

Development of Trajectory Simulation Capabilities for the Planetary Entry Systems Synthesis Tool



AE8900 MS Special Problems Report
Space Systems Design Lab (SSDL)
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

Author
Devin Matthew Kipp

Advisor
Dr. Robert D. Braun
Space Systems Design Lab (SSDL)

May 17, 2005

Development of Trajectory Simulation Capabilities for the Planetary Entry Systems Synthesis Tool

Devin M. Kipp

Georgia Institute of Technology, Atlanta, GA, 30332

Dr. Robert Braun

Faculty Advisor, Georgia Institute of Technology, Atlanta, GA, 30332

A Simulated Trajectory Analysis code for Planetary Entry (STAPLE) has been developed as a contributing disciplinary analysis tool within the framework of a larger entry systems synthesis tool. STAPLE performs 3-degree of freedom trajectory analysis for atmospheric flight at arbitrary planetary bodies. The tool is applicable to both ballistic and lifting entry trajectory simulations and includes a modest guidance capability that enables aerocapture and controlled entry simulations. Event modeling capabilities include staging, parachute inflation, parachute release, and propulsive descent based on a gravity turn guidance law. Performance has been validated against existing analysis codes, Mars Pathfinder flight experience, and Aeroassist Flight Experiment studies.

Nomenclature

ϕ = planetocentric latitude	h = planetocentric altitude
μ = gravitational parameter	I_{sp} = specific impulse
ω = planetary rotation rate	L = specific lift
A = reference area	q = dynamic pressure
a_0 = reference sound speed	r = radial distance to planet center
a_T = thrust acceleration	r_0 = planetary surface radius
C_D = aerodynamic drag coefficient	T = thrust
C_L = aerodynamic lift coefficient	V = relative velocity
D = specific drag	γ = flight path angle
dt = integration time step	θ = planetocentric longitude
g = local gravitational acceleration	ρ = local atmospheric density
g_0 = surface gravitational acceleration	ρ_0 = reference density at surface
G_h = gain on altitude rate term	σ = bank angle
G_q = gain on dynamic pressure term	φ = off-nadir angle
H = atmospheric scale height	ψ = azimuth angle

I. Introduction

A Simulated Trajectory Analysis tool for Planetary Entry (STAPLE) has been developed as a contributing analysis within the framework of a much larger Planetary Entry Systems Synthesis Tool (PESST). STAPLE is responsible for analyses pertaining to the planetary trajectory simulation. Within the scope of applicability for future PESST applications, STAPLE is capable of performing both ballistic and lifting entry trajectory simulations at arbitrary planetary bodies. STAPLE also includes a modest guidance capability that enables aerocapture and guided entry simulations.

Initial versions of PESST utilized the NASA developed Program to Optimize Simulated Trajectories (POST) to perform trajectory analysis. POST proved to be a viable option but is subject to governmental restrictions under the International Traffic in Arms Regulations (ITAR) that limit its distribution and use, particularly to and by foreign nationals. As an alternative to POST, STAPLE increases the simplicity, utility, and speed of PESST.

In addition to its compatibility with PESST, STAPLE has utility as a standalone tool to perform offline trajectory analysis. As such, a graphical user interface has been developed to assist inexperienced users. STAPLE can also be run in a script input mode that is faster for experienced users and can be easily wrapped into integrated design environments like ModelCenter.

II. Theory and Implementation

STAPLE is developed within the framework of MATLAB 7.0 to take advantage of its mathematical utility, ease of programming, and built-in visualization capabilities. The internal workings of STAPLE are outlined in this section.

A. Atmosphere and Gravity Models

STAPLE utilizes a simplified spherical model to approximate the target body. The gravity model is specified by the surface radius (r_0) and gravitational parameter (μ) of a target body, from which the force of gravity at the surface (g_0) is determined. Local gravitational acceleration is determined throughout the trajectory as a function of planetocentric altitude (h) according to an inverse square gravity model:

$$g = g_0 \left(\frac{r_0}{r_0 + h} \right)^2 \quad (1)$$

The atmosphere of the target planet can be modeled according to an exponential atmosphere model or through tabular data input. An exponential model is specified by inputting surface density (ρ_0) and atmospheric scale height (H) of the target planet. Local atmospheric density is then determined throughout the trajectory as a function of altitude according to:

$$\rho = \rho_0 e^{\left(\frac{-h}{H} \right)} \quad (2)$$

Alternately the atmospheric density profile can be input in tabular form either using a POST formatted atmospheric input file utilizing a monovariant table of density vs. altitude or using a simple comma delimited file of similar format. In both cases the local atmospheric density is determined by natural log - linear interpolation of table values. The POST formatted input option was developed for ease of importing pre-existing atmospheric models directly into STAPLE. The simpler data format was included to avoid the unnecessary complexity of POST formatting for atmospheric profiles generated from MarsGRAM or other engineering-level atmospheric modeling tools. STAPLE includes several default atmospheric profiles including the 1962 US Standard Atmosphere and the Kass atmosphere utilized for Mars Exploration Rovers mission planning.

Local atmospheric temperature is of little interest for the purposes of STAPLE and thus is not modeled within the tool. Instead, to determine the local Mach number of the entry vehicle, a reference speed of sound (a_0) is included as an input. It turns out that atmospheres tend to behave in such a way that sound speed does not vary too much with altitude, so approximating the local Mach number from a_0 gives a good approximate answer. This is demonstrated graphically in Figure I, which shows the significance of this assumption on Earth relative to the 1962 standard

daytime atmosphere. As you can see the error is less than ~7% for altitudes less than 65000m. If higher accuracy is desired than sound speed can be input in tabular form as a function of altitude.

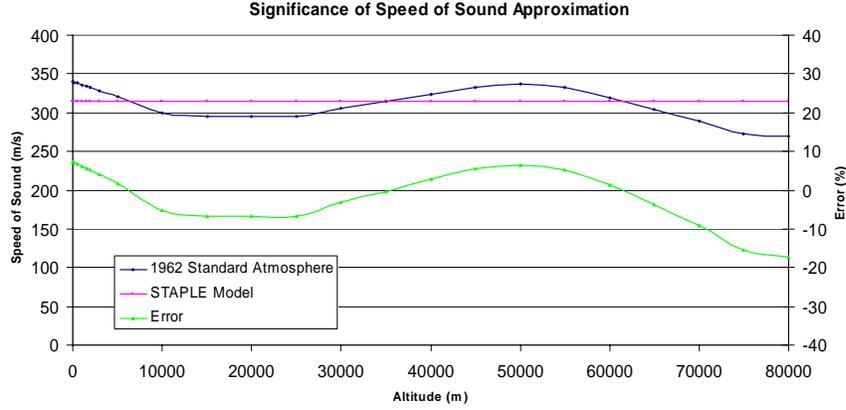


Figure I: Significance of Constant Sound Speed Approximation

In summary, a complete planetary model can be specified by only five key variables; r_0 , μ , ρ_0 , H , and a_0 . If an exponential atmosphere model is undesirable then a tabular file can be input instead. For many entry problems planetary rotation rate (ω) can impact the trajectory significantly, and is thus included as an additional planetary variable. These six constants, from Vallado^[8], are stored within the model for Mars, Earth, Venus, and Titan and can also be specified for an arbitrary target body.

Table I: STAPLE Planetary Variables

	<i>Earth</i>	<i>Mars</i>	<i>Venus</i>	<i>Titan</i>
Surface Density (kg/m ³)	1.2260066	0.0068	16.02	4.904
Atmospheric Scale Height (m)	7257	17391.3	6226.65	140000
Planetary Radius (m)	6378136	3397200	6052000	2720332
Reference Sound Speed (m/s)	315	222	529.6	194
Gravitational Parameter (m ³ /s ²)	3.986*10 ¹⁴	4.305*10 ¹³	3.257*10 ¹⁴	1.006*10 ¹³
Planetary Rotation Rate (rad/s)	7.29*10 ⁻⁵	7.08*10 ⁻⁵	-2.99*10 ⁻⁷	4.56*10 ⁻⁶

B. Trajectory Propagation

STAPLE propagates the 3-DOF planetary entry trajectory by integrating the applicable equations of motion. Assuming a constant mass, non-thrusting vehicle, the 3-DOF point mass dynamics of the entry vehicle over a spherical rotating planet are governed by the following equations of motion^[3,4]:

$$\dot{\theta} = \frac{\cos(\gamma)\cos(\phi) * V}{r \cos(\phi)} \quad (3)$$

$$\dot{\phi} = \frac{\cos(\gamma)\sin(\phi) * V}{r} \quad (4)$$

$$\dot{r} = -\sin(\gamma) * V \quad (5)$$

$$\dot{\psi} = -\frac{\cos(\gamma)\cos(\phi)\tan(\phi)}{r} * V + \frac{L\cos(\gamma)\sin(\sigma)}{V} + \dot{\psi}_\omega \quad (6)$$

$$\dot{\gamma} = \left(g - \frac{V^2}{r} \right) * \frac{\cos(\gamma)}{V} - \frac{L \cos(\sigma)}{V} + \dot{\gamma}_\omega \quad (7)$$

$$\dot{V} = -D + g \sin(\gamma) + \dot{V}_\omega \quad (8)$$

The first three equations are the kinematic equations which yield time derivatives of longitude (θ), latitude (ϕ), and radial distance to the planet center (r) respectively. The latter three equations are the force equations which yield time derivatives of azimuth (ψ), flight path angle (γ), and velocity (V) respectively. Azimuth is measured clockwise from due North and flight path angle is measured positive downward from the local horizontal. Other terms in equations (3)-(8) are lift (L) and drag (D) specific forces, gravity (g), and the vehicle bank angle (σ), which is defined such that the horizontal component of lift is zero at zero bank angle.

Terms denoted with an ω subscript are effects due to planetary rotation. In general the self rotation rate of a planet is small and these effects are negligible. However, for long-range high-speed flight (e.g. shuttle entry) or large radius bodies (e.g. Jupiter) the effect becomes significant. To maintain the generality of STAPLE these terms are included as specified in equations (9)-(11)^[3]:

$$\dot{\psi}_\omega = -2\omega(\tan \gamma \cos \phi \sin \varphi + \sin \phi) - \frac{\omega^2 r}{\cos \gamma} \sin \phi \cos \phi \cos \varphi \quad (9)$$

$$\dot{\gamma}_\omega = 2\omega \cos \phi \cos \psi + \frac{\omega^2 r}{V} \cos \phi (\cos \gamma \cos \phi - \sin \gamma \sin \phi \sin \varphi) \quad (10)$$

$$\dot{V}_\omega = -\omega^2 r \cos \phi (\sin \gamma \cos \phi + \cos \gamma \sin \phi \sin \varphi) \quad (11)$$

C. Event Modeling

STAPLE is capable of modeling several different event types which may be encountered during a typical planetary entry. These include parachute deployment, parachute release, vehicle staging, and descent propulsion events. The events can occur separately or overlap (e.g. heatshield separation while on parachute). Each of these event types is triggered by any of several variables including: dynamic pressure, Mach number, altitude, density, deceleration, velocity, and multiple timer options. An event begins when the designated variable passes through a preset ‘trigger value’ with either increasing or decreasing slope according to the chosen input option. The event types are covered in detail below in regards to functionality and application.

Vehicle Staging

Staging events during planetary entry will typically include heatshield and/or backshell separation and are often triggered by Mach number or by a timer which initializes after main chute deployment. Staging events simply decrement the total vehicle mass by a specified amount. Current modeling does not allow for changes to the vehicle aerodynamics after staging because it is assumed that vehicle aerodynamics are dominated by the vehicle forebody (so staging from the aftbody will have little aerodynamic effect) and that forebody staging events (e.g. heatshield) occur late in the entry trajectory where parachute aerodynamics are dominant.

Parachute Deployment and Release

Several assumptions go into the modeling of a parachute deployment event within STAPLE. The most significant of these is the assumption that parachute aerodynamics dominate overall vehicle dynamics to the point that the aerodynamic forces on the vehicle itself become negligible after chute inflation occurs. It is assumed that this transition occurs instantly and no bridging function is included. In addition it is assumed that the parachute drag coefficient is constant with Mach number for the regime in which it is utilized, and hence the aerodynamic properties of the parachute are specified by a single drag coefficient. If multiple chutes are deployed then the one deployed most recently becomes dominant.

Simulating a parachute deployment event requires specification of several variables. As with all events, a trigger variable, slope, and value must be specified at which the event is to initialize. For a parachute this trigger is typically a value of dynamic pressure or Mach number with negative slope. Parachute parameters must also be specified including inflated diameter, inflation rate, and drag coefficient. The inflation rate is specified to avoid an infinite g-spike upon chute inflation and the rate can be set to control the magnitude of the g-spike that occurs.

Parachute release is typically triggered by a target altitude and/or velocity. Upon parachute release the vehicle aerodynamics are restored. The mass decrement resulting from parachute release is not modeled, but, if desired, could be by including an additional staging event in the simulation. Parachute release typically will occur immediately prior to the propulsive descent phase, if it all, but there are applications (e.g. ballute release after Aerocapture) where this is not the case.

Propulsive Descent

The propulsive descent event included in STAPLE models a simple gravity turn. Specification of available thrust, target altitude, and target velocity are required as inputs, along with a history of the simulated trajectory. The gravity turn analysis determines when to initiate thruster firing, which is always anti-parallel to the velocity vector, to achieve the target conditions. A solution is obtained by assuming a constant mass system dominated by gravitational and thrust accelerations, with negligible aerodynamic forces. Required acceleration from thrust at given initial conditions is found by solving:

$$4g(h_i - h_f) \left(\frac{a_T}{g} \right)^2 - 2(V_i^2 - V_f^2) \cos \phi_i \left(\frac{a_T}{g} \right) - ((V_i^2 - V_f^2) + (V_i^2 - V_f^2) \cos^2 \phi_i + 4g(h_i - h_f)) = 0 \quad (12)$$

An iterative solution is achieved by solving for required thrust at each time step in the trajectory and determining where available thrust meets the thrust requirement. It is at this point that the gravity turn is initiated. The gravity turn maneuver is then simulated by integrating the relevant equations of motion, under the assumption that gravity and thrust forces dominate and aerodynamic forces and rotational effects can be neglected. Trajectory propagation does not assume constant mass, as was assumed in equation (12), but assumes constant thrust so that the acceleration due to thrust increases during the gravity turn. For most applications this is of little consequence but for missions that use significant propellant mass fractions during the gravity turn, the targeted conditions will not be achieved. The kinematic equations of motion for this phase are the same as in (1)-(3) while the force equations, (13)-(15), have changed with our assumptions. Mass flow (16) is modeled according to the rocket equation based on an entered specific impulse.

$$\dot{\psi} = -\frac{\cos(\gamma) \cos(\phi) \tan(\phi) * V}{r} \quad (13)$$

$$\dot{\gamma} = \left(g - \frac{V^2}{r} \right) * \frac{\cos(\gamma)}{V} \quad (14)$$

$$\dot{V} = -\frac{T}{m} + g \sin(\gamma) \quad (15)$$

$$\dot{m} = m \cdot \left(1 - \exp\left(\frac{-\dot{V}}{I_{sp} \cdot g_c} \right) \right) \quad (16)$$

D. Initial Conditions

STAPLE simulations begin at the input initial conditions in a planet-centered reference frame. The initial vehicle state is specified by six variables: relative velocity, altitude, longitude, latitude, azimuth, and flight path angle. Initial

vehicle parameters must also be specified which include, at minimum, mass, reference area, nose radius, emissivity, and aerodynamic coefficients. Aerodynamic coefficients can be input as constant or in tabular form as monovariant functions of either Mach or angle of attack. More detailed vehicle parameters must be specified for any events included in the trajectory, as described above. A sample input file is included in Appendix A.

E. Auxillary Calculations

STAPLE numerically integrates the equations of motion given in equations (3)-(8) using a specified time step to yield longitude, latitude, radial position, azimuth, flight path angle, energy, and velocity as functions of time. This integration requires the calculation of gravity, atmospheric density, lift, and drag at each time step. Calculation of gravity and density are from equations (1) and (2), specific lift and specific drag are calculated using:

$$L = \frac{1}{2} \rho V^2 \cdot \left(\frac{A}{m} \right) \cdot C_L \quad (17)$$

$$D = \frac{1}{2} \rho V^2 \cdot \left(\frac{A}{m} \right) \cdot C_D \quad (18)$$

In addition to these required variables, several auxiliary variables of interest are calculated at each time step. Landing site relative altitude is calculated based upon an input landing site altitude. Mach number is determined from the local sound speed. Dynamic pressure, vehicle ballistic coefficient and acceleration are determined from the standard relations. Projected range over the spherical planet is calculated according to:

$$Range = r_0 \cos^{-1}(\sin(\phi_0)\sin(\phi) + \cos(\phi_0)\cos(\phi)\cos(\theta_0 - \theta)) \quad (19)$$

Finally the stagnation point convective heat rate is determined according to the Sutton-Graves equation and is integrated to track the heat load. Equilibrium wall temperature is calculated from the Stephan-Boltzmann law.

F. Guidance

STAPLE is capable of simulating guided entry and aerocapture guidance through bank angle modulation. The guidance algorithm implemented in STAPLE is based on early development work of the Hybrid Predictor-corrector Aerocapture Scheme (HYPAS) for application to the Aeroassist Flight Experiment (AFE)^[15]. HYPAS algorithm derivation is outlined in reference 14 and was implemented in STAPLE according to the same scheme. The algorithm consists of two phases; an entry phase that guides the vehicle toward an equilibrium glide condition, and an exit phase that guides the vehicle towards a target apogee altitude.

Entry phase guidance commands bank angle according to the control law in equation 20. The first term specifies the bank angle required to achieve an equilibrium glide condition. The second term is useful in damping rapid altitude changes and the third term biases control towards a reference dynamic pressure, set to maintain a control margin by using only a fraction of the maximum available lift. The G_h and G_q terms are gains specified to control the dynamic response of the system. These gains are constant throughout the entry phase of the guidance algorithm.

$$\cos \sigma_{entry} = \frac{m}{C_L \cdot q \cdot A} \left(g - \frac{V^2}{r} \right) - G_h \left(\frac{\dot{h}}{q} \right) + G_q \left(\frac{q - q_{ref}}{q} \right) \quad (20)$$

$$\cos \sigma_{exit} = \frac{m}{C_L \cdot q \cdot A} \left(g - \frac{V^2}{r} \right) - G_h \left(\frac{\dot{h} - \dot{h}_{ref}}{q} \right) \quad (21)$$

Exit phase guidance commands bank angle according to the control law in equation 21. Gain, G_h , for this phase is set independently of the gains in the entry phase guidance formulation. The equation has a similar form to the entry phase control law but instead of biasing towards a target dynamic pressure the commanded bank angle is biased towards a reference altitude rate. This reference term is updated at each point in the trajectory by predicting the constant altitude rate necessary to reach the target apogee from the current vehicle state. Calculation of the

reference altitude rate is done by assuming an exponential atmosphere to predict exit conditions and utilizing orbital mechanics to calculate apogee altitude from the predicted exit state. In order to compensate for variations between the exponential model and the actual density experienced in flight, the predictor's atmosphere model is constantly updated during flight. This is achieved by calculating the local value of $\rho \cdot C_D$ from measured drag acceleration, which is then divided by the expected value of $\rho \cdot C_D$ to produce a density multiplier. This multiplier is updated at each time step and run through a low pass filter to reduce noise. Updating in this fashion allows for real-time adaptation to uncertainties encountered during flight.

III. Validation

Validation of the Simulated Trajectory Analysis code for Planetary Entry was accomplished by comparing results against actual flight experience and comparable POST simulations for two cases. Validation of ballistic entry was performed against Mars Pathfinder and validation of a guided lifting aerocapture trajectory was performed (for lack of actual flight experience) against AFE studies.

A. Ballistic Entry

The Mars Pathfinder (MPF) flight experience was chosen as the benchmark for validation of STAPLE's ballistic entry simulation capabilities. Ballistic entry at Mars is one of the primary applications for which STAPLE was developed. This fact, and the available wealth of published data, makes MPF a natural choice for initial tool validation. MPF simulations were performed under the same assumptions in both STAPLE and POST. Entry conditions, vehicle parameters, and event modeling were gathered from published data^[9,10,11,12]. Aerodynamic coefficients were modeled using Newtonian impact method calculations and are assumed constant until parachute deployment. For numerical values see the input file included in Appendix A. Results from the STAPLE simulation, shown in Figure II, compare extremely well with POST simulation results. Graphical comparison, shown in Figure III, shows the plots lie almost on top of each other with minor deviations in the magnitudes of peak heat rate and peak deceleration. Looking at the acceleration and flight path angle plots we see a more significant deviation occurring just after parachute deployment occurs ($\sim t=144$ s). This deviation is due to different parachute deployment models: STAPLE assumes that the change in area with time is constant, while the POST model assumes that the change in diameter with time is constant. The effect of this difference on the simulation is small, but worthy of note.

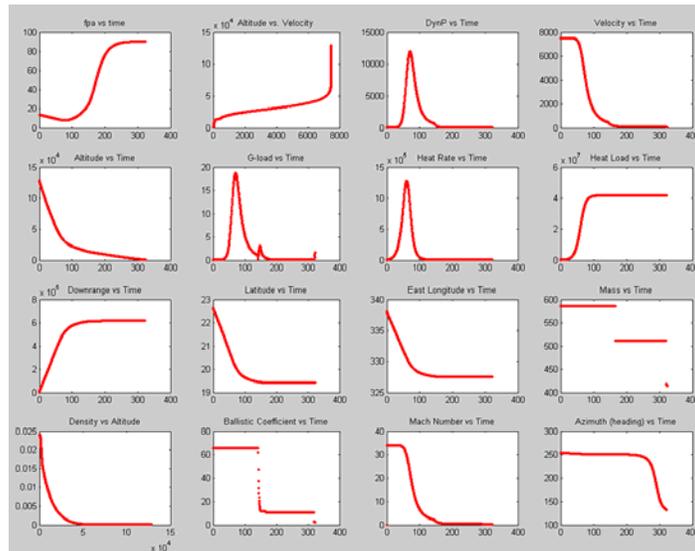


Figure II: Results of MPF Simulation in STAPLE

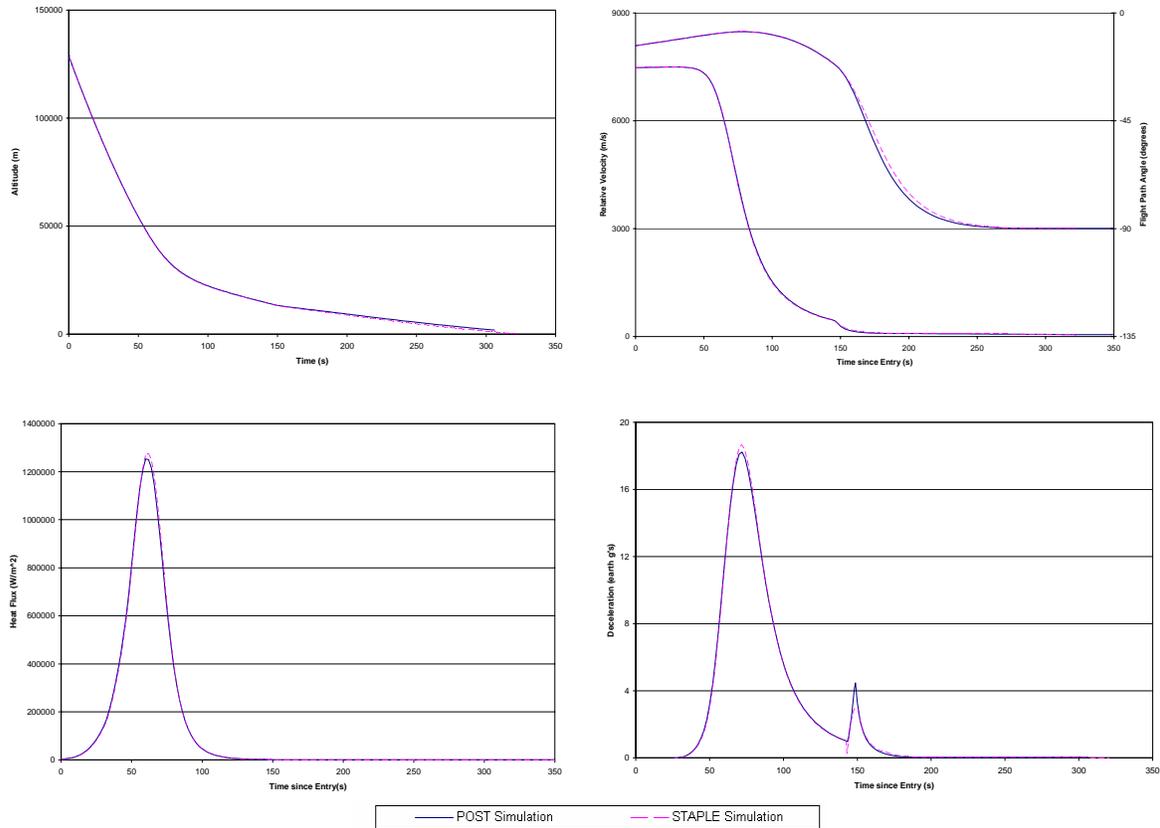


Figure III: Comparison of STAPLE and POST Simulations

We can better compare the two simulations by looking at percent differences. Figure IV plots the deviation of STAPLE compared to POST for several key variables. Due to the difference in parachute deployment modeling discussed above, comparison is made only up until parachute deployment occurs.

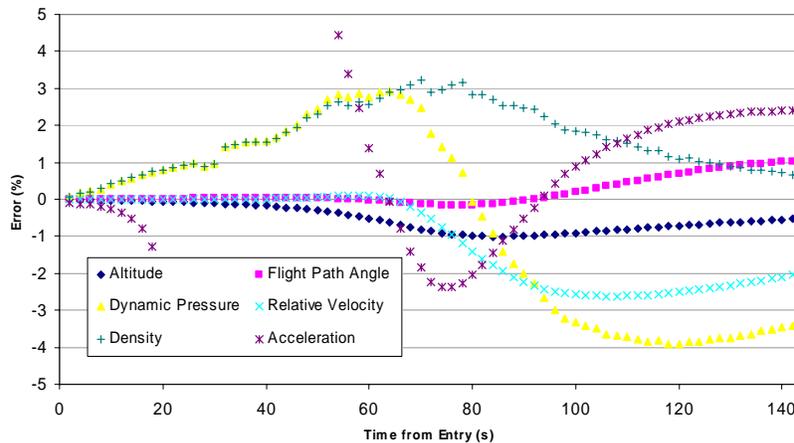


Figure IV: Percent Deviation between STAPLE and POST Simulations

With the exception of a brief spike to ~10% error in the acceleration term at peak deceleration, all of the simulated variables remain within $\pm 5\%$ of the POST simulation results. All variables begin with negligible error indicating that the initial conditions are virtually identical. Almost immediately the simulated density deviates slightly and this effect directly leads to the deviation of dynamic pressure and acceleration. The propagation of these

effects leads to deviation of the other variables. The two simulations use the same atmospheric input tables which list density as a function of altitude, meaning that the density variation arises from a variation in altitude vs. time or from numerical differences in interpolating the table values. The latter of these hypotheses was ruled out after investigation. The former hypothesis is likely since at high altitudes a very small change in altitude can lead to a large percent change in density. In fact, at the initial conditions a 0.1% change in altitude causes a 1.8% change in density. It is believed that the difference in altitude vs. time is linked to numerical differences in integration. Despite this notable difference, STAPLE results compare quite favorably to the POST simulation.

It is of further interest to compare simulation results against actual flight data. The initial comparison was poor due to differences between the flight atmosphere and the engineering model as shown in Figure V. This difference was resolved by using atmospheric data reconstructed from MPF flight data for the STAPLE simulation^[10]. Since this data was not available in tabular form it was reconstructed from available graphs, which resulted in a low-resolution reconstruction, likely with some inaccuracies.

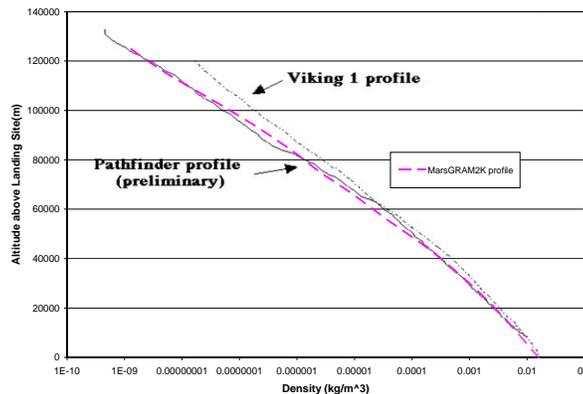


Figure V: Comparison of Flight Atmosphere with Atmospheric Model

Simulation results using this corrected atmosphere are shown in Figure VI against MPF flight data^[9,10,11,12]. The Figure illustrates that the simulation performs very well for the first ~75 seconds of flight, and then begins to deviate from the actual flight data. This deviation is expected because it is exceedingly difficult to model all of the dynamics that are introduced in the flight environment. The approximations of constant drag coefficient and zero angle of attack, for example, are known to be inaccurate. In fact the actual angle of attack spikes to ~5 degrees 60 seconds after entry^[10]. Nonetheless, comparison against POST simulation and MPF flight data validates the simulation capability of STAPLE for ballistic entry on Mars.

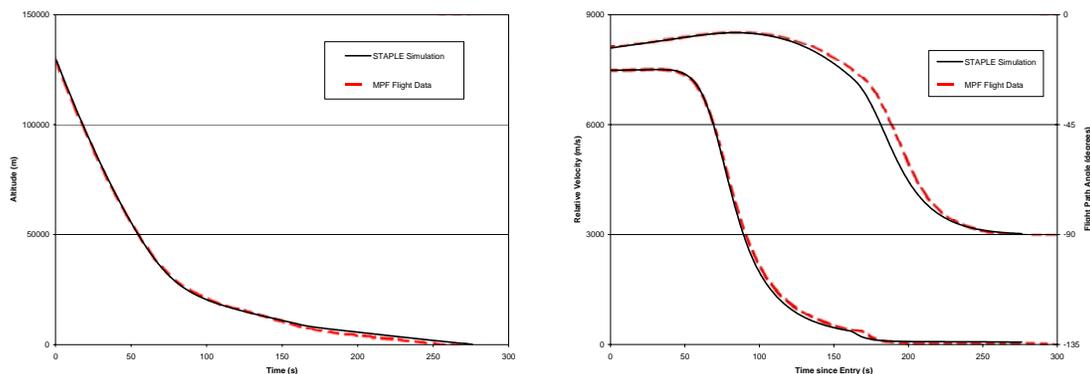


Figure VI: Comparison of STAPLE Results to MPF Flight Experience

A final note of interest, shown below, is the validation of ground track generation with STAPLE. The simulated landing site is within $\pm 0.25^\circ$ longitude and $\pm 0.2^\circ$ latitude of the actual landing site, which translates to a 15km error that is well within the 300km x 50km landing footprint for the mission.

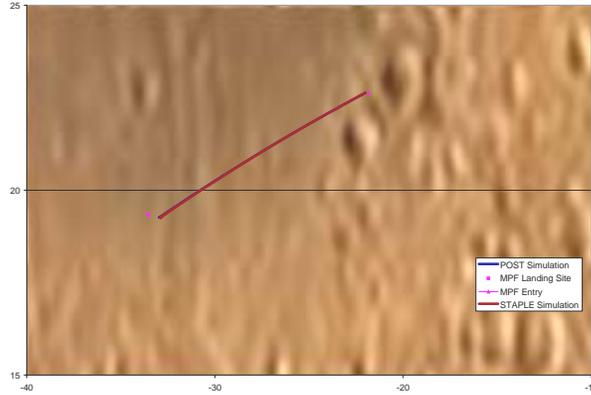


Figure VII: Actual and Simulated MPF Ground Tracks

B. Aerocapture

Validation of the HYPAS-based algorithm implemented in STAPLE was performed against simulated Earth aerocapture results from the Aeroassist Flight Experiment (AFE) project^[13]. The study used for comparison employed an unspecified 3-DOF trajectory code to perform their analysis. The problem specifies entry conditions at atmospheric interface, phase change velocity, and target apogee parameters. The original problem also includes a target final inclination, thus placing limits on the sign of the bank angle command to maintain inclination within a deadband limit. The guidance algorithm implemented in STAPLE does not include this functionality and, as seen in the figure, simply assigns the sign of the bank angle to positive at all times. The problem specifies a target apogee of 200 nautical miles (370,000 m) which is achieved within 1% by both the original algorithm and the STAPLE implementation.

The reference study utilized the 1962 US standard atmosphere model for its analysis, but this model produced poor STAPLE simulation results. The problem pertains to the logic that updates STAPLE's internal density model based on measured drag acceleration, and is currently unresolved. As a workaround solution, the STAPLE simulation utilizes an exponential model for atmospheric density. Essentially this means that the encountered atmosphere is identical to the one modeled within the guidance algorithm, and updates to the internal model are no longer necessary. While using a different atmospheric model from the reference data limits the benefit of the validation exercise, the comparison is still worthwhile.

STAPLE simulation was performed using a nominal exponential Earth atmosphere. It was found that the gains specified in the reference material did not work adequately for STAPLE simulation, perhaps due to english/metric conversion or due to the varied atmosphere model. Instead the gains were set at $G_{h1}=10$, $G_q=0.5$, and $G_q=4$. Comparison against original study results proved difficult as data availability was only in the form of distorted viewgraphs. Nonetheless a comparison can be made by superimposing STAPLE simulation results over these images as is done in Figure VIII.

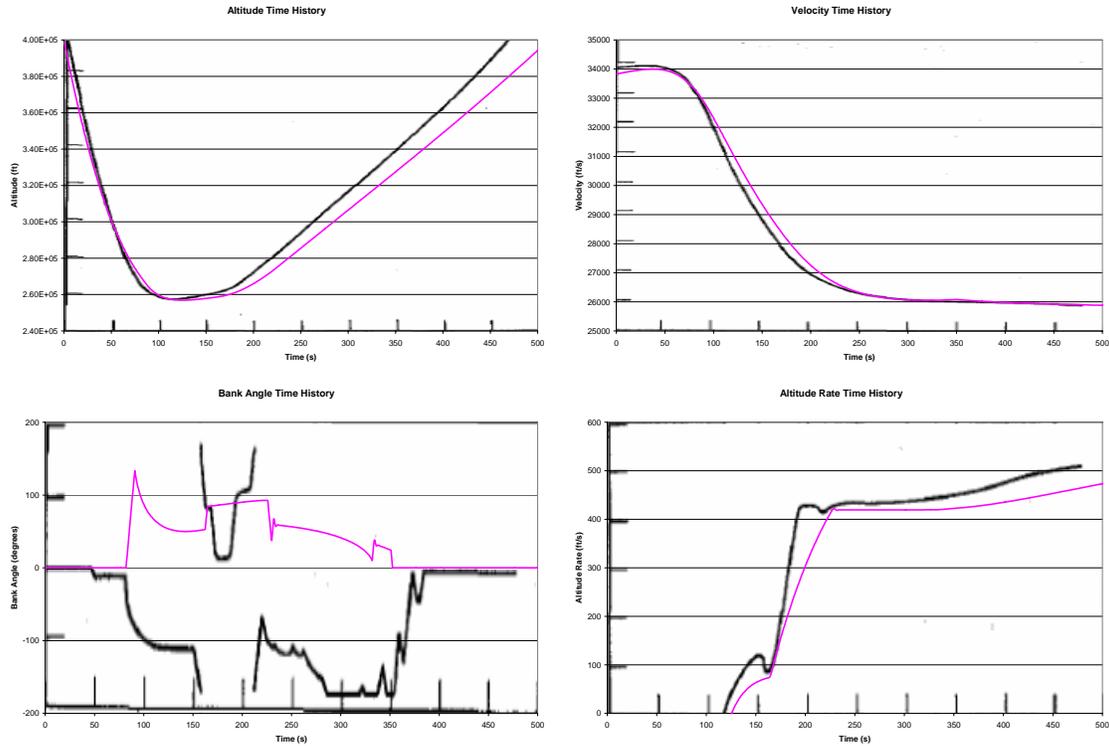


Figure VIII: STAPLE Comparison to AFE Analysis

While distortion of the background images makes exact comparison difficult, initial visual comparison shows significantly dissimilar results with many trend similarities. Starting with the altitude history plot, it can be seen that the STAPLE simulation descends to approximately the same minimum altitude as the reference results, and begins ascending ~ 20 s later at approximately the same slope. This same observation can be made from the altitude rate history plot. The velocity history plot appears to show a discrepancy in initial velocity, but inspection shows that this is due to distortion of the background image. The simulated velocity takes slightly more time to decrease than the reference values and then, from ~ 250 s to EOS, appears right on top of the reference data. These deviations are all consistent and are explained by the varied atmosphere model. Comparison between bank angle schedules is difficult because of the different gains that were needed, but the plot is shown for completeness. Recall that STAPLE does not implement bank reversals so comparison should be made between the absolute values of bank angle. And finally, shown in Figure IX, is a comparison of the reference altitude rates. These values represent the constant altitude rate, from the current spacecraft state, that will achieve the target apogee condition based on current predictions. Again, since we are comparing simulations utilizing different atmosphere models, it is difficult to glean much from the comparison. Nonetheless, the behavioral trends and proximity in magnitude indicate that the STAPLE guidance algorithm is performing correctly.

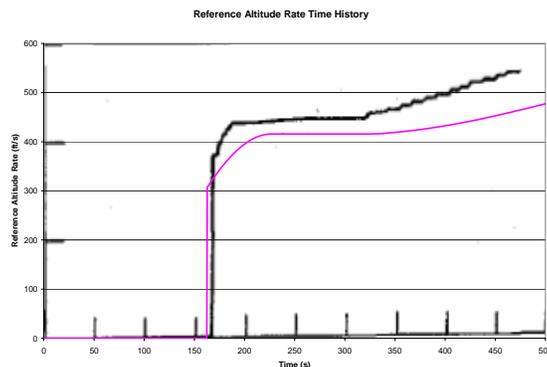


Figure IX: STAPLE Comparison to AFE Analysis

IV. User Interface

STAPLE has two operational I/O modes: a script I/O mode for experienced users and a graphical user interface (GUI) mode for inexperienced users. Outputs are shown graphically and time histories of each variable are stored in a comma delimited text file for easy import into excel.

A. GUI Input

The GUI input mode is designed to allow inexperienced users to analyze simple entry trajectories. As such, the utility of STAPLE is limited when run in this mode. Bank angle modulation is not allowed and trajectories must be flown at constant angle of attack.

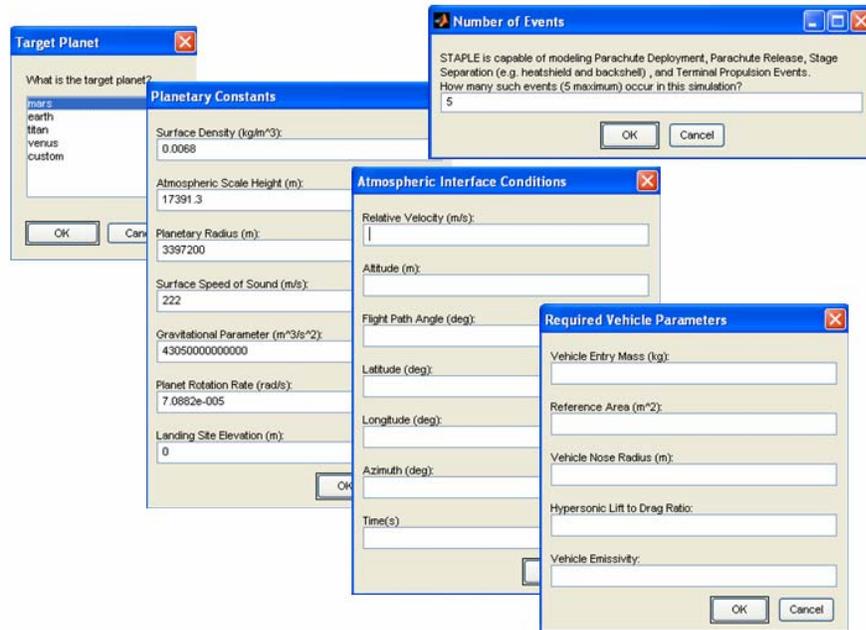


Figure X: STAPLE GUI Input Windows

GUI input walks users through the required inputs as shown in Figures X and XI, generally flowing from left to right in the Figures. Users begin by selecting the target body, and then selecting key planetary constants. These constants are populated automatically for the chosen body, but can be modified if desired or if a 'custom' planet is chosen. Next the user defines the vehicle state at atmospheric interface; including velocity, altitude, flight path angle, latitude, longitude, azimuth and time. The following prompt accepts inputs relating to vehicle parameters; including mass, area, nose radius, lift-to-drag ratio, and emissivity. Finally the user has to specify trajectory events that are to be modeled, beginning with the number of events - which the GUI input mode limits to five. The user is then prompted, based on the number of events, to select event types and input key parameters associated with the events. Event types include parachute deployment, parachute release, separation, and terminal propulsion. Each event type has a specific GUI depending on the input requirements for that event type. These inputs are discussed in the event modeling section of this report.

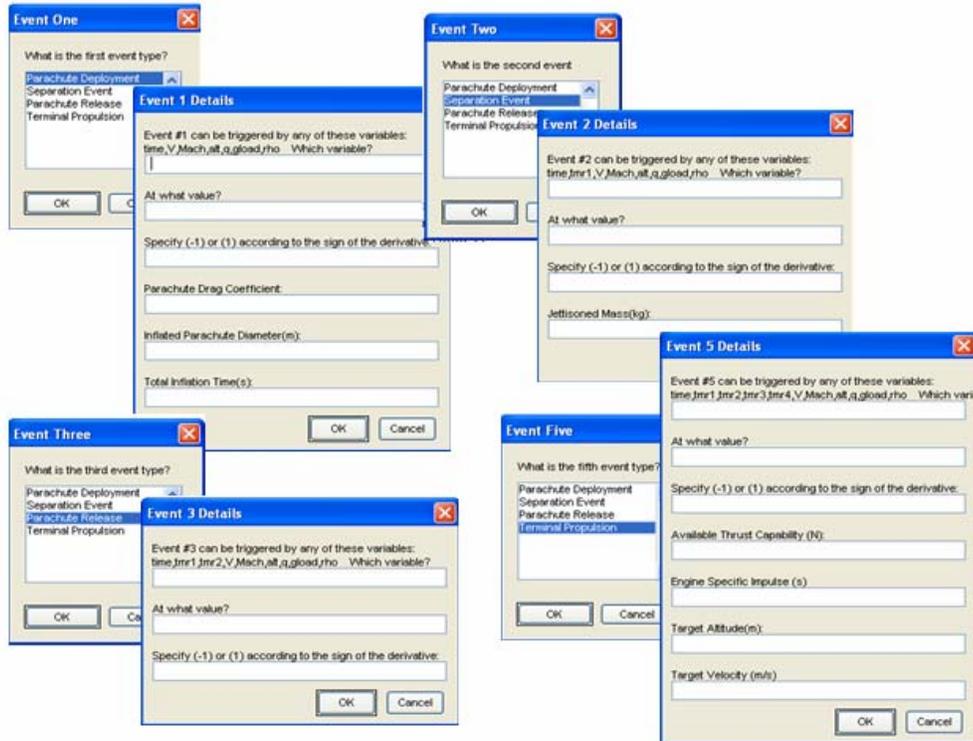


Figure XI: STAPLE GUI Event Input Windows

Finally, the GUI allows users to specify atmospheric input files and aerodynamic coefficient tables if desired. Otherwise the user can select an exponential model based on the reference density and scale height that have already been specified. Likewise, the user can choose to omit coefficient tables and use constant values instead.

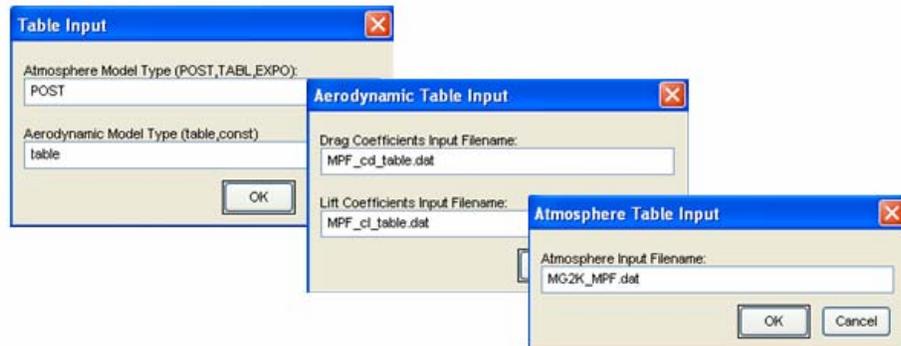


Figure XII: STAPLE GUI Table Input Windows

Clearly there are many required inputs and it takes some time to enter all the data correctly. For this reason an input script is automatically generated while the GUI is run that saves the input values. If slight modifications to the problem are desired, then changes can be made to this script file and STAPLE can be run automatically from the script.

B. Script Input

A script input mode is also included for experienced users. This input mode allows greater flexibility and enables the full functionality of STAPLE. The script simply stores key input parameters in a compact m-file, and is automatically read into STAPLE. Examples of script input files for MPF and AFE simulations are included in Appendices A and B.

C. Output

STAPLE automatically generates a comma delimited output file that stores values for all internal variables at each time step. This format allows for easy import into Excel to compare against data from other sources. Additionally, STAPLE automates trajectory visualization using the built-in plotting capabilities of Matlab. Default visualization is the standard 16 plot set shown in Figure II and on the right side of Figure XIII.

V. Sample Applications

Prior to availability of an official version of STAPLE, the code was already being utilized for several applications. It has been used for analysis in support of the RASC Mars Human Precursor project led by Dr. Joel Levine at NASA LaRC to provide an estimate of the upper limit on landed mass for specific mars entry problems, to investigate the sensitivity of this limit to entry flight path angle, and to provide conceptual analysis of a mars airplane mission entry. Sample application results are summarized in Figures XIII and XIV. Additionally, working versions of the code have been distributed to, and utilized by, at least two teams participating in a Mars Scout proposal competition as part of the 2005 senior spacecraft design course at Georgia Tech.

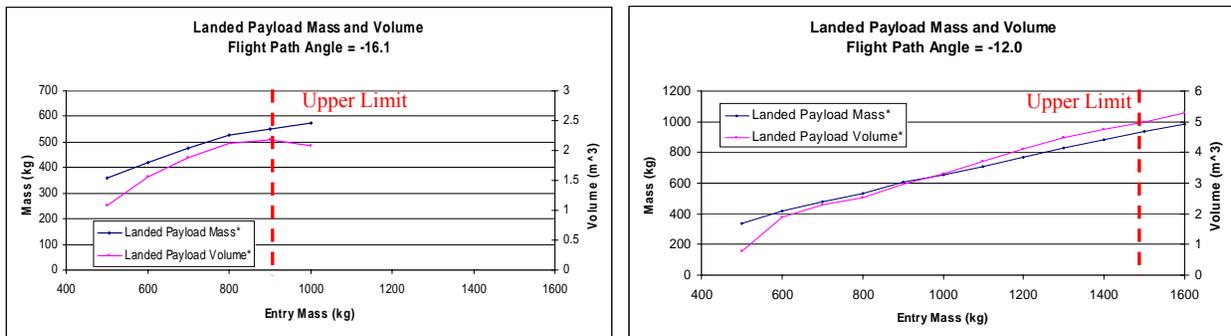


Figure XIII: Sample Application Results: Landed Mass Limits and Sensitivity to Entry FPA for Given Entry Conditions

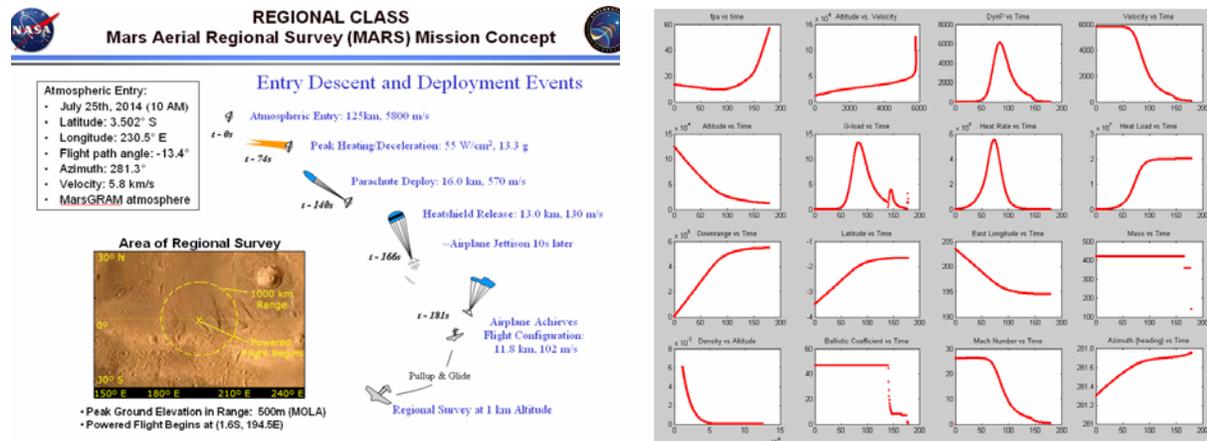


Figure XIV: Sample Application Results: Conceptual Entry Analysis for Mars Airplane

VI. Conclusion

A 3-DOF Simulated Trajectory Analysis code for Planetary Entry has been developed to perform planetary trajectory analysis within the Planetary Entry Systems Synthesis Tool. Capabilities include ballistic and lifting entry simulations as well as aerocapture simulations at arbitrary planetary bodies. Comparable performance to the NASA Program to Optimize Simulated Trajectories has been demonstrated for a ballistic entry and descent simulation at Mars including parachute inflation, heatshield separation, parachute release, and backshell separation events. Additionally, favorable comparison to Mars Pathfinder flight experience has demonstrated tool applicability for conceptual level design of similar missions. Finally, a modest bank angle modulated guidance capability has been demonstrated that produces comparable results to Aeroassist Flight Experiment studies for Earth aerocapture applications. STAPLE utility as a standalone tool has been demonstrated and, utilizing the standard Matlab plug-in, the tool can be readily integrated within the ModelCenter design environment for PESST integration. Additionally, for future web-based integration efforts, STAPLE can be converted to a standalone executable file utilizing the native Matlab-to-C compiler (MCC).

Appendix A: MPF Input File

```

%%Atmospheric Interface Conditions
%%
Vatm=7478.44161;      % RELATIVE velocity (m/s)
hatm=125000;          % Altitude relative to spehrical Planet(m)
fpa=deg2rad(13.650);  % RELATIVE Flight Path angle(degrees), down is positive
lat=deg2rad(22.6303); % Planetocentric Latitude
long=deg2rad(337.998); % Planetocentric Longitude
azimuth=deg2rad(253.674254); % RELATIVE azimuth or heading angle (Due East is 90 degrees)
time=0;               % Start time in seconds
landsiteelevation = -3682; % Landing Site Elevation relative to reference sphere(m)
planet='mars';        % input first four letters of planet name in lowercase string

%%Planetary Constants
%%
%%For Mars%%
if (strncmp(planet,'mars',4))
    %go=3.718;          % (m/s^2)
    rho_o=0.0068;      % (kg/m^3)
    H=17391.3;         % (m)
    Ro=3397200;        % (m)
    ao =222;           % Reference Speed(m/s) of sound near surface
    mu = 4.305*10^13;  % Gravitational Parameter (m^3/s^2)
    omega = 7.0882*10^-5; %Planetary Rotation Rate (rad/s)
%%For Earth%%
elseif (strncmp(planet,'earth',5))
    %go=9.81;          % (m/s^2)
    rho_o=1.2260066;   % (kg/m^3)
    H=7257;            % (m)
    Ro=6378136;        % (m)
    ao = 331;          % Reference Speed(m/s) of sound near surface
    mu = 3.986004415*10^14; % Gravitational Parameter (m^3/s^2)
    omega = 7.292*10^-5; %Planetary Rotation Rate (rad/s)
elseif (strncmp(planet,'venus',5))
    %go=8.87;          % (m/s^2)
    rho_o=16.02;        % (kg/m^3)
    H=6226.65;         % (m)
    Ro=6052000;        % (m)
    ao = 529.6 ;       % Reference Speed(m/s) of sound near surface
    mu = 3.257*10^14;  % Gravitational Parameter (m^3/s^2)
    omega = -2.9925539*10^-7; %Planetary Rotation Rate (rad/s)
elseif (strncmp(planet,'titan',5))
    %go=1.36;          % (m/s^2)
    rho_o=4.904;        % (kg/m^3)
    H=140000;          % (m)
    Ro=2720332;        % (m)
    ao = 194 ;         % Reference Speed(m/s) of sound near surface
    mu = 1.006428*10^13; % Gravitational Parameter (m^3/s^2)
    omega = 4.560686*10^-6; %Planetary Rotation Rate (rad/s)
end
go=mu/Ro^2;

%%Vehicle Parameters
%%
m=585;                % Vehicle Entry Mass(kg)
m_heatshield=73.9;    % Heatshield Mass(kg) separated 20s after parachute deployment
m_backshell=94;       % Backshell Mass(kg) mass of backshell, chutes separated at chute cut-off
A=pi()*((2.65/2)^2);  %Reference Area(m^2)
chutediameter=12.5;   % Inflated Chute Diameter (m)
nose_radius = 0.6638; % nose radius of the ballistic entry vehicle (m)
Cd=1.63;              % Hypersonic Drag Coefficient (dimensionless)
chuteCd=0.41;         % Drag Coefficient of the Parachute (dimensionless)
L_over_D=0;           % Hypersonic in-plane L/D (dimensionless)
BCo=m/(Cd*A);         % Ballistic Coefficient at entry
emissivity=0.8;
initial_bank_angle=0; % Initial Bank Angle

```

```

alpha_o=0;          % Initial Angle of Attack

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Event Parameters (maximum of 5 events)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Event type
% 0 - No event
% 1 - Parachute Deploy (Cd, Diameter, inflation_time)
% 2 - Separation Event (change in mass)
% 3 - Parachute Release
% 4 - TERMINATE SIMULATION or begin Terminal Propulsion (thrust (N), isp (s), target_alt (m),target_V (m/s))
event_types = [1,2,3,2,4]; %specify events chronologically, use 0's to fill event_type to length of 5
trigger_variables = {'q','Mach','alt','tmr3','alt'};
trigger_values = [583, 0.6, 300, 1, 0];
trigger_slopes = [-1, -1, -1, 1, -1]; %specify if the variable is increasing or decreasing
event1_specs = [0.41, 12.5, 8];
event2_specs = [73.9];
event3_specs = [];
event4_specs = [94];
event5_specs = [3000,320,100,0]; %[0] if there is no terminal propulsion phase
guidance=0;

%Density Model
atm_type = 'POST';
atm_filename='MPF_recon2.dat';
%Speed of Sound Model
sos_type = 'const';
sos_filename='sos_test.dat';
%Aerodynamic Models
aero_type = 'const';
CD_filename = 'MPF_cd_table.dat';
CL_filename = 'MPF_cl_table.dat';

```

Appendix B: AFE Input File

```
%%Atmospheric Interface Conditions
%%
Vatm=10311;          % RELATIVE velocity (m/s)
%Vatm=9838;
hatm=121920;        % Altitude relative to spehrcal Planet(m)
fpa=deg2rad(4.5);   % RELATIVE Flight Path angle(degrees), down is positive
%fpa=deg2rad(4.716);
lat=deg2rad(0);     % Planetocentric Latitude
long=deg2rad(0);    % Planetocentric Longitude
azimuth=deg2rad(90); % RELATIVE azimuth or heading angle (Due East is 90 degrees)
time=0;             % Start time in seconds
landsiteelevation = 0; % Landing Site Elevation relative to reference sphere(m)
planet='earth';    % input first four letters of planet name in lowercase string

%%Planetary Constants
%%
%%For Mars%%
if (strncmp(planet,'mars',4))
    %go=3.718;        % (m/s^2)
    rho_o=0.0068;    % (kg/m^3)
    H=17391.3;       % (m)
    Ro=3397200;      % (m)
    ao =222;         % Reference Speed(m/s) of sound near surface
    mu = 4.305*10^13; % Gravitational Parameter (m^3/s^2)
    omega = 7.0882*10^-5; %Planetary Rotation Rate (rad/s)
%%For Earth%%
elseif (strncmp(planet,'earth',5))
    %go=9.81;        % (m/s^2)
    rho_o=1.2260066; % (kg/m^3)
    H=7257;          % (m)
    Ro=6378136;      % (m)
    ao = 331;        % Reference Speed(m/s) of sound near surface
    mu = 3.986004415*10^14; % Gravitational Parameter (m^3/s^2)
    omega = 7.292*10^-5; %Planetary Rotation Rate (rad/s)
elseif (strncmp(planet,'venus',5))
    %go=8.87;        % (m/s^2)
    rho_o=16.02;     % (kg/m^3)
    H=6226.65;       % (m)
    Ro=6052000;      % (m)
    ao = 529.6;      % Reference Speed(m/s) of sound near surface
    mu = 3.257*10^14; % Gravitational Parameter (m^3/s^2)
    omega = -2.9925539*10^-7; %Planetary Rotation Rate (rad/s)
elseif (strncmp(planet,'titan',5))
    %go=1.36;        % (m/s^2)
    rho_o=4.904;     % (kg/m^3)
    H=140000;        % (m)
    Ro=2720332;      % (m)
    ao = 194;        % Reference Speed(m/s) of sound near surface
    mu = 1.006428*10^13; % Gravitational Parameter (m^3/s^2)
    omega = 4.560686*10^-6; %Planetary Rotation Rate (rad/s)
end
go=mu/Ro^2;

%%Vehicle Parameters
%%
m=1179.34;          % Vehicle Entry Mass(kg)
A=14.3;             %Reference Area(m^2)
nose_radius = 0.6638; % nose radius of the ballistic entry vehicle (m)
Cd=1.53;            % Hypersonic Drag Coefficient (dimensionless)
L_over_D=0.29;      % Hypersonic in-plane L/D (dimensionless)
BCo=m/(Cd*A);       % Ballistic Coefficient at entry
emissivity=0.8;
initial_bank_angle=pi();
```

```

alpha_o=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Event Parameters (maximum of 5 events)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Event type
% 0 - No event
% 1 - Parachute Deploy (Cd, Diameter, inflation_time)
% 2 - Separation Event (change in mass)
% 3 - Parachute Release
% 4 - TERMINATE SIMULATION or begin Terminal Propulsion (thrust, isp, target_alt,target_V)
event_types = [0,0,0,0,0]; %specify events chronologically, use 0's to fill event_type to length of 5
%trigger_variables = {'q','Mach','alt','tmr3','tmr4'};
%trigger_values = [703, 0.6, 300, 1, 1];
%trigger_slopes = [-1, -1, -1, 1, 1]; %specify if the variable is increasing or decreasing
%event1_specs = [0.41, 12.5, 8];
%event2_specs = [73.9];
%event3_specs = [];
%event4_specs = [94];
%event5_specs = [3000,320,0,10]; %[0] if there is no terminal propulsion phase
guidance=1;
Kp(1)=1;

sos_type='const';
atm_type='EXPO';
aero_type = 'const';

```

Appendix C: List of STAPLE Files

Atmosphere Files: (Density vs. Altitude in Metric Units)

- 1962US_StandardAtm.dat - 1962 US Standard Daytime Atmospheric
- Kassatm.dat - Kass Atmosphere model for Mars
- MG2K_2012_AM.dat - MarsGRAM 2000 atmosphere generated for 2012 landing opportunity
- MG2K_MPF.dat - MarsGRAM 2000 atmosphere generated for MPF landing
- MPF_recon2.dat - MPF reconstructed flight atmosphere

GUI Files:

- AtmosphericInterfaceInput.m - GUI for Atmospheric Interface Condition Input
- CreateScript.m - Generates Script from GUI inputs
- EventInput.m - GUI for Event Input Parameters
- PlanetInput.m - GUI for Planetary Input Parameters
- TableInput.m - GUI for Input of Tabular Data
- VehicleParametersInput.m - GUI for Vehicle Input Parameters

Input Files:

- AFE_INPUT.m - Aeroassist Flight Experiment Aerocapture Input File
- MPF_INPUT.m - Mars Pathfinder Entry Input File
- MPF_cd_table.dat - Mars Pathfinder hypersonic drag coefficient table
- MPF_cl_table.dat - Mars Pathfinder hypersonic lift coefficient table

Auxiliary Functions:

- bank_command2.m - Generates bank angle commands for guidance
- Chapmans.m - Heat Rate Calculations
- Equilibriumwalltemperature.m - Equilibrium Wall Temperature Calculations
- Predict_apogee.m - Apogee Predictor for Aerocapture Guidance
- ReadAeroFromFile.m - Reads Aerodynamic Data from Comma Delimited Source
- ReadAtmosFromPOSTFile.m - Reads Atmospheric Data from POST files
- ReadAtmosFromTABLFile.m - Reads Atmospheric Data from Comma Delimited Source
- STAPLE_gturn.m - Calculates Time to Initiate Gravity Turn based on Trajectory

Main Function:

- STAPLE_V1.m - Main function

References

General

- 1 POST users manual
- 2 Braun, R.D., "Planetary Entry, Descent, and Landing Short Course Notes" <http://www.ae.gatech.edu/~rbraun/PlanetaryEDL.pdf>, August 2004.
- 3 Vinh, N.X., Buseman, A., Culp, R.D., Hypersonic and Planetary Entry Flight Mechanics, The University of Michigan Press, Ann Arbor, MI, 1980.(pp19-28)
- 4 Mease, K.D., Chen, D.T., "A Three-Dimensional Predictive Entry Guidance Approach," AIAA 2000-3959, 2000.
- 5 Regan, F.J., Anandakrishnan, S.M., *Dynamics of Atmospheric Re-Entry*, AIAA Education Series, Washington, D.C., 1993.
- 6 Martin, J.J., *Atmospheric Re-Entry*, Prentice-Hall, New Jersey, 1966
- 7 Buning, H., "Elements of Space and Re-Entry Mechanics" from *Flight Mechanics of Spacecraft*, Michigan University Summer Conference Proceedings, 1966

Planetary Data

- 8 Vallado, D.A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press, El Segundo, CA, 2001 (pp906-907)

MPF validation

- 9 Milos, F.S., Chen, Y.K., "Mars Pathfinder Entry Temperature Data, Aerothermal Heating, and Heatshield Material Response," *Journal of Spacecraft and Rockets* vol. 36 no.3 (pp380-391), 1999.
- 10 Spencer, D.A., Blanchard, R.C., "Mars Pathfinder Atmospheric Entry Reconstruction," AAS 98-146, 1998.
- 11 Spencer, D.A., "Mars Pathfinder Entry, Descent, and Landing Reconstruction," *Journal of Spacecraft and Rockets* vol. 36 no. 3 (pp357-366), 1999.
- 12 Desai, P.N., "Flight Reconstruction of the Mars Pathfinder Disk-Gap-Band Parachute Drag Coefficient," 17th AIAA Aerodynamic Decelerator Systems Technology Conference, Monterey, CA, May 2003.

Aerocapture Guidance and Validation

- 13 1962 Atmosphere, <http://roger.ecn.purdue.edu/~propulsi/propulsion/flow/stdatm.html>
- 14 Cerimele, C.J., Gamble, J.D. "A Simplified Guidance Algorithm for Lifting Aeroassist Orbital Transfer Vehicles," AIAA-85-0348
- 15 Masciarelli, J.P., Queen, E.M., "Guidance Algorithms for Aerocapture at Titan", AIAA 2003-4804