NASA/TM-2009-215950

A Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing (SAPE)

Jamshid A. Samareh
Langley Research Center, Hampton, Virginia

November 2009
Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA’s STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA’s mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at [http://www.sti.nasa.gov](http://www.sti.nasa.gov)
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to: NASA STI Help Desk NASA Center for AeroSpace Information 7115 Standard Drive Hanover, MD 21076-1320
A Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing (SAPE)

Jamshid A. Samareh
Langley Research Center, Hampton, Virginia

November 2009
The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.
Abstract

SAPE is a Python-based multidisciplinary analysis tool for systems analysis of planetary entry, descent, and landing (EDL) for Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Titan. The purpose of SAPE is to provide a variable-fidelity capability for conceptual and preliminary analysis within the same framework. SAPE includes the following analysis modules: geometry, trajectory, aerodynamics, aerothermal, thermal protection system, and structural sizing. SAPE uses the Python language—a platform-independent open-source software—for integration and for the user interface. The development has relied heavily on the object-oriented programming capabilities that are available in Python. Modules are provided to interface with commercial and government off-the-shelf software components (e.g., thermal protection systems and finite-element analysis). SAPE runs on Microsoft Windows and Apple Mac OS X and has been partially tested on Linux.

Nomenclature

\( c \) = gas mass fraction
\( C_D \) = drag coefficient
\( C_H \) = heat transfer coefficient
\( C_p \) = pressure coefficient
\( D \) = drag, N
\( g \) = gravity, m/s²
\( h \) = height above planet, m
\( H \) = ellipsled height, m
\( K \) = heating constant
\( L \) = lift, N
\( m \) = mass, kg
\( M \) = molecular mass
\( q \) = stagnation-point heating rate, W/m²
\( r \) = radius, m
\( R \) = body radius, m
\( s \) = distance downrange, m
\( t \) = time, s
\( u \) = velocity, m/s
\( V \) = velocity, m/s
\( W \) = ellipsled width, m
\( \beta \) = backshell half-cone angle, deg
\( \theta \) = forward shell half-cone angle, deg
\( \gamma \) = flight path angle, deg
\( \rho \) = density, kg/m³

Subscripts

\( b \) = base
\( c \) = convective heating
\( e \) = boundary layer edge
\( n \) = nose
\( p \) = planet
\( r \) = radiative heating
\( s \) = probe shoulder
I. Introduction

Systems analysis of a planetary entry, descent, and landing (EDL) is a multidisciplinary activity in nature. The purpose of the analysis is to gain a better understanding of various entry system concepts and their limitations. Systems analysis teams typically include one or more systems engineers and discipline-specific experts in flight mechanics, aerodynamics, aerothermodynamics, structural analysis, and thermal protection systems (TPS). Other discipline experts, for example, propulsion experts, are consulted as needed. The systems analysis process may take from several weeks to several years (references 1 and 2 provide a good summary of typical systems analysis activities). Integrated tools, like SAPE, improve the performance of the systems-analysis team by automating and streamlining the process, and this improvement can reduce the errors that stem from manual data transfer among discipline experts. The role of discipline experts in the systems-analysis process is indispensable and cannot be replaced by any tool.

Several efforts have been made to develop integrated tools with a focus on different aspects of planetary EDL. For example, the integrated design systems (IDS) initiative developed an integrated tool by using a common gateway interface (CGI) to integrate codes for various disciplines from a distributed network. The IDS program used the Perl script for integration. Allen et al. provide more details on the IDS design and implementation. The Hyperprobe program is another IDS-like Web-based implementation, with the addition of a relational database management system. Unlike IDS and Hyperprobe, the Planetary Entry Systems Synthesis Tool (PESST) is based on Matlab.*

The key goals for SAPE development are:
1) Perform EDL systems analysis for any planetary body with an atmosphere.
2) Operate cross-platform (i.e., Windows, Mac, and Linux operating systems).
3) Use existing software components.
4) Use open-source software to avoid software licensing issues.
5) Perform low-fidelity systems analysis in one hour on a computer that is comparable to an average laptop.
6) Keep discipline experts in the analysis loop.

The next section discusses the use of the Python language for tool integration. Section III presents the design structure matrix for planetary EDL analysis. Section IV reviews the eight planetary bodies with atmospheres. Sections V through X introduce the six major modules that are included in SAPE. The last section presents recommendations for future enhancements.

II. Use of Python for Integration Environment

SAPE uses the Python† language to develop and integrate various modules. Python is a free, object-oriented, high-level programming language that is often used as a scripting language. Python is open source and is supported by a large community and managed by the Python Software Foundation. Python is available for most computer platforms and has a large number of standard libraries, which range from scientific to Web applications. Python users include

* [http://www.mathworks.com](http://www.mathworks.com)
† [http://www.python.org](http://www.python.org)
Google™, Yahoo®, YouTube™, Disney, the New York Stock Exchange, the European Organization for Nuclear Research (CERN), and many other large commercial entities. The United Space Alliance has used Python to streamline space shuttle-mission design.*

III. EDL Design Structure Matrix (DSM)

The multidisciplinary EDL problem can be decomposed into a set of key disciplines. These discipline areas, modules, can be represented in matrix form using the DSM approach. The matrix is a graphical approach for representing the interdependencies among these various modules. The DSM is a square matrix with the analysis modules positioned along a main diagonal. Table 1 shows a typical DSM representation for the integrated EDL analysis. For each analysis module shown on the DSM diagonal, relevant outputs are listed in the corresponding row; the inputs are listed in the corresponding column. For example, the required inputs for structural analysis are the outer mold line (OML), mass distribution, and loads. Structural analysis outputs include an estimate for the structural mass as well as displacements. DSM analysis tools, such as DeMAID,7 can be used to analyze the DSM and identify the critical dependencies.

The EDL integrated analysis process is a complex system, and SAPE currently includes only the six most important disciplines: geometry, trajectory, aerodynamics, aerothermodynamics, TPS, and structural sizing.

IV. Planetary Bodies with Atmosphere

Eight planetary bodies in our solar system have an appreciable atmosphere; these are Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Titan. The data that are included in SAPE for all eight planetary bodies are based on the NASA Goddard Planetary Fact Sheet† (see table 2). These bodies can be divided further into small bodies (Venus, Earth, Mars, and Titan) and large gas giants (Jupiter, Saturn, Uranus, and Neptune). Planetary entry characteristics8 for each body are listed in table 3.

SAPE users can either use the nominal atmospheric profiles9 that are built into SAPE or provide the atmospheric profile in a tabular form. Details on the nominal atmospheric profiles are available for Earth, Mars, Neptune, Titan, and Venus in Justus et al.10 and for Jupiter, Neptune, Saturn, and Uranus in Justus et al.11 (see figure 1.)

---

* [http://www.python.org/about/success/usa](http://www.python.org/about/success/usa) (last visited on February 09, 2009)
<table>
<thead>
<tr>
<th>Optimization module</th>
<th>Database module</th>
<th>Geometry module</th>
<th>Pack’g &amp; mass pro</th>
<th>Trajectory analysis</th>
<th>Propulsion analysis</th>
<th>Aero analysis</th>
<th>Aerothermal analysis</th>
<th>TPS sizing</th>
<th>Structural analysis</th>
<th>Stability and control</th>
<th>Cost analysis</th>
<th>Reliability analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization module</td>
<td>Design variables</td>
<td>Geometry variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
</tr>
<tr>
<td>Database module</td>
<td>Obj &amp; const.</td>
<td>Database module</td>
<td>Geometry variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
<td>Design variables</td>
</tr>
<tr>
<td>Geometry module</td>
<td>TBD</td>
<td>Geometry module</td>
<td>OML</td>
<td>Parameters</td>
<td>Flow path</td>
<td>OML</td>
<td>OML</td>
<td>OML</td>
<td>Dimensions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pack’g &amp; mass pro</td>
<td>Weight breakdown</td>
<td>Pack’g &amp; mass pro</td>
<td>Dry mass</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trajectory analysis</td>
<td>Trajectory profile</td>
<td>Fuel weight</td>
<td>Trajectory analysis</td>
<td>Flight conditions</td>
<td>Flight conditions</td>
<td>Flight conditions</td>
<td>G-load</td>
<td>Inertia loads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Propulsion analysis</td>
<td>TBD</td>
<td>Propulsion weight</td>
<td>TSFC</td>
<td>Propulsion analysis</td>
<td>Propulsion environment</td>
<td>Propulsion heating</td>
<td>Propulsion loads</td>
<td>Propulsion loads</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerodynamics analysis</td>
<td>TBD</td>
<td>Aerodynamics weight breakdown</td>
<td>Integrated aero</td>
<td>Aero analysis</td>
<td>Pressure distribution</td>
<td>Aero load</td>
<td>Distributed loads</td>
<td>Integrated aero</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerothermal analysis</td>
<td>TBD</td>
<td></td>
<td>Heating corrections</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPS Sizing</td>
<td>TBD</td>
<td>TPS weight</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TPS Sizing</td>
<td>TPS Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural analysis</td>
<td>TBD</td>
<td>Structural weight</td>
<td>Structural displacements</td>
<td></td>
<td></td>
<td></td>
<td>Sizing data</td>
<td>Structural analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stability and Control</td>
<td>Control margins</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Stability and control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost analysis</td>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cost analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability analysis</td>
<td>Reliability index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reliability analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Columns are inputs, and Rows are outputs
Table 2. Planetary Body Characteristics

<table>
<thead>
<tr>
<th>Planets</th>
<th>Earth</th>
<th>Mars</th>
<th>Titan</th>
<th>Venus</th>
<th>Neptune</th>
<th>Jupiter</th>
<th>Saturn</th>
<th>Uranus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius, km</td>
<td>6378</td>
<td>3397</td>
<td>2575</td>
<td>6052</td>
<td>24764</td>
<td>71492</td>
<td>60268</td>
<td>25559</td>
</tr>
<tr>
<td>Gravity, m/s²</td>
<td>9.8</td>
<td>3.7</td>
<td>1.35</td>
<td>8.9</td>
<td>11</td>
<td>23.1</td>
<td>9</td>
<td>8.7</td>
</tr>
<tr>
<td>Surface atmo. ρ kg</td>
<td>1.22</td>
<td>0.02</td>
<td>1.55</td>
<td>65</td>
<td>0.45*</td>
<td>0.16*</td>
<td>0.19*</td>
<td>0.42*</td>
</tr>
<tr>
<td>Atmo. height, km</td>
<td>8.5</td>
<td>11.1</td>
<td>10</td>
<td>15.9</td>
<td>20</td>
<td>27</td>
<td>59.5</td>
<td>27.7</td>
</tr>
<tr>
<td>Atmo. interface, km</td>
<td>120**</td>
<td>120**</td>
<td>800**</td>
<td>130**</td>
<td>630**</td>
<td>395</td>
<td>750</td>
<td>555</td>
</tr>
<tr>
<td>Atmo. mole. wt.</td>
<td>28.870</td>
<td>43.501</td>
<td>28.269</td>
<td>43.450</td>
<td>2.373</td>
<td>2.219</td>
<td>2.081</td>
<td>2.271</td>
</tr>
</tbody>
</table>

* at one bar

** based on refs. 9-11 at $10^{-9} < \rho < 10^{-8}$

Table 3. Planetary Entry Characteristics

<table>
<thead>
<tr>
<th>Composition</th>
<th>Entry speeds</th>
<th>Heat rates and loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small planets: Venus, Earth, Mars, Titan</td>
<td>Co₂, N₂, O₂</td>
<td>5–12 km/sec</td>
</tr>
<tr>
<td>Gas giant planets: Jupiter, Saturn, Uranus, Neptune</td>
<td>H₂, He</td>
<td>25–60 km/sec</td>
</tr>
</tbody>
</table>

Figure 1. Nominal atmospheric profiles.¹⁰⁻¹¹
V. Geometry Module

The geometry module is written entirely as an object in Python, and three types of geometries can be created: sphere-cone probes, tori, and asymmetric ellipsleds.

SAPE can model most existing sphere-cone probes (See appendix A.) The creation of sphere-cone probe geometry uses five flags and a set of geometry parameters. The flags are used to define various probe sections: spherical nose (0, 1), conical forward section (0, 1), shoulder fillet (0, 1), aft-body conical section (0, n), and back-end section (1). The numbers in parentheses indicate the possible number of individual sections. The possible back-end shapes are flat, spherical cap, ellipsoid cap, and fillet. Figure 2(a) shows a sample sphere-cone model, which consists of a spherical nose, a conical forward section, a shoulder fillet, three aft-conical sections, and a flat backend (flags: 1, 1, 1, 3, and flat). Additional information and figures on sphere-cone geometry models are given in appendix A.

The ellipsled is an elliptically blunted cylinder that was originally proposed by Muth et al.\textsuperscript{12} for an Earth-return aeroapture application. Edquist et al.\textsuperscript{13} present a modified version of the ellipsled in which the nose is defined by two ellipsoids with different minor axes. The modified ellipsled is defined by five parameters (figure 2(b)): total length, nose length, width, height, and offset ratio. The geometry module can also generate tori, which are donut-like shapes that are defined by major and minor radii (figure 2(c)).

\begin{align*}
\text{Offset} &= \frac{R_{\text{up}} - R_{\text{bottom}}}{H} \\
\text{ratio}_i &= \frac{L_i}{\sum L_i}
\end{align*}

(a) Sphere cone. \hspace{1cm} (b) Ellipsled. \hspace{1cm} (c) Torus.

Figure 2. Available geometry shapes and parameters.
VI. Aerodynamics

SAPE currently includes two aerodynamic models: an analytical expression for aerodynamic drag at a zero angle of attack and DACFREE code. The analytical expression for the drag of a sphere cone\(^{14}\)

\[
C_D = C_{p_{\text{max}}} \left[ \sin^2 \theta + \frac{R_n}{2R_b} (1 - 2 \sin^2 \theta \cos^2 \theta - 4 \sin^4 \theta) \right]
\]  

(1)

The DACFREE\(^{15}\) code has been used in many planetary entry studies and uses standard free-molecular and modified Newtonian methods. The geometry that is input is a set of triangles with appropriate boundary conditions. Python was used to wrap the DACFREE code and integrate it into SAPE. Figures 3 shows comparisons of integrated DACFREE results with those of Moss et al.\(^{16}\) (top), Moss et al.\(^{17}\) (center), and Edquist et al.\(^{13}\) (bottom).

VII. Aerothermodynamics

SAPE currently includes a simple analytical expression for stagnation-point heating. However, the plan is to include the Langley Approximate Three-Dimensional Convective Heating (LATCH) code\(^{18}\), which is an approximate three-dimensional heating code that is based on the axisymmetric analog for general three-dimensional boundary layers.

SAPE currently uses the analytical expression for stagnation heating rate that was developed by Sutton and Graves\(^{19}\):

\[
q = KV^3 \sqrt{\frac{\rho}{R_n}}
\]  

(2)

where \(V\) is the velocity, \(\rho\) is the atmospheric density, and \(R_n\) is the nose radius. What makes Sutton and Graves’s analytical expression unique is that the constant \(K\) is derived for a general gas mixture. The coefficient \(K\) is a function of the atmospheric mass fractions, the molecular weights, and their individual transport properties. Sutton and Graves provide two expressions for \(K\):

\[
K = \frac{0.1106}{\sqrt{\sum \frac{c_i}{M_i \gamma_i}}}, \quad K = \frac{1}{\sqrt{\sum \frac{c_i}{K_i^2}}}
\]  

(3)

where \(c_i\) are the mass fractions. The parameters \(\gamma_i\) that are used in Eq. (3) are the transport parameters of the base gas of a mixture and should not be confused with the flight-path angle that is used in the trajectory section (other constants are presented in table 4).
Figure 3. Comparisons of DACFREE results with those in references 13, 16, 17.
Tauber and Sutton\textsuperscript{20} present a simple procedure to approximate the radiative heating for Earth and Mars. Both the Sutton/Graves and the Tauber/Sutton procedures provide only approximations and should be corrected with available higher fidelity numerical solutions and experimental results.

The plan is to include TSCAAPE\textsuperscript{21}, which is a script that is tailored to perform aeroheating analysis for sphere cones using LAURA\textsuperscript{22} computer code. The script currently supports Earth, Mars, Titan, Venus, Saturn, and Neptune.

### VIII. Trajectory

SAPE includes a simple trajectory code. Work is underway to include POST-2,\textsuperscript{23} which is a more sophisticated trajectory code. The simple trajectory code is based on equations for planar flight,\textsuperscript{24} as shown in figure 4.

\[
\begin{align*}
\frac{dV}{dt} &= -\frac{D}{m} - g \sin \gamma, \quad \frac{V}{dt} = \frac{L}{m} - \left(\frac{V^2}{r}\right) \cos \gamma, \\
\frac{ds}{dt} &= \frac{R_p}{r} V \cos \gamma, \quad \frac{dr}{dt} = \frac{dh}{dt} = V \sin \gamma,
\end{align*}
\]

where \( g = g_s \left( \frac{R_p}{R_p + r} \right)^2 \).

Readers should consult Griffin and French\textsuperscript{24} for additional information. The above equation set is a system of first-order differential equations with its initial values—\( V, h, \) and \( \gamma \)—defined at atmospheric entry.

![Figure 4. Planer flight.](image-url)
SAPE uses the Cash and Karp Runge-Kutta method with the adaptive step size. This particular method is fifth-order accurate and more efficient by a factor of two than standard step-doubling methods. The method accuracy is controlled by defining a maximum error, and the method adjusts the time step to the user-defined level of accuracy. Users also can force a minimum time step to create a larger number of data points along the trajectory. A simple, one-dimensional minimization algorithm based on Ridder’s approach is included for the aerocapture applications. The Ridder algorithm is a variation of the false-position root finding approach with superlinear convergence property.

The trajectory code implementation is in C++ for efficiency, and the code was verified against several test cases. Figure 5 shows a comparison of the results from the simple trajectory code with those of POST-2, and the results are in excellent agreement.

The second set of test cases was based on the aerocapture study of Hall and Lee, in which shallow and steep entries into four planetary bodies (Mars, Neptune, Titan, and Venus) were analyzed. Table 5 compares the results from SAPE trajectory with those from Hall and Lee, and the results are in good agreement.

![Figure 5. Comparison with POST (symbols).](image)

(a) L/D = 0

(b) L/D = 0.25
IX. Thermal Protection System (TPS)

SAPE currently provides an interface to the TPS sizing code FIAT. The actual FIAT code is not part of SAPE and should be obtained directly from NASA Ames Research Center. Necessary inputs to FIAT includes the recovery enthalpy $H_r$, the heat flux term $\rho C_v$, pressure $p$, and radiative heat flux $q_r$. Trajectory and aerothermal modules generate the free stream density $\rho_\infty$, velocity $V_\infty$, convective heat flux $q_c$, and radiative heat flux $q_r$ variables. Figure 6 shows a sample TPS result, which is for a Stardust-class vehicle with a ballistic coefficient of 60 kg/m², an $L/D$ of 0.2, an entry velocity of 12 km/s, and a flight-path angle of $-10^\circ$. The heat rate includes a margin of 30%. Figure 6(a) shows the resulting heat rates and the total heat load, and figure 6(b) shows the back wall temperature and the recession rate for Phenolic Impregnated Carbon Ablator (PICA) concept. FIAT calculated the optimal thickness to be 2.8 cm.

<table>
<thead>
<tr>
<th></th>
<th>Peak ballute temperature (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hall et al.</td>
</tr>
<tr>
<td>Titan (shallow)</td>
<td>1030.0</td>
</tr>
<tr>
<td>Titan (steep)</td>
<td>1073.0</td>
</tr>
<tr>
<td>Neptune (shallow)</td>
<td>1257.0</td>
</tr>
<tr>
<td>Neptune (steep)</td>
<td>1369.0</td>
</tr>
<tr>
<td>Venus (shallow)</td>
<td>908.0</td>
</tr>
<tr>
<td>Venus (steep)</td>
<td>974.0</td>
</tr>
<tr>
<td>Mars (shallow)</td>
<td>805.0</td>
</tr>
<tr>
<td>Mars (steep)</td>
<td>868.0</td>
</tr>
</tbody>
</table>

Figure 6. Heating and TPS sample results for a Stardust-class probe.

(a) Heat rate and heat load.  
(b) Wall temperature and recession rate.

Table 5. Comparison SAPE Results with Results of Hall and Lee
X. **Finite-Element Analysis and Sizing**

SAPE is capable of analyzing and sizing certain classes of planetary probes. Automating the generation and the analysis of the finite-element model (FEM) are quite challenging. These tasks can be simplified by restricting the analysis to sphere-cone shapes. The current FEM topology that is used in SAPE is based on the existing structural topology, which consists of the shell, the external support structure, and the internal payload support. Figure 7 shows various structural topologies for a generic probe.

![Figure 7. Sample structural topologies for Pathfinder class probe.](image)

After generating the FEM, finite-element analysis (FEA) codes are used to automatically size the vehicle. The FEA code provides estimates for the structural mass for given launch and entry loads. Figure 8 shows the von Mises stresses on a typical aeroshell structural model.

![Figure 8. Sample FEA result.](image)
XI. Summary

SAPE is a Python-based multidisciplinary analysis tool that has been developed for systems analysis for planetary entry, descent, and landing (EDL) for Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Titan. SAPE includes the following analysis modules: geometry, trajectory, aerodynamics, aerothermal, thermal protection system, and structural sizing. The geometry module can model most existing sphere-cone probes, ellipsleds, and tori. The aerodynamic module includes an analytical expression for drag and DACFREE code. The aerothermal module is based on Sutton-Graves approximation, and the plan is to include the Langley Approximate Three-Dimensional Convective Heating (LATCH) code. SAPE includes a simple trajectory code, which is based on equations for planer flight. The work is underway to include POST-2. SAPE currently provides an interface to the TPS sizing code FIAT. SAPE is capable of analyzing and sizing certain classes of planetary probes. The current FEM topology that is used in SAPE is based on the existing sphere-cone structural topology, which consists of the shell, the external support structure, and the internal payload support. SAPE modules have been validated against existing tools; however, the integrated system has not been fully validated yet. One major short-term goal is to verify the current integrated system. The medium-term goals are to

1. Integrate an updated version of the FEA script discussed in section X.
2. Integrate POST-2.
3. Include sensitivity analysis capability.
4. Include system-based Monte Carlo simulation capability.
5. Include LATCH and TSCAPE codes.

XII. Acknowledgement

The author would like to acknowledge Jim Hoffman for his work on FEM generation, Alireza Mazaheri for developing TSCAPE, Aaron Olds for developing scripts for POST-2, and Erik Tyler for his help with FEM sizing (all from Analytical Mechanics Associates, Inc.).
Appendix A: Possible Sphere-Cone Shapes

Figure A1 shows the existing sphere-cone probes that are presented by Davies and Arcadi. The numbers that are shown below the name of each probe are the SAPE geometry flags, as used in figure 2(a). Table A1 shows geometry parameters for shape shown in figure A1.

Figure A1. Existing sphere-cone probes.
(s) spherical nose, (c) conical section, and (f) fillet
<table>
<thead>
<tr>
<th>Model</th>
<th>$r_s$</th>
<th>$L/r_s$</th>
<th>$r_a/r_s$</th>
<th>$r_b/r_s$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>Backshell Conic Length Ratios</th>
<th>Backend Shapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARD</td>
<td>1.4</td>
<td>1.457</td>
<td>2.4</td>
<td>0.1</td>
<td>0</td>
<td>33, 90, 0</td>
<td>0.75, -0.31, 0.25</td>
<td>Flat</td>
</tr>
<tr>
<td>Apollo</td>
<td>1.955</td>
<td>1.85166</td>
<td>2.4</td>
<td>0.1</td>
<td>0</td>
<td>33</td>
<td>1</td>
<td>Fillet</td>
</tr>
<tr>
<td>Beagle</td>
<td>0.45</td>
<td>1.11</td>
<td>0.93</td>
<td>0.0644</td>
<td>70</td>
<td>47, 47</td>
<td>0.5, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>Fire-II</td>
<td>0.336</td>
<td>1.564</td>
<td>2.78</td>
<td>0</td>
<td>0</td>
<td>90, 0, 90, 33</td>
<td>0.04, 0.04, 0.89</td>
<td>Flat</td>
</tr>
<tr>
<td>Galileo Probe</td>
<td>0.63</td>
<td>1.47</td>
<td>0.35</td>
<td>0</td>
<td>44.86</td>
<td>0, 60, 0</td>
<td>0.07, 0.07, 0.07</td>
<td>Ellipsoid</td>
</tr>
<tr>
<td>Genesis</td>
<td>0.75</td>
<td>1.267</td>
<td>0.57</td>
<td>0.044</td>
<td>60</td>
<td>20, 60</td>
<td>0.5, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>HAYABUSA</td>
<td>0.202</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>45</td>
<td>45, 90, 45</td>
<td>0.79, -0.06, 0.21</td>
<td>Flat</td>
</tr>
<tr>
<td>Huygens</td>
<td>1.35</td>
<td>0.73</td>
<td>0.926</td>
<td>0</td>
<td>60</td>
<td>90</td>
<td>-0.6</td>
<td>Ellipsoid</td>
</tr>
<tr>
<td>MER</td>
<td>1.325</td>
<td>1.138</td>
<td>0.5</td>
<td>0.05</td>
<td>70</td>
<td>46.63</td>
<td>1</td>
<td>Flat</td>
</tr>
<tr>
<td>Microprobe</td>
<td>0.175</td>
<td>1.561</td>
<td>0.5</td>
<td>0.05</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIRKA</td>
<td>0.5</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>135</td>
<td>0</td>
<td>1</td>
<td>Flat</td>
</tr>
<tr>
<td>MSL</td>
<td>2.25</td>
<td>1.116</td>
<td>0.5</td>
<td>0.05</td>
<td>70</td>
<td>36.9, 59</td>
<td>0.5, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>OREX</td>
<td>1.7</td>
<td>0.888</td>
<td>0.794</td>
<td>0</td>
<td>50</td>
<td>0, 75, 60</td>
<td>0.1, 0.7, 0.2</td>
<td>Flat</td>
</tr>
<tr>
<td>PAET</td>
<td>0.457</td>
<td>1.4</td>
<td>1</td>
<td>0</td>
<td>55</td>
<td></td>
<td></td>
<td>Ellipsoid</td>
</tr>
<tr>
<td>Pathfinder</td>
<td>1.325</td>
<td>1.066</td>
<td>0.5</td>
<td>0.05</td>
<td>70</td>
<td>49, 49</td>
<td>0.5, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>Reentry-F</td>
<td>0.3965</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>57</td>
<td></td>
<td></td>
<td>Flat</td>
</tr>
<tr>
<td>Stardust</td>
<td>0.4135</td>
<td>1.347</td>
<td>0.5625</td>
<td>0.05</td>
<td>60</td>
<td>30, 30</td>
<td>0.5, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>Venus Pioneer Probe</td>
<td>0.38</td>
<td>1.342</td>
<td>0.5</td>
<td>0</td>
<td>45</td>
<td>0, 90</td>
<td>0.3, -0.15</td>
<td>Ellipsoid</td>
</tr>
<tr>
<td>Venus Pioneer Sounder</td>
<td>0.71</td>
<td>1.465</td>
<td>0.5</td>
<td>0</td>
<td>45</td>
<td>90, 90, 0, 60</td>
<td>-0.12, -0.04, -0.04, 0.5</td>
<td>Flat</td>
</tr>
<tr>
<td>Viking</td>
<td>1.75</td>
<td>0.9381</td>
<td>0.5</td>
<td>0.015</td>
<td>70</td>
<td>39, 62.2</td>
<td>0.5076, 0.4924</td>
<td>Flat</td>
</tr>
</tbody>
</table>

**References**

A Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing (SAPE)

SAPE is a Python-based multidisciplinary tool for systems analysis of planetary entry, descent, and landing (EDL) for Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Titan. The tool is partially developed, and the goal is to provide a variable-fidelity capability for conceptual and preliminary analysis within the same framework. SAPE currently includes the following analysis modules: geometry, trajectory, aerodynamics, aerothermal, thermal protection system, and structural sizing. SAPE uses the Python language—a platform-independent open-source software—for integration and for the user interface. The development has relied heavily on the object-oriented programming capabilities that are available in Python. Modules are provided to interface with commercial and government off-the-shelf software components (e.g., thermal protection systems and finite-element analysis). SAPE currently runs on Microsoft Windows and Apple Mac OS X and has been partially tested on Linux.

EDL; MDO; python; system analysis