe that is chosen. The Earth atmosphere model utilized by *Sparta* for the entry trajectory design and analysis is the 1976 US Standard Atmosphere for Earth called GAME—General Atmospheric Model for Earth that is modeled as a subroutine

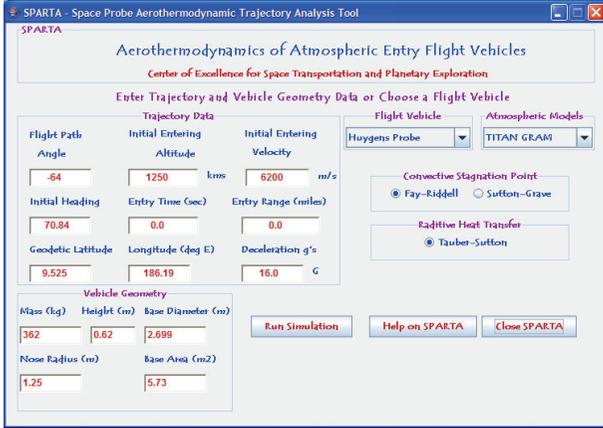


FIGURE 2: Sparta JAVA graphical user interface.

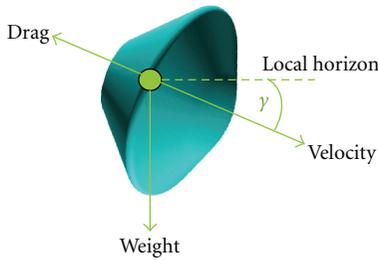


FIGURE 3: Aerodynamic forces acting on the atmospheric vehicle.

to calculate pressure, density, temperature, Reynolds number, and speed of sound as a function of altitude. GRAM Models [4] of MARS, Venus, Titan, and Neptune have been used to compute the atmospheric pressure, density, and temperature profiles. The advantage of the GRAM models is its ability to include seasonal dependence, geographical dependence also the ability to model uncertainties, and so forth.

2.3. Atmospheric Probe Model. The probe model developed for this trajectory is a point-mass model with two translations and one rotation (3-DOF) around a spherical planet. It integrates the equations of motions of a vehicle on a ballistic entry trajectory so that no lift is generated and the body acts only on gravity. Figure 3 shows the various aerodynamic forces acting on the body. The vehicle model is built from a number of parameters defining the geometry of the probe including body diameter, cone half-angle, as well as nose and shoulder radius. The aerodynamic properties of the probe are subsequently derived from the geometry of the vehicle.

Modified Newtonian flow theory (NFT) is used to evaluate the pressure coefficient C_p around the body and derive the drag coefficient C_d of the configuration. NFT follows the standard Newtonian sine-squared law, with an adjustment to give correct pressure coefficient at the stagnation point. The modification to the Newtonian flow is represented here (as it is in (1)) in terms of the pressure coefficient,

$$C_p = C_{p_{\max}} \sin^2 \theta, \quad (1)$$

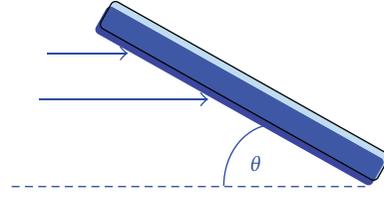


FIGURE 4: Modified NFT.

where C_p is local surface pressure coefficient,

$$C_p = \frac{p - p_{\infty}}{(1/2)\rho_{\infty} V_{\infty}^2}. \quad (2)$$

In (2), p is the local surface pressure, p_{∞} is the freestream static pressure, and $q_{\infty} = (1/2)\rho_{\infty} V_{\infty}^2$ is the freestream dynamic pressure. The *deflection angle* θ is the angle between the flow and the body surface as shown in Figure 4 and $C_{p_{\max}}$ is evaluated as the maximum pressure coefficient found behind a normal shockwave at the stagnation point. The Ballistic Coefficient of the probe is derived from the aerodynamic model as

$$\beta = \frac{m}{C_d}(S), \quad (3)$$

where m is the mass and S is the section area of the body. The higher the ballistic coefficient, the higher the heat and deceleration loads. Once the ballistic coefficient is determined, *Sparta* provides a number of entry profiles for various velocities (V_e) and entry angles (γ_e).

2.4. Java Framework and Parallel Sparta. Parallel *Sparta* is a set of pure Java programs based on the Java thread concept as well as the Java remote method invocation (RMI) to serve as a test suite to measure the suitability of the Java Object-Oriented-Programming (OOP) methodology to be used for large-scale applications in Aerospace engineering. It was decided to code in Java because it is easily portable and platform-independent. The Java OOP approach provides profound improved software productivity which is the main benefit of using Java and makes it phenomenally easier for a programmer to use *threads*, which is the primary route to parallelism and therefore speed on shared-memory machine architecture of many modern machines.

Multithreading mechanism provides improved performance from a code, because many machines have extra processors that can run the extra threads. In the "C" language, it is not only difficult to create threads but it is also difficult to manage threads, therefore programmers seldom use threads in their programs. However, in Java it is relatively easier to spawn a new thread and threads management is easier as well. Thus there is a natural tendency to use threads in programs and threaded applications are becoming much more natural and widespread. Albeit the main advantage of Java compilation is that the programs are statically compiled, it is still imperative to do some runtime checking because null references checking, array bounds checking and runtime type checking cannot be done during compile time. This

turn makes Java programs more robust, but it also makes the generated code a little slower than the equivalent “C” program. Nevertheless, many of these checks can be reduced or eliminated during runtime by the native code generator.

In *Sparta*, concurrency is achieved via the thread concept. Threads are run concurrently, the mapping of threads to processors as well as the scheduling being done by Java and the OS. Thus we have a way to get dynamic load-balancing of a parallel application without explicitly assigning tasks to processors: a threaded application is said to be *self-scheduling*. Java also provides a mechanism for synchronizing threads and for sending messages between threads. Furthermore we no longer need message-passing libraries such as MPI and PVM to communicate between threads, but we can use shared memory or remote method invocation (RMI) instead.

The Java programming language comes with a built-in suite features that make it extremely attractive for writing high-quality, portable parallel programs. Java’s native exception model, pure object formulation, and strong *typing* make programs easier to construct, debug, and maintain. Threads provide an elegant route to parallelism on shared-memory architectures. Anticipating great improvements in numerical performance, and for faster numerical convergence, it was decided to parallelize *Sparta* and put to test how a pure Java *Navier-Stokes* solver might perform. The suite includes a parallel Navier-Stokes solver. *Sparta* was tested on a 32-processor Intel machine and a 4-processor Sun server. While the indication on speedup is excellent on both machines, promising a high-quality thread scheduler, the single-processor performance obviously needs a lot of improvements.

3. OLAP Database Cubes and Multidimensional Analysis

3.1. OLAP Cubes. An OLAP cube has been constructed for multidimensional trajectory analysis [5]. A cube is a store of multidimensional data; it is defined by dimensions and measures. Different dimensions are populated in the database. Several measures are populated in the database for each of these dimensions. These variables that are characterized as measures: Effective nose radius (R_{eff}), Nose radius (R_n), Shoulder radius (R_s), Corner radius (R_c), vehicle’s mass (m), initial flight path angle (γ_e), and initial entry velocity (V_e). Once these dimensions are specified as different values in the cube and the 3-DOF trajectory engine is run, it populates the database with all these different permutations and combinations of the design parameters and the data can be fetched from the database either by specifying a query or by choosing the appropriate design parameter for a set of flight conditions.

OLAP cubes are constructed as dimensions and measures in the cubes editor as shown in Figure 5. Measures are the design parameters that are modeled in the OLAP database. These design parameters can be changed and forms the basis for multidimensional data analysis capability. Figure 6 shows 23 different probes that are modeled as the dimension

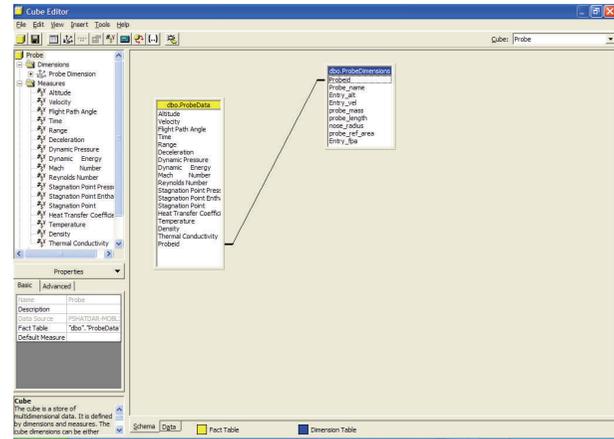


FIGURE 5: Dimensions modeled in OLAP cube for a planetary probe.

members in the cubes. Once the design parameters for these probes are modeled in the database as measures, from the trajectory graphical user interface, the flight vehicle is chosen, the appropriate dimensions and the measures from the OLAP cubes are loaded as inputs for the trajectory. Once the trajectory is run, data is generated from the 3-DOF trajectory engine and populated back into the database. From this point, it is data mining to retrieve the data for the specified set of flight conditions, thus for a change in the initial entry velocity, for instance, a new run is not necessary, it is just a fetch from the OLAP cubes by browsing the dimensions data.

3.2. The Mathematics of Trajectory Data Storage and Data Mining. Cubes are the logical storage medium for an OLAP database. The cube to an OLAP database is what the table is to a Relational Database Management Systems (RDBMS). A cube presents to the outside world, users, and front-end applications, a potential intersection point, that is, a cell for every member from every dimension with every member of every other dimension. Cubes give the ability to model multiple loops in the trajectory code without having to write loops. Several values for three dimensions are changed in *Sparta* cubes which are linked to the trajectory code. They also provide a capability to change up to 128 design parameters. The trajectory code is run once and the data-warehouse is populated by varying these design parameters. Data mining techniques are employed to drill down the data and queried for specific flight conditions from the database.

Sparta Cubes have multiple values for these three dimensions as shown in Figure 7: (a) initial flight path angles, (b) Initial entry velocity, and (c) Nose radius. The term cube implies *three dimensions*. In fact OLAP server cubes can contain up to 128 dimensions. For every point in the altitude integration, changing all the design parameters for a simulation run having five different values for each design parameter comprises 128 loops with five different values each and populates huge datasets in the database. This is just for one point in the altitude starting at an atmospheric

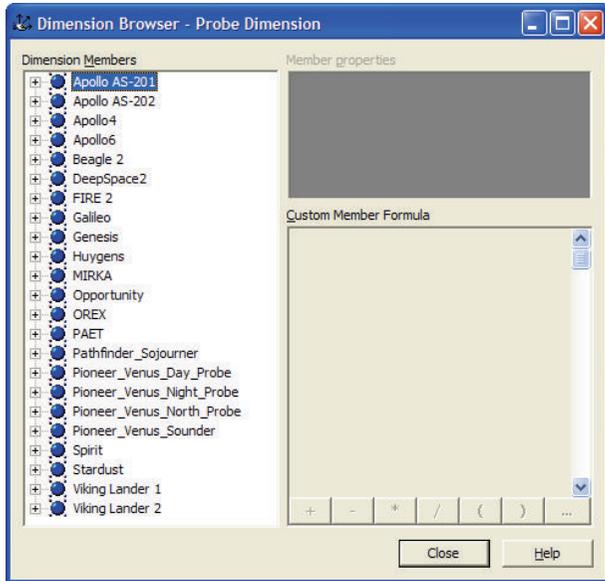
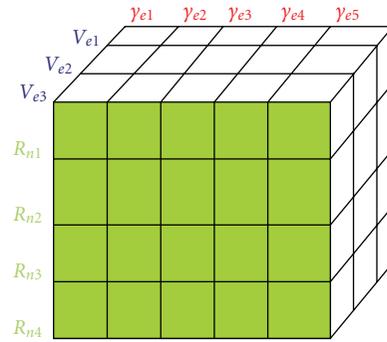


FIGURE 6: Planetary probes modeled in OLAP DB.

interface of 400 000 feet for *Apollo* type vehicles. A user-specified altitude increment of 1000 feet up to a *zero feet* at splashdown involves 400 simulations as the vehicle traverses through the trajectory simulation generating huge datasets. Choosing few design parameters generates little data and occupies little space. A judicious and limited set of design parameters for the probe becomes mandatory although in practice there are not 128 different variables to play with while generating trajectory data. Also, for such high volumes of data, efficient data retrieval and data mining techniques become imperative.

The trajectory engine takes a little longer if more dimensions are included in the simulation. Each variable can have a maximum of up to five values. If each variable has five different values, then in this case each variable forms a separate loop and the trajectory engine is run within these three loops populating the data for 125 permutations and combinations of these design variables for each single point in the altitude. This whole setup is populated for the entire altitude integration in the database. Hence a drill-down approach to fetch the data from the OLAP database becomes necessary if several dimensions are specified for the trajectory simulation run. Tables 1 and 2 show the data queried (drill-down approach) from the OLAP cubes database that has been populated by varying five different values for the entry angles, entry velocities, and nose radii (three of the 128 dimensions that can be modeled). Also, *Sparta* framework can fetch from the database the best trajectory based on either the minimum heating loads or the minimum *g-forces* that is chosen in the optimization section from the graphical user interface.

Data in Tables 1 and 2 has been fetched (or *queried*) from a user-specified query in the OLAP analysis manager and can be specifically tailored to suit one of the entry conditions that has been used in the OLAP cubes to populate the database.

FIGURE 7: *Sparta* OLAP cube as modeled in the database.

So, the idea is to run the trajectory engine only once for several different design conditions and populate the database with huge datasets that has all the different permutations and combinations of trajectory data pertinent to a specific flight vehicle and later retrieve the data for specified flight and design conditions under investigation. The initial entry flight conditions in the OLAP cubes for the *Apollo* vehicles were taken from *Apollo* experience report [6].

3.3. DOTNET-Driven Aerothermal Analysis. A relational database management system *Sparta* DB has been developed and integrated with DOTNET framework for web accessibility as shown in Figure 8. RDBMS allows data of existing planetary design to be spread out in different tables like aerothermal, geometry, materials, and so forth to be linked via a column, for example, flight vehicle [7].

The database manager allows selective data retrieval through .NET web application user interface as shown in Figure 9. A planetary probes database is developed in Microsoft's SQL Server and is populated via the DOTNET web application which is organized according to the planetary body. Choosing a planetary body from the populated drop-down list of the graphical user interface gives all the probes flown to the specific planet as illustrated in Figure 10.

The DOTNET Web application retrieves the data from the flight vehicle database and populates the browser with all the flight vehicles atmospheric entry specific to the planet. Each flight vehicle contains aerothermal, geometry, and TPS information from the database and is displayed in the web browser [7]. Clicking on a specific flight vehicle launches the GUI with all the aerothermal data retrieved from the database and populated in the GUI as shown in Figure 12. The 3-DOF point mass trajectory code can be executed from here.

Thus the trajectory code can be run as an independent stand-alone mode and also from the web. Trajectory code that is run from the web interfaces with the web server as the computations are done on the server and the trajectory data is shown on the browser as a formatted HTML output that can be saved for plotting. Stand-alone trajectory code automatically generates the trajectory data in ASCII files and also invokes the visualization utility TecPlot [8] for plotting.

TABLE 1: OLAP driven trajectory analysis for *Apollo 4* at an entry velocity of 22500 ft/s and $R_n = 4.69$ m.

<i>Apollo 4</i> , entry angle: -12.0° , entry velocity: 22500 ft/s, nose radius: 4.69 m						
Altitude (ft)	Velocity (ft/sec)	Flight path angle γ (deg)	Mach number	Reynolds number	Stagnation point pressure (lbf/ft ²)	Stagnation point heat transfer rate \dot{q}_{conv} (W/cm ²)
250000.00	22513.69	12.00	23.83	65486.33	30.70	391.30
249000.00	22512.32	12.00	23.80	68432.20	32.16	395.82
248000.00	22510.96	12.01	23.77	71501.97	33.69	400.44

TABLE 2: OLAP driven trajectory analysis for *Apollo 4* at an entry velocity of 36545 ft/s and $R_n = 5.69$ m.

<i>Apollo 4</i> , entry angle: -6.93° , entry velocity: 36545 ft/s, nose radius: 5.69 m						
Altitude (ft)	Velocity (ft/sec)	Flight path angle γ (deg)	Mach number	Reynolds number	Stagnation point pressure (lbf/ft ²)	Stagnation point heat transfer rate \dot{q}_{conv} (W/cm ²)
250000.00	36545.00	6.93	33.75	106364.35	81.02	482.53
249000.00	36544.30	6.92	33.75	111144.56	84.87	482.90
248000.00	36544.00	6.91	33.70	116125.59	88.89	483.01

The trajectory code *Sparta* can be invoked from the DOTNET user interface and can be executed from the web as shown in the architecture in Figure 8. *Sparta* is hooked to an OLAP database for multidimensional analysis capability. OLAP has the capability to run in one simulation several different trajectory conditions and the output is stored back into the database and the data can be queried for appropriate trajectory type. For instance, given any one planetary probe dimension from the database, an OLAP simulation can be set up to run for three types of trajectory: *Nominal*, *Undershoot*, and *Overshoot trajectory*. This aids as a very good preliminary design tool for a trajectory based TPS sizing and design.

The DOTNET interface provides a lot of options to choose from to set up the trajectory simulation. From the DOTNET interface, user can choose a flight vehicle and the appropriate trajectory data, flight vehicle's geometry and dimensions data are fetched from the database and populated into the GUI. At this point, the user has the flexibility to change the values or use the populated values. The user can also choose *Fay-Riddell* or *Sutton-Graves* correlation for convective stagnation point analysis and *Tauber-Sutton* for radiative stagnation point analysis. Based on the flight vehicle that is chosen, the heatshield material is automatically selected from the database. OLAP db has different classes of ablators. Carbon- or silicon-based ablators can be set up for any vehicle to do the preliminary analysis. Similarly, any atmospheric model can be invoked for the vehicle that is chosen. Figure 11 shows the DOTNET framework UI and when a flight vehicle like Mars *Pathfinder* is chosen from the UI, the appropriate trajectory, geometry, heatshield and material data, and atmosphere to be flown are all automatically populated into the UI from the database as shown in Figure 12. The atmospheric model MarsGRAM and the heatshield material *SLA_561V* that was used for the *Pathfinder* probe are invoked.

Simulation parameters are given for the user to choose the simulation termination conditions for the trajectory run. The options available are *Surface Impact*, *Apoapsis*, *Mach number*, and *Altitude*. Choosing *Surface Impact* causes the

trajectory to run the simulation and generate data until the vehicle impacts the surface at the target body. *Apoapsis* is to aerocapture the vehicle at a prescribed Apoapsis altitude and terminate the trajectory run at that altitude. *Mach number* is to run the trajectory until the Mach number fall below the specified Mach number value. Altitude option is to terminate the trajectory run until a user-specified altitude is reached in the simulation.

3.4. Driving Google Re-entry Visualization from DOTNET Framework. The commercial launch of Google Earth (GE) in 2005 revolutionized Earth entry fly-by simulations. Although its data analysis capabilities are underpowered compared to more traditional geographical information system (GIS) packages, it is significantly simpler to use and more richly interactive than previous geobrowsers. For a simple geospatial exploration of data, it is unparalleled in ease of use. The GE commercial off the shelf (COTS) technology package [9] is free of cost, bringing access to high-quality geographical Earth-entry imagery within the reach of any aerospace research scientist equipped with a computer that is connected to the web.

3.4.1. Keyhole Markup Language. Keyhole Markup Language (KML), is an XML-based language schema for expressing geographic annotation and visualization on existing or future web-based, two-dimensional maps and three-dimensional Earth browsers. As such, it has two main strands. The first part of KML is the semantic layer which consists of a set of tags for marking up geospatial primitives like two- and three-dimensional specifications for points, lines, and polygons that represents the geospatial data, while the second part of it is concerned with how that data should be displayed within Google Earth. It contains, therefore, a mixture of semantic and presentation layers. In order to make GE accessible as a visualization platform, data has to be produced in KML format so that re-entry trajectory can

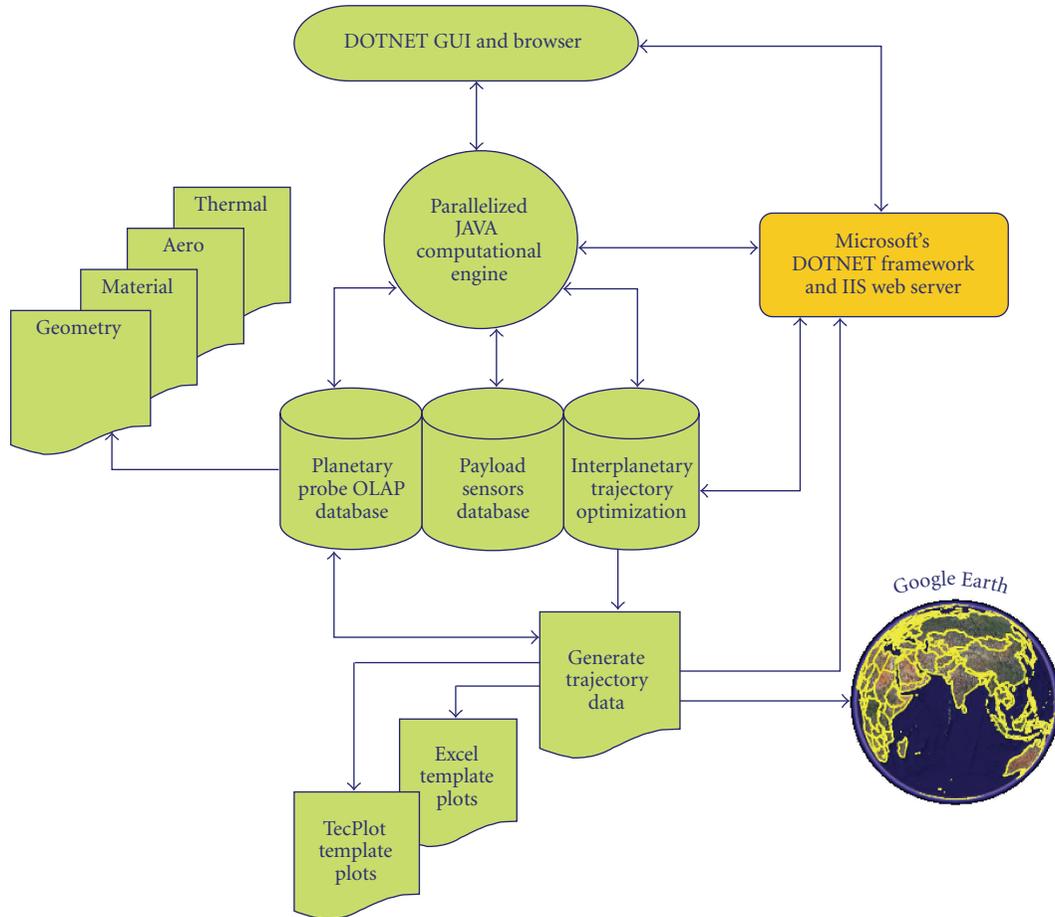


FIGURE 8: DOTNET web application architecture.

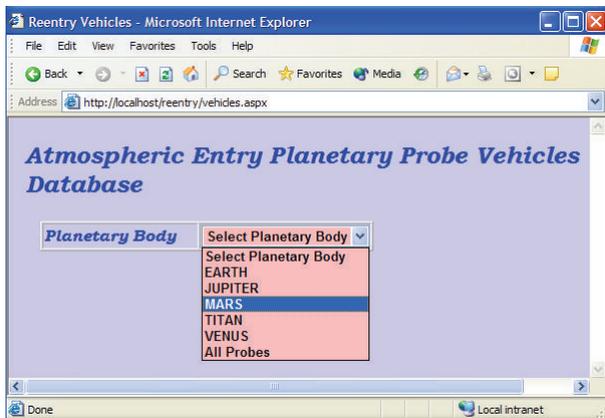


FIGURE 9: Atmospheric entry DOTNET web user interface.

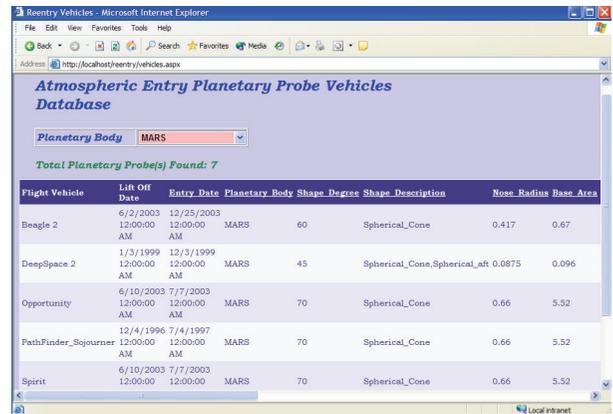


FIGURE 10: Atmospheric entry planetary vehicle web application.

be simulated. This task, although not taxing for the well-educated programmer familiar with XML, is not entirely trivial. KML is another data format whose intricacies one must learn.

The framework employs a pure Java library which generates KML output to display the most commonly interesting forms of geospatial data. *Sparta* framework gives the option

to output the trajectory data into KML, the native format for Google Earth so that re-entry visualizations can be simulated in the Google Earth commercial package as shown in Figure 13. Using this API requires no prior knowledge of KML for the end-user. The API expects data in the form of arrays, in which it is likely to already exist, and will create graphical elements of the sort which users are most likely

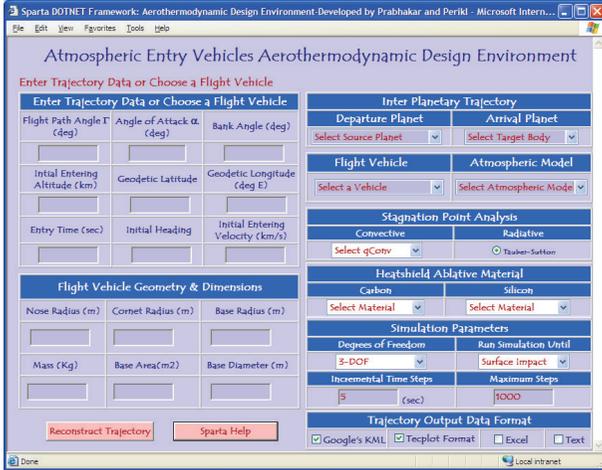


FIGURE 11: Sparta DOTNET framework user interface.

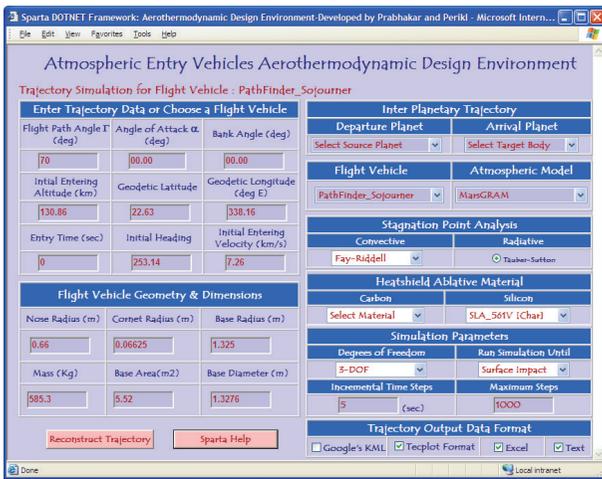


FIGURE 12: Sparta framework flight vehicle user interface.

to want. Google Earth interprets its native KML format and produces fly-by visualization along the entry trajectory.

4. Trajectory Analysis

4.1. Apollo Geometry. The forebody geometry for *Apollo 4* is geometrically similar to that used for the other *Apollo* missions. It is a 30-degree cone with a nose radius of 4.69 m, vehicle height of 3.91 m, base area of 12.02 m², and vehicle mass of 5424.9 kg. The ballistic coefficient β has been calculated to be 395.8 kg/m².

The *Apollo* capsules were chosen for demonstration of the trajectory analysis capability. A sample trajectory data is shown in Table 3 and in Figures 15(a)–15(f). In the example presented the GUI is first populated with *Apollo* specific mission and design information retrieved from the planetary probes database. The atmospheric entry conditions and vehicle dimensions are used in the *Sparta* environment to perform the trajectory calculations. A fourth-order *Runge-Kutta* integration is employed for trajectory calculations to

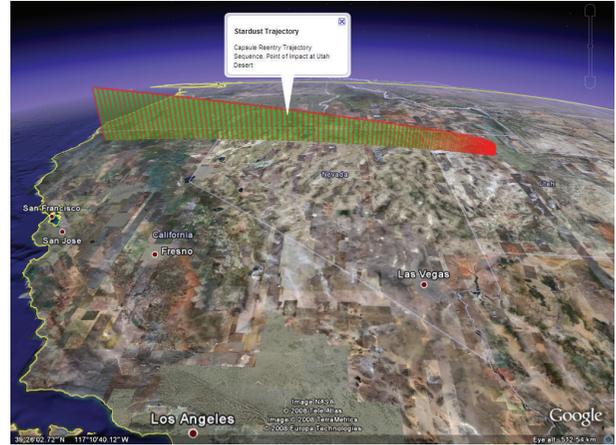


FIGURE 13: Google Earth: Stardust entry trajectory visualization.

advance the solutions of differential equations to solve the following set:

$$\begin{aligned} \frac{dV}{dh} &= \frac{g[(Q/\beta) - \sin(\gamma)]}{V \sin(\gamma)}, \\ \frac{d\gamma}{dh} &= \frac{\cos(\gamma)[-g + (V^2/(R_e + h))]}{V^2 \sin(\gamma)}, \\ \frac{dt}{dh} &= \frac{-1}{V \sin(\gamma)}, \\ \frac{dr}{dh} &= R_e \frac{d\theta/dt}{dh/dt} = \frac{-R_e \cos(\gamma)}{(R_e + h) \sin(\gamma)}. \end{aligned} \tag{4}$$

4.2. Undershoot and Overshoot Trajectory. For planetary entry vehicle design, the atmospheric entry envelope is usually bounded by *undershoot* and *overshoot* trajectory. Generally, a *nominal* trajectory for a given entry angle is defined, along with *overshoot* and *undershoot* trajectories. The *overshoot* trajectory is characterized by the shallowest entry angle given the uncertainty in targeting, and for TPS design purposes is associated with lower heat flux than the *nominal* but with a higher heat load due to the longer heat pulse. This trajectory is generally used for TPS sizing since it leads to the greatest TPS thickness.

Stagnation point heating analysis is performed using the *Fay-Riddell* [10] correlation. Transport properties such as the coefficient of viscosity and thermal conductivity are modeled using *Sutherland's law* for each data point along the flight trajectory. User inputs include initial latitude, longitude, inertial velocity, re-entering altitude, and initial flight path angle and these values can be changed in the GUI if desired.

Trajectory data generated by *Sparta* for MARS pathfinder probe is shown in Table 4 that used MARSGRAM atmospheric model. The trajectory code generates the vehicle's freestream velocity, flight path angle, deceleration, stagnation point heating, Reynolds number, and Mach number along the trajectory as a function of altitude. Re-entry time and range are calculated as well. The trajectories computed are shown in Figures 15(a)–15(f) for the selected

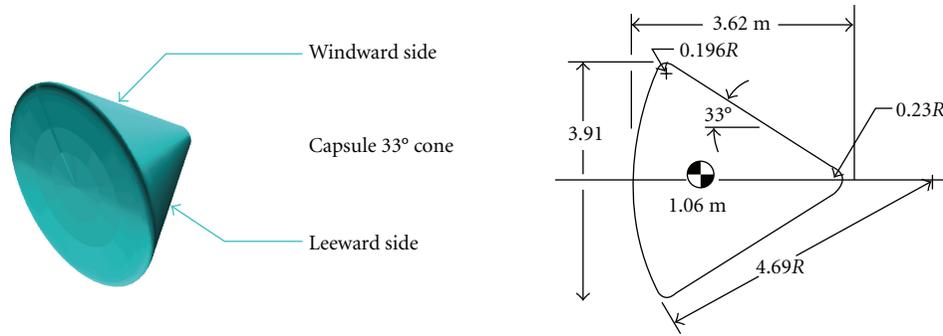


FIGURE 14: Geometry of the *Apollo 4* aeroshell capsule.

TABLE 3: *Apollo 4* Trajectory data calculated by *Sparta* trajectory engine.

Trajectory analysis for flight vehicle: <i>Apollo 4</i>						
Altitude (ft)	Velocity (ft/sec)	Flight path angle γ (deg)	Mach number	Reynolds number	Stag point pressure (lbf/ft ²)	Stag point heat transfer rate \dot{q}_{conv} (W/cm ²)
250000.00	22513.69	12.00	23.83	25524.59	30.70	191.30
249000.00	22512.32	12.00	23.80	26672.87	32.16	195.82
248000.00	22510.96	12.01	23.77	27869.44	33.69	200.44
247000.00	22509.59	12.01	23.73	29116.20	35.28	205.17
246000.00	22508.23	12.02	23.70	30415.12	36.95	209.99
245000.00	22506.86	12.02	23.67	31768.23	38.69	214.90

atmospheric entry conditions and the figures show the variation of flight velocity, Reynolds number, stagnation point pressure, and deceleration against atmospheric altitude during hypersonic ballistic re-entry. Three different flight vehicle trajectories have been analyzed for different values of flight path angle (γ) and Ballistic Coefficient (β).

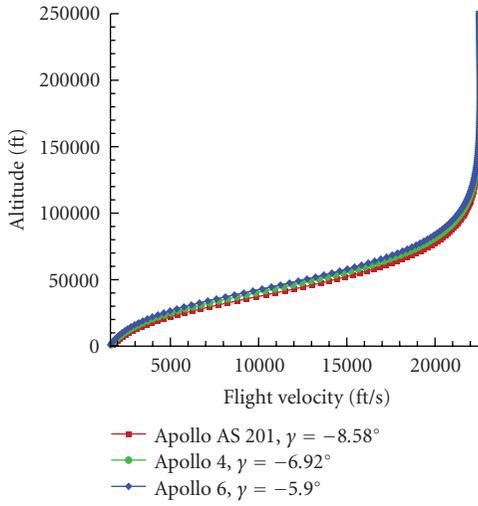
The *Sparta* design environment automatically links the 3-DOF trajectory code to appropriate empirical models of atmospheres depending on the planet that is selected to calculate pressure, density, temperature, Reynolds number, and speed of sound as a function of altitude. The computed trajectory data can be imported for plotting and further analysis. The vehicle's flight path angle and the free stream conditions are used at selected flight conditions along the trajectory to build customized grids tailored to capture the hypersonic reacting flow phenomena.

4.3. *Trajectory Data.* The initial entry velocity for *Apollo 4* was 36 545 ft/s at an atmospheric interface of 400 000 ft [6] and initial flight path angle of -6.93° [11]. The initial entry velocity used in *Sparta* is 22500 ft/s at an atmospheric interface of 250 000 ft and initial flight path angle of -12.00° and hence there is a difference in the Mach numbers, deceleration loads between the trajectory data, and the actual *Apollo 4* flight trajectory data. With respect to aeroheating, the key parameters in the ballistic trajectory simulation are the entry angle, entry velocity at atmospheric interface, and the vehicle mass. *Sparta* code has the capability to generate trajectory data for the chosen flight vehicle (from the GUI) by varying several parameters: the user can change, the nose radius, initial flight path angle, initial entry velocity, and so

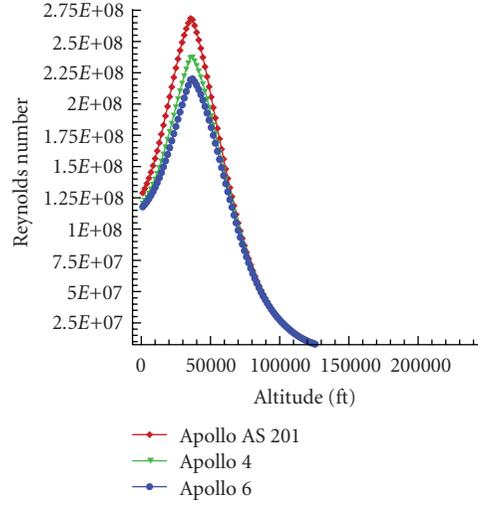
forth and study the change in trajectory because of change in the design parameters. This is intended to be a good research simulation and educational tool.

Apollo Capsules were chosen for trajectory analysis since they have identical vehicle dimensions and the plots 15(a)–15(f) show different trajectories since their initial flight path angles and ballistic coefficients are different for these flight vehicles. A sample trajectory data is shown in Table 3. Figures 15(a)–15(f) show the variation of flight velocity, Reynolds number, Stagnation point pressure, and deceleration against atmospheric altitude during re-entry. Three different flight vehicle trajectories have been analyzed for different values of flight path angle (γ) and ballistic coefficient (β). Figures 15(a)–15(f) show *Apollo AS201* with an initial flight path angle of -8.58° , *Apollo 4* with an initial flight path angle of -6.92° , and *Apollo 6* with an initial flight path angle of -5.9° . These plots also show different trajectories since their initial flight path angles and ballistic coefficients are different for these flight vehicles. The maximum deceleration appears at about 50 000 ft altitude close to the height of maximum stagnation point heating load of 700 BTU/ft²\second at 52 000 ft altitude.

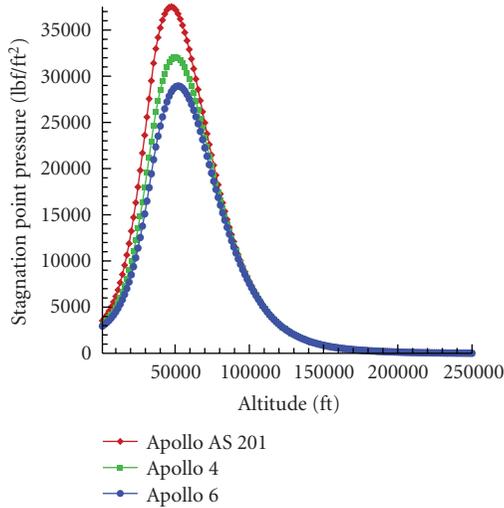
4.4. *Trajectory Data.* *Sparta* has been compared and evaluated against the industry standard tools like Program to Optimize Simulated Trajectories: POST [12], and Trajectory Optimization tool: TrajOpt. The POST and OTIS [13] tools are the products of many years of research and development. Although both have the capability of modeling most vehicles and trajectories and are based on tested and sound-proven algorithms, unfortunately their heritage code and legacy



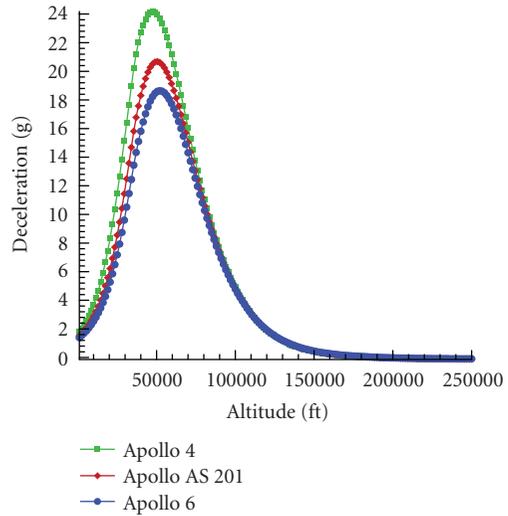
(a) Flight velocity with altitude



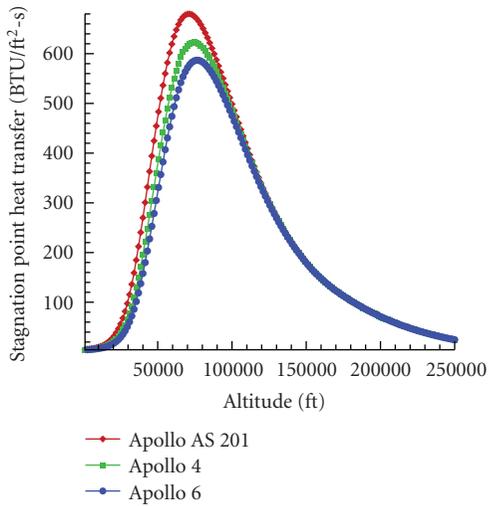
(b) Reynolds number versus altitude



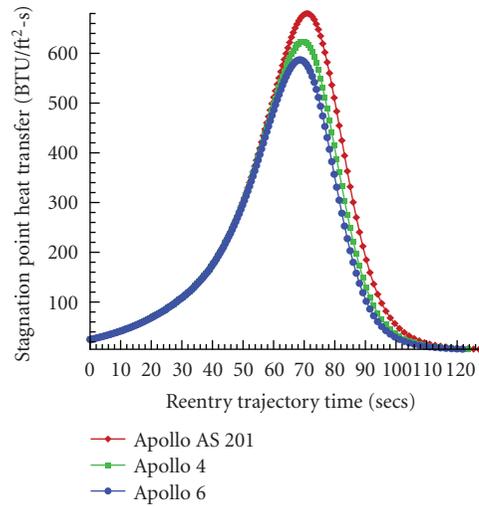
(c) Stagnation point pressure versus altitude



(d) Deceleration versus altitude



(e) Stagnation point heat versus altitude



(f) Stagnation point heat versus time

FIGURE 15: (a)–(f) Trajectory data: flight velocity, stagnation point heating, deceleration versus altitude.

TABLE 4: Pathfinder flight trajectory data reconstructed by *Sparta* trajectory engine.

Trajectory analysis for flight vehicle: MARS pathfinder probe						
Altitude (km)	Velocity (km/sec)	Flight path angle γ (deg)	Mach number	Freestream temperature (K)	Convective heat transfer rate \dot{q}_{conv} (W/cm ²)	Radiative heat transfer rate \dot{q}_{rad} (W/cm ²)
60.20	7.40	9.56	40.00	130.11	58.10	1.40
55.80	7.34	9.26	37.20	147.45	69.80	2.30
50.09	7.16	8.84	34.60	163.78	85.40	3.80
45.32	6.89	8.48	33.20	164.06	98.30	5.20
40.05	6.38	8.09	30.10	172.01	100.10	3.60
35.09	5.59	7.74	26.4	172.26	85.7	2.09

structures make them unwieldy to wrap them within a CFD solver. They are primarily coded in FORTRAN and they use long detailed name lists to generate an input deck for the trajectory engine to parse. POST and OTIS are linked to gradient-based optimizers that generally require good initial guesses of the design variables to produce meaningful solutions. This burdens the users and hinders the autonomous execution especially when linked with CFD flow solvers. We know from experience with these tools and other commercially available trajectory software that trajectory calculations consume a significant analysis cycle time because optimization is inherently computationally expensive and many optimizations fail or produce accurate results requiring user intervention to rectify the optimization.

Sparta trajectory engine was designed with this mind; hence the trajectory calculations are completely autonomous and driven from a GUI. The trajectory engine is hooked to an OLAP database for multidimensional analysis capability [5]. OLAP is an Online Analytical Processing database that has a comprehensive list of atmospheric entry probes and their vehicle dimensions, trajectory data, aerothermal data, and material properties like carbon-, silicon-, and carbon-phenolic-based ablators. OLAP has the capability to run in one simulation several different trajectory conditions and the output is stored back into the database and the data can be queried for appropriate trajectory type. For instance, given any one planetary probe dimensions from the database, an OLAP simulation can be set up to run for three types of trajectory: *Nominal*, *Undershoot*, and *Overshoot trajectory* or all three types can be set up to run simultaneously using threads.

For the benchmarking analysis, *Sparta* trajectory data has been compared against POST and TrajOpt trajectory data for two different vehicles flown in two different atmospheres. *Apollo* capsule was chosen for Earth entry and Mars Pathfinder for Martian planetary entry. The flight vehicle configuration that was evaluated in POST, TrajOpt, and *Sparta* for the benchmark analysis was *Apollo 6* re-entry profile. Figures 16(a) and 16(b) show the re-entry flight velocity and Altitude envelope. Velocity at peak heat was found to be 8.32 km/s in *Sparta* calculations and around 8 km/s in the other trajectory tools that were used for benchmarking for the *Apollo 6* capsule as shown in Table 5. For the Martian pathfinder, velocity at peak heating was found to be 6.5 km/s with a maximum convective heat flux

of 101.56 W/cm² as shown in Table 6. Figure 16(b) compares altitude versus velocity for all the trajectory codes in comparison against the actual pathfinder flight data which is based on the onboard accelerometer measurements [14]. The trajectory tools seem to agree with the reconstructed flight data and with the trajectory data generated by *Sparta* on a machine to machine specific architecture code comparison.

POST uses a more traditional direct shooting approach that calculates the state variables as a function of time throughout the entire trajectory and this guarantees that the physics of the problem are accurate at all times during the simulation. This makes POST a perfect candidate for benchmarking against *Sparta*.

5. Convective and Radiative Stagnation Point Heating Analysis

5.1. Fay-Riddell, Sutton-Grave, and Mike Tauber Heating Correlations. Introducing the average specific heat, constant Lewis, and Prandtl-number ($Le = 1$ and $Pr = 0.71$), Fay and Riddell proposed an empirical correlation to define the stagnation point heat flux (*heating rate per unit area*) toward a fully catalytic, hot wall. Stagnation point heat transfer rate for Earth has been modeled using *Fay-Riddell* theory [10] and *Sutton-Grave* correlation [15] as described in (5) which gives the heat flux as

$$\dot{q}_{\text{conv}} = k \sqrt{\frac{\rho}{R_n}} \left(\frac{V_\infty}{1000} \right)^3, \quad (5)$$

where \dot{q}_{conv} is the convective heat-transfer rate into the flight body, per unit area, k is a constant based on the planetary atmosphere, ρ is the freestream density, and V_∞ is the flight velocity. The heating rate for \dot{q}_{conv} is in W/cm² if the velocity is given in m/s and the density in kg/m³. The maximum heating rate for an entry profile is evaluated at the stagnation point from (5) and one can see that the bigger the nose radius, the lower the heat rates: the total heat load of the mission is derived from the integration of the heat rates over the heat peak during entry. Radiative heat transfer is computed using the *Tauber-Sutton* Radiative heating correlation for Earth and Mars [16] as

$$\dot{q}_{\text{rad}} = C r_n^a \rho^b f(V), \quad (6)$$

where C is a constant based on the planetary atmosphere, R_n is the hemispherical nose radius in m , ρ is the freestream

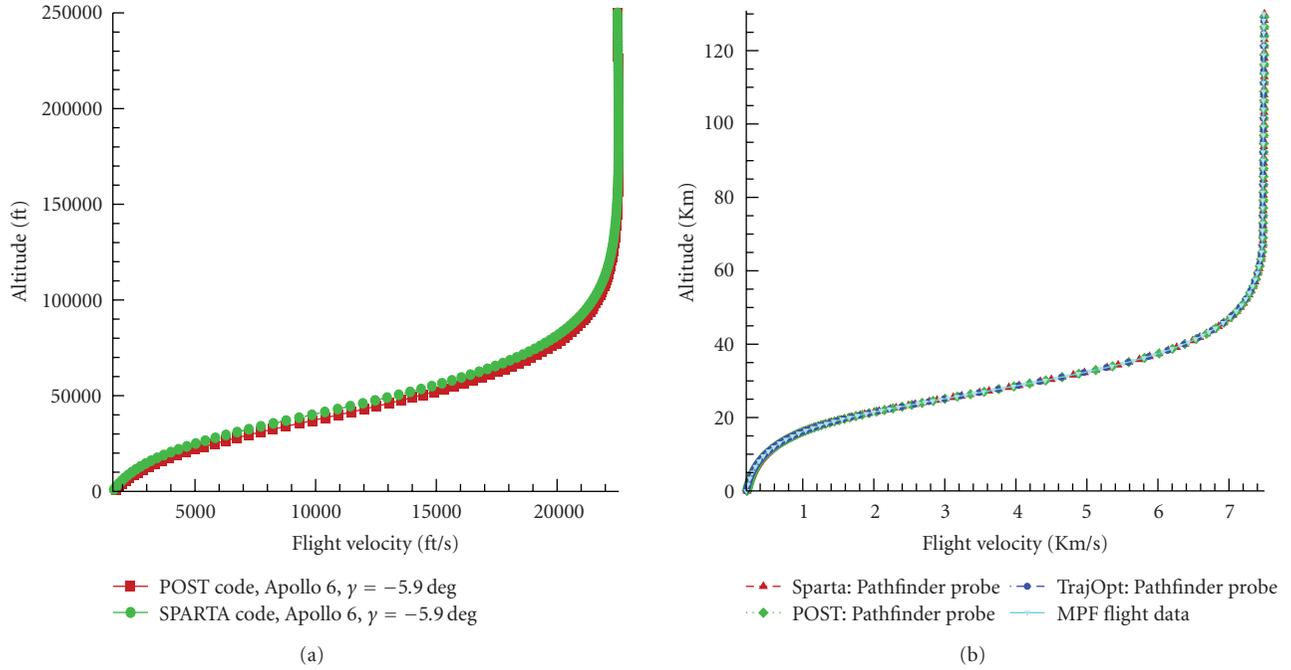


FIGURE 16: (a) SPARTA versus POST. (b) SPARTA versus POST and TrajOpt.

TABLE 5: Benchmarking analysis for *Apollo 6* at an entry angle of -5.9° .

Codes	Velocity at peak heating (km/s)	Altitude at peak heating (km)	Maximum deceleration (g's)	Mach number
POST	8.45	24.01	12.15	26.40
TrajOpt	7.98	23.00	11.16	24.93
<i>Sparta</i>	8.32	23.16	11.96	26.00

density in kg/m^3 , and $f(V)$ is a tabulated function of velocity given in [16]. The total stagnation point heat load is computed by adding the stagnation point convective and radiative components as

$$\dot{q}_{\text{tot}} = \dot{q}_{\text{conv}} + \dot{q}_{\text{rad}}. \quad (7)$$

When a probe re-enters a planetary atmosphere at hypersonic velocities, the temperature of the shock later formed around it may become sufficiently hot to emit a significant amount of radiation. The radiation emanates mainly from the inviscid region of the shock layer where the flow is highly or almost totally dissociated and significantly ionized, most strongly in the stagnation region. The radiation tends to cool the inviscid region. As the radiation passes through the cold boundary layer adjacent to the wall, the radiation is weakened by the absorption, and in return, the boundary layer is heated.

This energy balance occurs from the hot inviscid region to the cold boundary layer region of the shock layer, that is, toward the wall direction. A small segment of the radiative heat flux reaching the wall is reflected by the wall to an extent determined by the reflectance of the wall. The reflected radiation then causes radiative transfer in the direction from

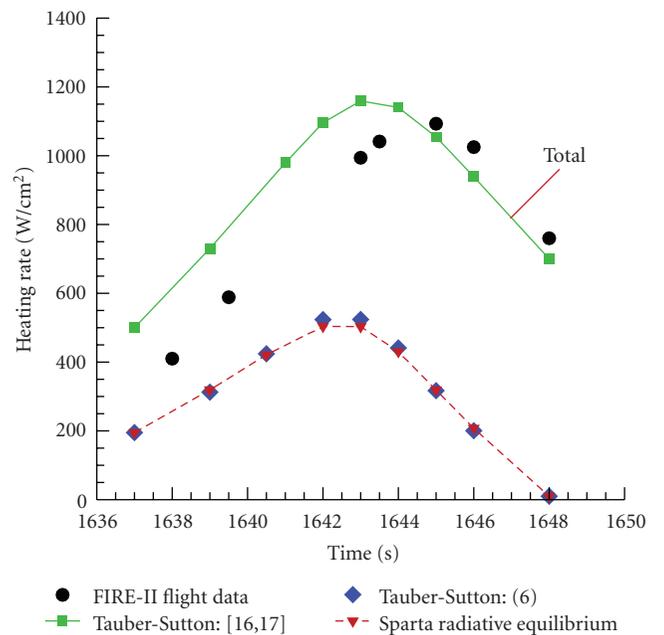


FIGURE 17: Radiative stagnation point heating of FIRE II data at a re-entry velocity of 11.37 km/s.

TABLE 6: Benchmarking analysis for mars pathfinder probe at a relative entry angle of -13.65° .

Codes	Velocity at peak heating (km/s)	Altitude at peak heating (km)	Maximum deceleration (g's)	Convective heat flux (W/cm ²)
POST	6.5	41.5	16.50	101.13
TrajOpt	6.7	39.5	17.80	119.79
<i>Sparta</i>	6.5	41.0	16.26	101.56

the wall to the shock wave, that is, toward the shock direction. When the magnitudes of the radiation are as large as to be comparable with the convective heat flux at the wall, the radiative transport phenomenon must be accounted for in the calculation of the heat load to the vehicle.

The plots in Figure 17 show the comparison of calculated radiative heating with FIRE II flight data and the engineering correlations described by Tauber in [16, 17]. The vehicle configuration for FIRE II is an axisymmetric, 66° sphere cone, and truncated spherical forebody, with a small corner radius as described in [18]. This trajectory point corresponds to the first FIRE II heatshield, with a nose radius of 0.935 m. The results from the radiative engineering correlations are tabulated in Table 7.

The magnitude of the radiative heating to a blunt entry vehicle is a function of velocity, altitude (density), and the shock-standoff distance. For the analysis performed for *Apollo* entries, the stagnation-point-standoff distance was assumed to vary as it would for a sphere. This distance determines the thickness of air in the shock layer that radiates at both nonequilibrium and equilibrium conditions.

5.2. Effective Nose Radius for Nonspherical Configurations.

The convective and radiative heating correlations of *Fay-Riddell* and *Sutton-Tauber* can be used to calculate the stagnation point heating rates only on a hemispherical nose: when $R_n = R_b$. When this is not the case, these correlations cannot be readily applied and an effective nose radius has to be determined to calculate the stagnation point heating rate on a blunt body. Effective radius R_{eff} is the equivalent hemispherical radius which will produce the same velocity gradient as that computed for the blunt body [19]. An effective nose radius has been determined for every probe at a *zero* angles of attack. For non-spherical configurations, an equivalent hemispherical nose radius is employed to calculate the stagnation-point values of the heat transfer coefficient as shown in [20] that gives the same shock standoff distances in an adiabatic flow. It has been observed by Ellison [21] that the effective nose radius R_{eff} decreases as the nose radius R_n decreases and as the corner radius R_c increases.

6. The Role of Heatshield in a Hypersonic Environment

6.1. The Hypersonic Environment. During its entry into a planetary atmosphere, a probe converts a large fraction of its kinetic energy into heat. Only a small part of this heat enters the probe, but this fraction of heat still amounts to a few hundreds or thousands of kW/m² at peak heating conditions.

Such higher fluxes require a dedicated thermal protection system (TPS). Maximum heat fluxes are encountered in the hypersonic part of the trajectory, typically at Mach 25, and in conditions where the flow is a continuum. In this situation, a strong shock wave in front of the vehicle dissociates the atmosphere, ionization and emission of thermal radiation occur. The heat shield's primary function is to preserve a payload from overheating. In certain cases, the shield is also used as a structural element. At hypersonic entries, the vehicle is surrounded by a flow of high temperature (typically more than 6000 K), high velocity plasma that is partially dissociated and ionized and whose degrees of freedom are excited (vibration and electronic). In addition the gas can be in chemical and thermal equilibrium, depending on thermodynamic and entry conditions and various couplings between the processes can appear. The heat transfer for a re-entry vehicle is characterized by the following mechanisms as described below.

The flow which is warmer than the TPS material diffuses and transfers heat by convection into the shield. A transition of the flow from laminar to turbulent increases this flux by 100–200% or sometimes more. Heat conducted through the wall is stored by conduction. Conduction is characterized by the thermal conductivity (κ) of the TPS material and storage of heat is characterized by the specific heat C of the material. The atomic species in the plasma tend to recombine on a relatively cold wall around 1000–2000 K. When they recombine, the atoms release their heat of recombination. The number of recombination atoms on the heatshield depends on the catalycity of the material, and on the diffusion of species in the plasma. Catalytic materials can experience heat fluxes three times higher than non catalytic ones placed in the same environment. The radiative heat flux is emitted by a high temperature gas and the corresponding flux heating the wall can still be high even when the radiative power emitted by the flow is low compared to its enthalpy, a situation where it is not necessary to account for a radiation source term into the flow energy equation. The wall of most TPS materials can be characterized as a black body with temperature T and emissivity ε which emits a flux $\varepsilon\sigma T^4$, where σ is the *Boltzmann* constant: $\sigma = 5.667 \cdot 10^{-8} \text{ Wm}^{-2}\text{K}^{-4}$. The wall reflects a fraction of the incoming radiative heat flux. The endothermic heat flux can be significant and in certain flight regimes, oxidation of the surface may occur. Full oxidation can be diffusion limited by the transport of available oxygen through the boundary layer.

Radiative heat shields require high emissivity and thermal diffusivity (ceramics and light ablators) and limit the heat input by reradiating a large fraction of the incident flux. Ablative heat shields (light medium and dense) will lose mass

TABLE 7: Stagnation point radiative heating rate comparisons for FIRE II.

V_e (km/s)	γ_e°	$m/(C_d)(S)$ (kg/m ²)	V_∞ (km/s)	\dot{q}_{conv} W/cm ²	\dot{q}_{rad} W/cm ²	\dot{q}_{tot} W/cm ²
11.37	-8.5	40	9.25	325	10	335
12.50	-10.0	40	10.47	510	108	618
14.26	-10.0	40	11.86	730	360	1090

to absorb the incoming heat. Reflective heat shields have high emissivity (ultra-pure silica fibers, Teflon). They simply deflect the incoming radiative heat flux. Heat sink heat shields (metals such as Beryllium) will absorb and distribute the heat (large specific heat and large conductivity). In practice, TPS materials combine these functions to a greater or lesser degree. The most commonly used technologies for very high heat fluxes greater than about 10 MW/m² are densely loaded phenolic ablators such as carbon and silica phenolic. Carbon phenolic was used in the US Pioneer Venus and Galileo Jupiter Probes and these materials are also used in large rocket nozzle applications. Surface recession through ablation can be a significant part of the original ablator thickness. The choice and selection of the material has a large impact on the mass of the shield.

6.2. Ablative and Nonablating TPS Materials. The TPS protects (insulates) a body from the severe heating encountered during hypersonic flight through a planetary atmosphere. Since TPS is a single point-of-failure subsystem, it is extremely critical and its performance needs to be validated through ground testing followed by rigorous analysis. Typically, there are two classes of TPS: *Reusable (or Nonablating) TPS*, where after exposure to the entry environment there are no changes in the mass or properties of the TPS materials. Typically, reusable TPS materials are limited to *relatively* mild entry environments (e.g., the space shuttle). The prime characteristics of the reusable TPS is that the radiative and convective heating results in a significant amount of energy being reradiated from the heated surface with the remainder conducted into the TPS material.

In contrast, the ablative TPS materials accommodate high heating rates and heat loads through phase change and mass loss. Ablative materials have been the classical approach to TPS used for over 40 years in a broad range of applications. Till date, all the NASA planetary entry probes have used ablative TPS materials. The characteristics of an ablative TPS material are illustrated in Figure 18. Reinforced composites employing organic resins as binders are used in most ablative TPS materials. When heated, the resin pyrolyzes producing gaseous products (*mostly hydrocarbons*) that percolate toward the heated surface and are injected into the boundary layer. Resin pyrolysis also produces a carbonaceous residue that deposits on the reinforcement. The resulting surface material is termed *char*. The pyrolysis process is typically endothermic and the pyrolysis gases are heated as they percolate toward the surface thus transferring some energy from the solid to the gas. The injection of the pyrolysis gases into the boundary layer alters the boundary layer properties, typically resulting in a reduction

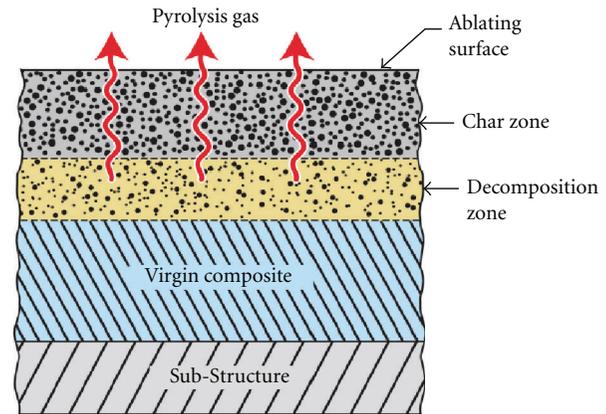


FIGURE 18: Decomposing ablator.

in convective heating. However, the gases may undergo chemical reactions with the boundary layer gases that will have an effect on the net heating to the surface. Furthermore, the phenomenon called *surface recession* is bound to happen and this is the effect of the chemical reactions between the surface material and boundary layer species that results in consumption of the surface material. These reactions can be endothermic (vaporization, sublimation) or exothermic (oxidation) and is bound to have a significant impact on net energy to the surface.

There are several classes of ablative materials and each class has its own prominent performance limitations. Typically, ablative TPS materials are categorized by density, that is, low density, mid density, and high density. Material strength increases with density, but so does the thermal conductivity. Consequently, materials selection for a given mission entry environment requires a balance between ablative and insulation efficiency while recognizing the optimal performance regime for each class of materials. When a material is used outside of its optimal zone, its performance is inefficient which leads to a nonminimal TPS mass fraction. It is that fraction of the entry probe mass that is devoted to TPS material which is called the TPS mass fraction.

The objective of the convective and radiative heating point calculation is to finally do the thermal protection system sizing for both ablating and nonablating materials. Before TPS sizing can be performed, four critical parameters must be evaluated: peak heat flux, heat load, peak deceleration, and peak dynamic pressure. A material thermal response model must be developed and validated with appropriate test data. Ablative heatshield materials are selected based on peak heat flux and dynamic pressure from the trajectory and the TPS thickness (mass) of the material stack is determined by integrating the total heat load.

For manned missions, peak deceleration is of significant importance because it might result in a major catastrophe if the deceleration loads exceed crew tolerance limits. For instance, the maximum deceleration upper limit for manned return to Earth from low Earth orbit (LEO) or lunar return is 10 Gs. On the other hand, for Martian atmospheric entry after long exposure to zero gravity, the upper limit is 4 Gs. Also, peak dynamic pressure plays a crucial role in the selection of the outermost TPS material if spallation is an issue.

In the *Sparta* code, whenever a trajectory simulation exceeds this maximum stagnation point pressure, the code alerts the user and aborts the simulation run. For *SLA-561V* (*Pathfinder* ablative TPS), the maximum stagnation point pressure is 25.332 kPa 0.25 Earth *atm* and maximum heat flux is around 300 W/cm². The *Pathfinder* aeroshell ablative material (*SLA-561V*) can maintain its physical structural integrity at stagnation point pressures of 25.332 kPa (0.25 Earth *atm*) or less. At stagnation point pressures exceeding 25.332 kPa, *surface spallation* of the aeroshell ablative material occurs, changing the aerodynamic characteristics of the aeroshell effectively, and creates an uncertainty in the heating profiles. Similarly for *PICA* (*Stardust* TPS material), the maximum heat flux is 1600 W/cm² (the stagnation point pressure 1.0 Earth *atm*). For carbon-phenolic (*Galileo* probe), the maximum heat flux is 35 000 W/cm² and 8 *atm*. Similarly, thermo-physical values were compiled for several different TPS materials and are available in the database.

This check point data exists in the database for all the TPS materials and *Sparta* fetches the maximum critical values for materials from the material properties database whenever an appropriate TPS material is chosen from the trajectory user interface and checks it during the simulation run to see if the trajectory exceeds this limit at any point in the trajectory run. Thus, trajectories have to be designed such that their peak values like peak stagnation pressure, peak heat load, and peak heat rate have to be less than the material properties critical values in the database for those materials that's chosen for the simulation. *Currently, no other trajectory code can do this kind of a db-driven trajectory simulation and material violation check.* Ignoring these values results in TPS material failure and TPS failure is a single point of mission failure. Currently, the heat shield materials used on a space vehicle such as space shuttle can withstand a total heat flux of the order of 50 W/cm² without causing ablation of the material. Advanced heat shield materials are believed to be able to withstand up to about 70 W/cm². When the sum of the convective and radiative heat fluxes exceeds such a maximum allowed heat flux value, the heat shield material ablates. In the ablation process, the heat energy is converted into the latent energy of vaporization.

6.3. Galileo Ablative TPS. The *Galileo* probe to Jupiter was the most challenging entry mission ever undertaken by NASA and is considered history's most difficult atmospheric entry. The probe employed a 44.86-degree blunt cone aeroshell as shown in Figure 19 and it entered the Jovian atmosphere at a velocity of ~47.4 km/s (atmosphere relative

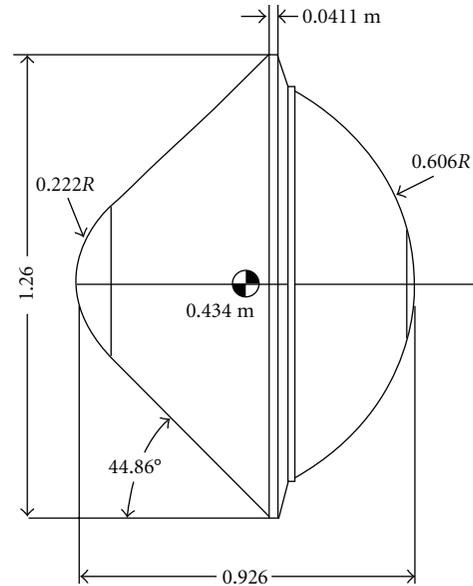


FIGURE 19: Galileo probe.

speed at 450 km above the 1 bar reference altitude). The forebody TPS employed fully dense carbon Phenolic ($\rho = 1450 \text{ kg/m}^3$) that, at the time, was the best ablator available. The entry environment was very severe and estimates of the peak heating (combined convective and radiative) were on the order of 35 kW/cm² with a total integrated heat load of ~200 kJ/cm². *Galileo* experienced a peak deceleration of about 230 G's (2.3 km/s²) and the peak stagnation point pressure before aeroshell jettison was 9 bars (900 kPa).

The shock layer maximum temperature peaked at approximately 16000 K (for comparison: the solar photosphere is merely 5800 K). Nearly, 26.5% of the *Galileo* probe's original entry mass of 335 kg was vaporized during the 70-second heat pulse as shown in Figure 20. The probe experienced a maximum ablation rate of 7.4 kg/s. The Forebody heat shield lost approximately 79 kg to ablation; approximately 90% of the loss occurred during 16 seconds of the heating pulse, mainly due to shock-layer radiation. Computational predictions showed that the boundary layer was predicted to be blown off the surface at peak heating and the convective heating approached zero. Total heat flux peaked at approximately 15000 W/cm². By way of comparison, the peak total heat flux experienced by the Mars *Pathfinder* aeroshell, the highest experienced by a successful Mars lander, was 106 W/cm², and the *Apollo-4* (AS-501) command module, re-entering at 10.77 km/s (atmosphere relative speed at 121.9 km altitude), experienced a peak total heat flux of 497 W/cm². Conservative design was employed in designing the *Galileo* Probe. In the case of the *Galileo* probe, the radiative heat flux and turbulent shock layer along with the TPS material response were not clearly understood because of the extreme state of the Probe's entry conditions. The ablative TPS material used was Carbon-Phenolic and it was earlier used for the Pioneer Venus Probes which were the design ancestors to the *Galileo* Probe. The TPS material

surface recession near the base of its frustum was actually greater than the predicted or expected values for the probe.

6.4. Materials Properties Database. Typically, there are two classes of ablating materials that are used for planetary probe's heatshield. Carbon- and silicon-based ablators. Phenolic impregnated carbon ablator (PICA) is an ablative material that uses the decomposition of the polymer to absorb energy and ablation mechanisms and reradiation to reject heat at the surface. Gas pyrolysis and surface ablation are complicated processes involving many physical phenomena. A comprehensive database of existing planetary probe designs is provided in the *Sparta* framework. Trajectory and geometry data, their vehicle dimensions, aerothermal data, and material properties like carbon-, silicon-, and carbon-phenolic-based ablator are stored for each probe in the database. In addition to the capsule shapes, base areas, nose radii, payload masses, and the ballistic coefficients of the probes are stored in the database. Figure 21 shows configurations generated from the database for the *Pathfinder*, *Viking*, and *Genesis* class vehicles. The user can modify the existing vehicle designs by changing geometric features available in the GUI.

A relational database management system has been developed to access the planetary probes database. The database extends across several tables. Tables 8 and 9, for instance, are linked through a common column *Flight Vehicle* as it is evident from the tables. Separate tables for the vehicle geometry, thermal protection system, as well as trajectory and aerodynamic data are included. The database manager allows selective data retrieval through user-entered queries. Materials properties like carbon-, silicon-, and carbon-phenolic-based ablators have been obtained from the NASA Ames Thermal Protection Materials and System Branch TPSX Internet Database [22] and *William-Donald's* thermo-physical property data [23] and modeled into the local *Sparta* OLAP cubes database and integrated with the user interface. Thus a dynamic TPS sizing approach is modeled. Pyrolysis, ablation, and recession are calculated for the forebody at every point in the trajectory for the material chosen from the user design interface.

For a hypersonic re-entry vehicle entering the Earth's atmosphere on return from a planetary mission, its entry velocity is hyperbolic, that is greater than the escape velocity of 11.3 km/sec. From the *Apollo* flights experience, we know that the hyperbolic entries will produce a massively ablating environment. For a Martian entry, for which the escape velocity is 5.1 km/sec, a massively ablating environment may occur if the entry speed is greater than 8 km/sec, and if the nose radius of the vehicle is large of the order of 3 m or larger. Thus, it is imperative to accurately model the ablation temperature for TPS materials.

6.5. TPS Sizing. Calculating thermal response of an entry vehicle's thermal protection system can be quite complex. A high-fidelity calculation of the thermal response using a program such as Aerotherm's Charring Material Thermal Response and ablation program or CMA [24] can be quite

involved, particularly when one considers the enormous amount of data required as input by the program. In particular, generating the surface chemistry input and the pyrolysis gas enthalpy requires that an additional equilibrium, or nonequilibrium chemistry code, be run. Extensive input deck has to be set up manually for these chemistry codes and also it requires sound knowledge of the composition of the material as well as the composition of the pyrolysis gas. To complete this analysis, a thermal response model must be developed and correlated to arc jet test data, a set of surface chemistry tables that encompass a wide range of surface conditions must be available, and a pyrolysis gas enthalpy model must be developed. If these models do not exist and the exact composition of the pyrolysis gas and the TPS material are not known, constructing and running a high-fidelity CMA model is problematic from both an accuracy and solution convergence standpoint. CMA is a one-dimensional material thermal response code with in-depth decomposition that solves the energy equation with pyrolysis gas effects.

In *Sparta*, the vehicle thermal response is calculated by an approximate method that uses heat of ablation data to estimate heat shield recession during entry. This analysis is coupled to a one-dimensional finite-difference calculation that determines in-depth thermal response. The one-dimensional finite-difference in-depth solution does not account for pyrolysis gas energy absorption through the material but it takes into account the thermal decomposition of the material. Since it is hooked to the *Sparta* framework, the inputs come from the trajectory data, including relative velocity, atmospheric density, pressure, and convective heat rate as a function of time. The approximate method then calculates radiative heating, recovery enthalpy, wall enthalpy, surface pressure, and heat transfer coefficient. Ultimately, it determines recession thickness, total thickness, and heat shield area mass based on thermal response at the stagnation point. A uniform thickness heatshield is modeled. In the existing atmospheric re-entry database system, a material database has been constructed and added for common ablative thermal protection and substructure materials. User-defined materials can be easily added to the database without having to modify the thermal protection systems subsection code of the *Sparta's* framework.

Stored constants for ablative materials include the decomposition kinetic constants used in the *Arrhenius* formulation for density decomposition, the resin volume fraction, the heats of formation, thermal conductivity, specific heat, emissivity, and heat of ablation curve fit constants. There are two components to the approximate technique presented here [25]. The first component makes use of a steady-state ablation assumption and employs the heat of ablation, or Q^* , to estimate recession during entry. The second component involves calculating the in-depth temperature response to predict the amount of material required as insulation to keep the bond-line temperature below a specified limit. Calculating the in-depth temperature response is accomplished using a finite-difference formulation for the in-depth conduction through the material. The recession rate at any instant in time can be calculated by (8) by using the heat of ablation. The total recession is

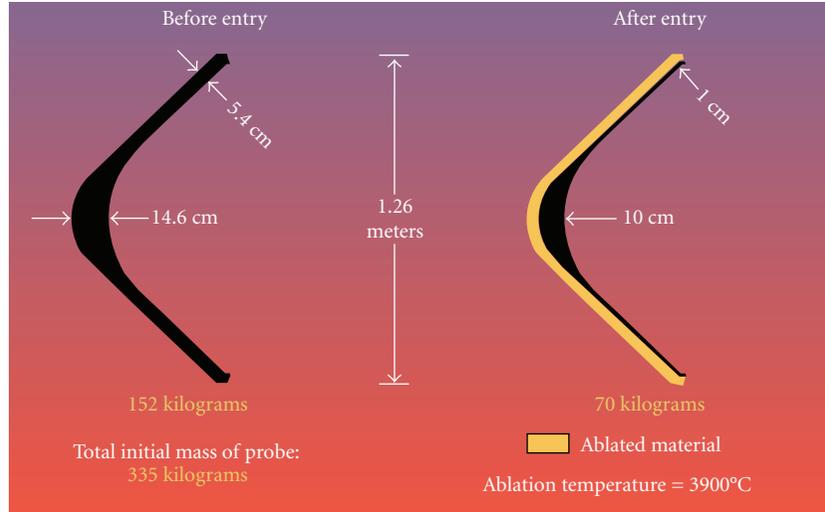


FIGURE 20: Galileo probe heat shield ablation: NASA picture.

TABLE 8: Database of entry vehicles and flight vehicle geometric dimensions.

Flight vehicles	Configuration (shape)	Nose radius (m)	Base area (m ²)	Length (m)	Diameter (m)	Mass (Kg)
<i>Apollo AS-202</i>	Capsule 33° Cone	4.69	12.02	3.62	3.91	—
<i>Apollo 4</i>	Capsule 33° Cone	4.69	12.02	3.62	3.91	5424.9
<i>Viking I</i>	70° Sphere Cone	0.88	9.65	1.66	3.54	980.0
<i>Pathfinder</i>	70° Cone	0.66	5.52	1.508	2.65	585.3
<i>Genesis</i>	59.81° Blunt Cone	0.43	12.02	0.93	1.51	201.0

then found by integrating the recession rate over the entire trajectory. This formulation is conservative and will generally overpredict recession rate:

$$\dot{S} = \frac{\dot{Q}^{hw}}{\rho Q^*}. \quad (8)$$

In (8), \dot{S} is the recession rate, \dot{Q}^{hw} is the hot wall heat flux, ρ is the material density, and Q^* is the heat of ablation. The hot wall heat flux can be calculated from

$$\dot{Q}^{hw} = \dot{Q}_{cw} \left(1 - \frac{H_w}{H_r}\right). \quad (9)$$

The stagnation point recovery enthalpy H_r shown in (9) is calculated from the following:

$$H_r = \frac{V^2}{2} + h_\infty - h_{ref} + \sum \Delta H_{fj}^0. \quad (10)$$

The free stream enthalpy (h_∞) for air is calculated based on the equilibrium properties for air based on curve fits by Srinivasan and Tannehill [26]. With the recovery enthalpy and the cold wall heat rate, the convective heat flux can be calculated as

$$\rho_e u_e C_H = \frac{\dot{Q}^{hw}}{H_r - H_w}. \quad (11)$$

The process of calculating the recovery enthalpy, heat transfer coefficient, and stagnation-point pressure is performed at every incremental time step in the trajectory.

The surface pressure is calculated using modified Newtonian theory (as explained in (1) and (2), where the pressure at the stagnation point is given by

$$P_{stag} = P_\infty + \frac{1}{2} \rho_\infty V^2 C_{p_{max}}. \quad (12)$$

The cold wall heat flux can be substituted for the hot wall heat flux in (11). Equation (13) shows the one-dimensional heat conduction along with the surface energy balance:

$$\rho C_p \frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial x} \left(\kappa A \frac{\partial T}{\partial x} \right), \quad (13)$$

$$\dot{q}_{conv} + \alpha \dot{q}_{rad} - \dot{q}_{cond} - \varepsilon \sigma T_w^4 = 0, \quad (14)$$

where T is the temperature, T_w is the surface temperature, κ is the thermal conductivity, x is measured from TPS surface, ρ is the instantaneous material density, C_p is the material specific heat, \dot{q}_{conv} is the convective heat flux, \dot{q}_{rad} is the radiative heat flux, ε is the material emissivity, α is the material absorptivity, and σ is the Stephen-Boltzmann's constant.

This formulation in (14) neglects various forms of chemical fluxes entering the surface as well as the pyrolysis energy rate and the net energy absorbed due to pyrolysis gas movement through the material in the in-depth solution. The material decomposition, or the change in density, is computed explicitly as if it were a material property. The change in density as a function of temperature is computed

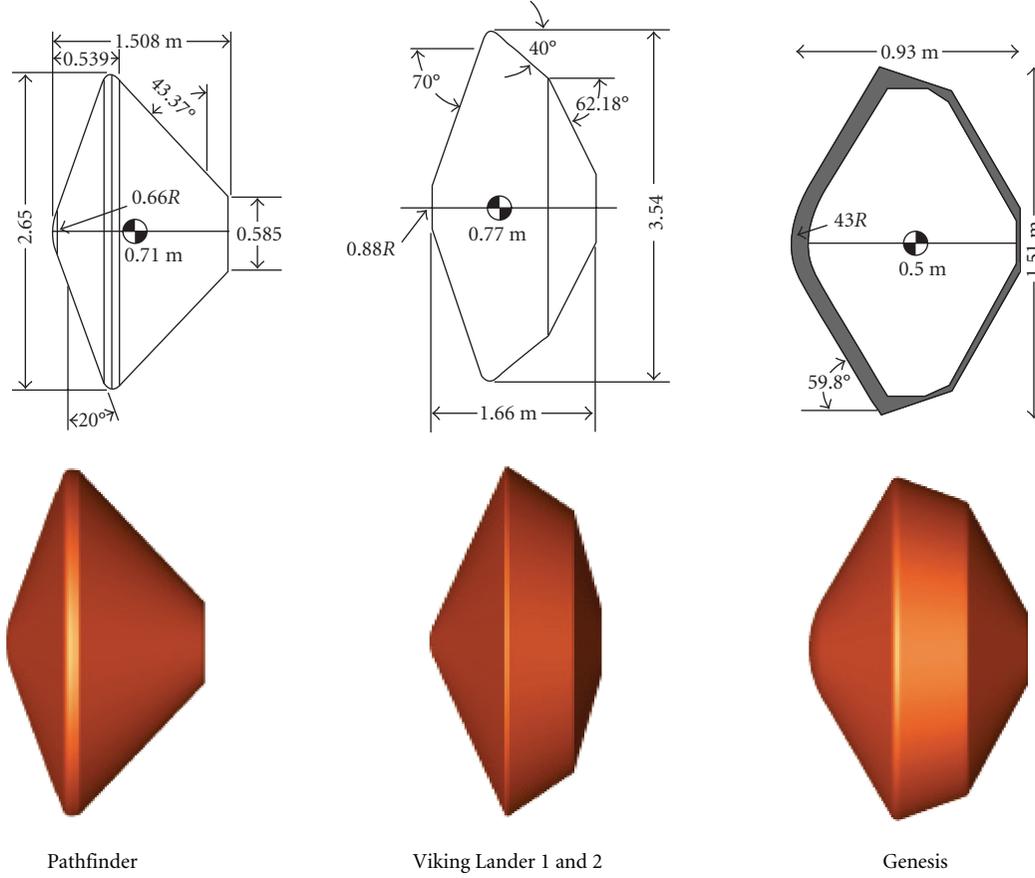


FIGURE 21: Probe database configurations and geometry constructions.

TABLE 9: Database of entry vehicles and flight entry trajectory data.

Flight vehicles	Inertial entry velocity (km/s)	Relative entry velocity (km/s)	Peak heat velocity (km/s)	Inertial entry angle (degrees)	Control method
<i>Apollo AS-202</i>	—	8.29	7.77	—	Roll modulation
<i>Apollo 4</i>	11.14	10.73	10.25	-6.92	Roll modulation
<i>Viking Lander 1</i>	4.61	4.42	4.02	-16.99	3-axis RCS
<i>Pathfinder</i>	7.26	7.48	6.61	-14.06	Ballistic
<i>Genesis</i>	11.00	10.80	9.2	-8.00	Ballistic

using the same Arrhenius equation used by CMA [24], which is shown in (15). A three-component model for the density decomposition is used in this formulation. All of the parameters for (15) are contained in the thermophysical material property database:

$$\frac{\partial \rho_i}{\partial t} = -B_i \exp\left(-\frac{Ea_i}{RT}\right) \rho_{o_i} \left(\frac{\rho_i - \rho_{r_i}}{\rho_{o_i}}\right)^{\psi_i}. \quad (15)$$

The implicit discretization of the one-dimensional heat conduction equation is performed using a finite-volume or

control-volume technique [27]. The results of the discretization for the node at the surface and any interior node are, respectively,

$$\begin{aligned} & \left(1 + \frac{2k\Delta t}{\rho C_p \Delta x_i^2}\right) T_i^{n+1} - \frac{2k\Delta t}{\rho C_p \Delta x_i^2} T_{i+1}^{n+1} \\ & = T_i^n + \dot{q}_{\text{rad}} \frac{2\Delta t}{\rho C_p \Delta x_i} + \dot{q}_{\text{conv}} \frac{2\Delta t}{\rho C_p \Delta x_i} - \varepsilon \sigma T_i^{n4} \frac{2\Delta t}{\rho C_p \Delta x_i}, \\ T_i^n & = -\frac{k\Delta t}{\rho C_p \Delta x_i^2} T_{i-1}^{n+1} + \left(1 + \frac{2k\Delta t}{\rho C_p \Delta x_i^2}\right) T_i^{n+1} \\ & \quad - \frac{k\Delta t}{\rho C_p \Delta x_i^2} T_{i+1}^{n+1}. \end{aligned} \quad (16)$$

TABLE 10: Stagnation Point TPS Recession thickness and in-depth thermal response comparison.

Material	Thickness type	Trajectory type	Methodology		% difference
			CMA	<i>Sparta</i> approximate	
PICA	Recession	Lunar return (11 kms)	0.731	0.759	3.830
		Lunar return (14 kms)	3.823	1.19	—
	Without recession	Lunar return (11 kms)	7.608	7.747	1.827
		Lunar return (14 kms)	7.125	7.623	6.989
	Total	Lunar return (11 kms)	8.104	8.243	1.715
		Lunar return (14 kms)	10.675	9.14	—

TABLE 11: Material and thermal properties of carbon and silicon-based ablators.

Material	Material type	Density (ρ) (kg/m ³)	Thermal conductivity (κ) (watt/m ² ·k)	Specific heat (C_p) (J/kg ² ·k)
ATJ	Carbon-based	1729.2	1179.69	711.76
Avcoat-5026-39	Silicon based	512.6	25.00	1110.0
RCC	Reusable CC	1650.2	120.0	1312.0

Equation (16) can be assembled into a banded, tri-diagonal matrix and solved using the *Thomas algorithm* [28] for the nodal temperatures.

Table 10 shows a comparison between the *Sparta* approximate heat of ablation technique and CMA solution. For PICA, the recession threshold temperature is 1500 K and the approximate techniques' solution used this threshold temperature value. The results show very good agreement between the *Sparta* approximate heat of ablation technique and CMA. The biggest discrepancy, as conjectured, is in the recession prediction; however, the recession predictions for the approximate technique are on the conservative side with the exception of the PICA, 14 km/s case. This 14 km/s entry case has a significantly higher surface pressure at the time of peak heating, and PICA's ablation performance is highly sensitive to pressure. But *Sparta's* approximate heat of ablation solution is independent of pressure. As such, it underpredicts the recession. Comparisons also show calculated thicknesses without the recession (the insulation required to maintain the bondline temperature above 250°C). This is essentially a comparison of the in-depth thermal responses, that is, without recession.

PICA is a mid-to-low-density ablator, and under these entry conditions, a large amount of ablation and pyrolysis will occur. Looking at the data in Table 10, it shows that the in-depth thermal responses between CMA and the approximate technique do not compare as well. This is a direct consequence of the approximate solution neglecting the effects of pyrolysis energy absorption, pyrolysis gas convection through the solid, and surface chemistry effects. While neglecting these physical effects, the approximate technique's in-depth solution is fairly good and is certainly within the accuracy required for conceptual design and preliminary analysis. Table 11 shows the thermal and material properties for RCC and Avcoat-5026-39 as described in [20]. Because of the high thermal conductivity (κ) of the nose cap wall and the thinness of the wall when compared to the radius of the nose cap, the *Biot* number is small

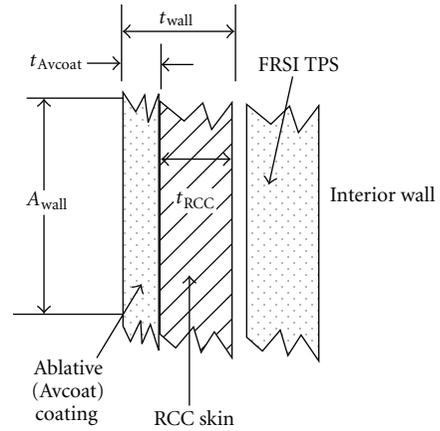


FIGURE 22: Heat shield wall of the re-entry vehicle.

and the nose cap wall is effectively modeled as a “lumped-mass” structure with constant temperature throughout. The material database contains carbon-, silicon-, and carbon-phenolic-based ablators which are populated in the graphical user interface. During the probe design phase, the user can select an ablative heat shield from the list populated in the user interface.

The wall thickness is defined as the depth of the ablative layer as shown in Figure 22 and the RCC thickness and the average head capacity of the wall is the thickness weighted heat capacity of the ablative and RCC wall layers. The “lumped-mass” temperature approximation for this wall segment can be written as

$$\dot{q}_{in} - \dot{q}_{out} = [\rho_{Avcoat} C p_{Avcoat} t_{Avcoat} + \rho_{RCC} C p_{RCC} t_{RCC}] \dot{T}_w, \quad (17)$$

where \dot{q}_{in} is the heat flux into the wall, \dot{q}_{out} is the heat flux out of the wall, \dot{T}_w is the time rate of change of the wall temperature, and the remaining terms in the brackets are the average heat capacity of the wall segment. The only way to

significantly vary the heat capacity is to change the material thickness because the values of specific heat and density are thermal properties of the material.

7. Summary and Conclusions

A trajectory-based, platform-independent engineering level analysis, OLAP database-driven Java GUI-based application, has been developed for aerothermal investigation of re-entry vehicles. The code calculates all flight conditions along the trajectories including aerodynamic and aerothermodynamic stagnation point quantities. A fourth-order *Runge-Kutta* integration is employed for trajectory calculations. In addition, a comprehensive database of existing planetary probe designs has been developed and integrated with the 3-DOF GUI tool. The Flight vehicle OLAP database has been modeled and linked to the planetary probe application in DOTNET to be accessed from the web as well. Appropriate atmospheric profiles have been developed and integrated with the code. For Venus, Mars, and Neptune, the GRAM models have been used. Empirical correlations for estimating the stagnation point hypersonic aerodynamic convective and radiative heat transfer have been modeled using *Fay-Riddell* and *Tauber-Sutton* correlations. Radiative heating calculations are compared with the FIRE II flight data. For nonspherical configurations, equivalent hemispherical nose radii are calculated that gives the same shock standoff distances in an adiabatic flow. A one-dimensional in-depth finite difference model is employed for ablation calculations for the TPS sizing. Comparative trajectory data analysis was done using the industry standard tools like POST—Program to Optimize Simulated Trajectories and TrajOpt—Trajectory Optimization tool. *Sparta* has been extensively benchmarked successfully against the industry standard trajectory codes. The industry standard benchmarking tools seem to agree with the results of *Sparta* thereby validating the tool and the tool works as expected for preliminary design and analysis. *Sparta* framework also employs an in-built application programming interface (API) to convert trajectory data into a formatted KML file that is used by Google Earth for simulating Earth-entry fly-by trajectory visualizations.

Nomenclature

C_p	Specific heat at constant pressure (kJ/kg · K)
C_v	Specific heat at constant volume
h	Enthalpy (J/kg)
K_n	Knudsen number
Le	Lewis number
m	Entry mass (Kg)
M	Mach number
P	Pressure (N/m ²)
P_r	Prandtl's number
\dot{q}_{conv}	Convective heat transfer rate W/cm ²
\dot{q}_{rad}	Radiative heat transfer rate W/cm ²
\dot{q}_{tot}	Total heat transfer rate W/cm ²
Q^*	Heat of ablation
R_b	Fore body radius (m)

R_c	Corner radius
R_e	Reynolds number
R_{eff}	Effective radius
R_n	Nose radius
R_s	Shoulder radius
S	Base area of the capsule (m ²)
\dot{S}	Recession rate
t	Time (sec)
T	Temperature (K)
V_∞	Freestream flight velocity (m/s)

Greek

α	Angle of attack, Material absorptivity
ρ	Freestream density (kg/m ³)
β	Ballistic Coefficient (kg/m ²)
Υ	Ratio of specific heats
ε	Surface emissivity of the material
σ	Stefan-Boltzmann's constant
γ	Flight path angle (°)
θ	Surface area inclination angle

Subscripts

b	forebody
c	corner
conv	convective
e	entry
n	nose
rad	radiative
tot	total
w	wall
∞	freestream

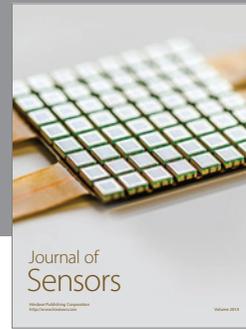
Acknowledgments

Most of the research work in this paper was carried out in the *Center of Excellence for Space Transportation and Exploration* and I would like to thank Dr. Periklis Papadopoulos for his invaluable support and guidance in this research. I would like to record my thanks to Sanjeev Gupta of Intel Corporation for many helpful suggestions and recommendations in implementing the OLAP database and schema that enabled the multidimensional analysis capability. His enthusiasm in extending ideas on modeling the DOTNET framework for planetary probe flight vehicles application is appreciated much. Also, I would like to thank Aleta Duvall of the Marshall Space Flight Center for providing the Global Reference Atmospheric Models (NeptuneGRAM, VenusGRAM, MarsGRAM, and TitanGRAM) for the atmospheric profiles.

References

- [1] C. Davies, "Planetary Mission Entry Vehicles," Quick Reference Guide, Ver. 2.1, NASA-ARC, 2002.
- [2] Program Development Corp., "GridPro/az3000 User's Guide and Reference Manual," 300 Hamilton Avenue, Suite 409, White Plains, NY 10601, 1996.

- [3] P. Papadopoulos and P. Subrahmanyam, "Trajectory based automatic grid generation tool for atmospheric entry CFD modeling," in *Proceedings of the 9th International Conference on Numerical Grid Generation in Computational Field Simulations*, pp. 190–201, San Jose, Calif, USA, June 2005.
- [4] C. G. Justus and D. L. Johnson, "Mars Global Reference Atmospheric Model 2001 Version (MARS-GRAM 2001) Users Guide," NASA/TM-2001-210961, April 2001.
- [5] P. Subrahmanyam, "Multidimensional aerothermodynamic analysis for planetary probes using OLAP cubes," in *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, Keystone, Colo, USA, August 2006, AIAA-2006-6274.
- [6] J. E. Pavlosky and St. L. G. Leger, "Apollo experience report—thermal protection subsystem," Tech. Rep. NASA TN D-7564, NASA, Washington, DC, USA, January 1974.
- [7] P. Papadopoulos and P. Subrahmanyam, "Web-based computational investigation of aerothermodynamics of atmospheric entry vehicles," *Journal of Spacecraft and Rockets*, vol. 43, no. 6, pp. 1184–1190, 2006.
- [8] "TecPlot® Version 9 User's Manual," Amtec Engineering, Inc., P. O. Box 3633, Bellevue, WA 98009-3633.
- [9] Google Earth, <http://earth.google.com>.
- [10] J. Fay and F. Riddell, "Theory of stagnation point heat transfer in dissociated air," *Journal of Aeronautical Sciences*, vol. 25, no. 2, 1958.
- [11] E. R. Hillje, "Entry aerodynamics at lunar conditions obtained from the flight of Apollo 4 (AS 501)," Tech. Rep. NASA TN D-5399, NASA, Washington, DC, USA, October 1969.
- [12] G. L. Brauer, D. E. Cornick, and R. Stevenson, "Capabilities and applications of the program to optimize simulated trajectories (POST)," Tech. Rep. NASA CR-2770, NASA, Washington, DC, USA, February 1977.
- [13] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.
- [14] D. A. Spencer, R. C. Blanchard, R. D. Braun, P. H. Kallemeyn, and S. W. Thurman, "Mars Pathfinder entry, descent, and landing reconstruction," *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 357–366, 1999.
- [15] K. Sutton and R. A. Graves, "A general stagnation point convective heating equation for arbitrary gas mixtures," Tech. Rep. NASA TR-376, NASA, Washington, DC, USA, 1971.
- [16] M. E. Tauber and K. Sutton, "Stagnation-point radiative heating relations for earth and mars entries," *Journal of Spacecraft and Rockets*, vol. 28, no. 1, pp. 40–42, 1991.
- [17] M. E. Tauber, "A review of high-speed, convective heat transfer computation methods," Tech. Rep. NASA TP-2914, NASA, Washington, DC, USA, July 1989.
- [18] D. L. Cauchon, "Radiative heating results from the FIRE II flight experiment at a reentry velocity of 11.4 kilometers per second," Tech. Rep. NASA TM X-1402, NASA, Washington, DC, USA, July 1967.
- [19] E. V. Zoby and E. M. Sullivan, "Effects of corner radius on stagnation-point velocity gradients on blunt axisymmetric bodies," Tech. Rep. NASA TM X-1067, NASA, Washington, DC, USA, 1965.
- [20] R. K. Crouch and G. D. Walberg, "An investigation of ablation behavior of Avcoat 5026/29M over a Wide Range of Thermal Environments," Tech. Rep. NASA TM X-1778, NASA, Washington, DC, USA, 1969.
- [21] J. C. Ellison, "Experimental stagnation-point velocity gradients and heat-transfer coefficients for a family of blunt bodies at Mach 8 and angles of attack," Tech. Rep. NASA TN D-5121, NASA, Washington, DC, USA, April 1969.
- [22] NASA Ames Thermal protective Materials and System Branch, TPSX, Ver 4.1, August 2005, <http://tpsx.arc.nasa.gov>.
- [23] S. D. Williams and M. C. Donald, "Thermal protection materials: thermophysical property data," Tech. Rep. NASA RP-1289, NASA, Washington, DC, USA, 1992.
- [24] Anon, "User's manual, aerotherm charring material thermal response and ablation program," Acurex UM-87-13/ATD, Aerotherm Division, Acurex Corporation, Mountain View, Calif, USA, November 1987.
- [25] P. Subrahmanyam, "Development of an interactive hypersonic flow solver framework for aerothermodynamic analysis," *Engineering Applications of Computational Fluid Mechanics Journal*, vol. 2, no. 4, pp. 436–455, 2008.
- [26] S. Srinivasan, J. C. Tannehill, and K. J. Weilmuenster, "Simplified curve fits for the thermodynamic properties of equilibrium air," Tech. Rep. NASA RP-1181, NASA, Washington, DC, USA, 1987.
- [27] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, Philadelphia, Pa, USA, 2nd edition, 1997.
- [28] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, New York, NY, USA, 2nd edition, 1988.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

