AE 8900 Individual Research Project

Qualitative and Quantitative Assessment of Optimal Trajectories by Implicit Simulation (OTIS) and Program to Optimize Simulated Trajectories (POST)

Prepared by: Doug Nelson
Submitted to: Dr. John Olds
Georgia Institute of Technology
April 26, 2001

Table of Contents

List of Figures

List of Tables

Abstract

There is an ongoing debate among aerospace professionals about which trajectory optimization program calculates the best results. The two programs that are most often included in the debate are Optimal Trajectories by Implicit Simulation (OTIS) written by The Boeing Corporation in conjunction with NASA-Glenn Research Center (GRC) and the Program to Optimize Simulated Trajectories (POST) written by Lockheed Martin Astronautics and NASA-Langley Research Center (LaRC).

The main difference between the two programs is the way that they represent the physics of the problem. POST uses the more traditional direct shooting approach that calculates the state variables as a function of time throughout the entire trajectory. This guarantees that the physics of the problem are accurate at all times during the simulation. On the other hand, OTIS has the capability to solve the trajectory problem in more than one way. In addition to explicitly calculating the trajectory with direct shooting when operating in Mode 3, OTIS can also solve the problem implicitly when run in Mode 4. The implicit method used in Mode 4 is known as the collocation method and uses a series of polynomials to represent the state variables. During the actual process of solving the problem, there is no guarantee that the problem satisfies all of the physics of the problem until it is solved.

The impetus of this project was the fact that both programs must satisfy the same laws of physics in the end and should therefore not predict widely differing results. Instead, the expectation was that each program would arrive at very nearly the same optimum trajectory with minor differences attributable to different optimizers and solution techniques. To prove this theory, both programs were used to optimize the trajectories of two separate launch vehicles:

1) ACRE-92 – A VTHL rocket-powered single stage to orbit vehicle
2) *Hyperion* – An HTHL rocket-based combined cycle vehicle

The results of the work support the initial theory. In both cases, the mass ratio (MR) calculated using OTIS differed from that calculated using POST by less than one half of one percent. The real differences in the two programs were more qualitative than quantitative as each requires slightly different styles of inputs and presents its own unique challenges to the user.

Acronyms and Symbols

ACRE-92	Advanced Concept Rocket Engine with sea-level thrust to weight of 92
C_d	Coefficient of Drag
C_l	Coefficient of Lift
CPU	Central Processing Unit
g	A unit of acceleration equal to one times Earth’s gravitational acceleration at the surface
GLOW	Gross Lift-Off Weight
GRC	Glenn Research Center
HTHL	Horizontal Take-Off Horizontal Landing
I_{sp}	Specific Impulse
I*	Specific Impulse Adjusted for All Losses
LaRC	Langley Research Center
MECO	Main Engine Cut-Off
MR	Mass Ratio
NASA	National Aeronautics and Space Administration
OTIS	Optimal Trajectories by Implicit Simulation
POST	Program to Optimize Simulated Trajectories
psf	Pounds per square foot
psi	Pounds per square inch
q	Dynamic Pressure
RBCC	Rocket-Based Combined Cycle
SSDL	Space Systems Design Laboratory at the Georgia Institute of Technology
SSTO	Single Stage to Orbit
VTHL	Vertical Take-Off Horizontal Landing

α	Angle-of-Attack
β	Sideslip Angle
ϕ	Bank Angle
γ	Flight Path Angle

1.0 Introduction

In the arena of conceptual launch vehicle design, one of the most critical and time consuming disciplines is that of predicting vehicle performance. It is important to predict the performance as accurately as possible because what appear to be minor changes in a vehicle's mass ratio (MR) can cause the predicted initial mass of the vehicle to skyrocket and result in an infeasible design over the course of a design iteration sequence. As a result, much attention is paid to the particular tool that is used to predict the trajectory. The two trajectory optimization computer programs most often used are:

1) Optimal Trajectories by Implicit Simulation (OTIS) written by Boeing and NASA-Glenn Research Center [1]
2) Program to Optimize Simulated Trajectories written by Lockheed Martin Astronautics and NASA-Langley Research Center[2]

Over the years, two distinct trajectory optimization camps have formed and the battle lines between the two have been drawn. The divide between the two camps has risen to the point where an OTIS user might not believe trajectory results calculated with POST and vice-versa simply because their tool of choice was not used. In most cases, the tool that is learned first is the one to which most people develop loyalty. As a result, few individuals know how to use both tools and are therefore unable to knowledgeably debate the merits of the competing code.

Since the laws of motion and the physics of flight are the same regardless of which tool is used, there is no obvious reason why one tool should be any less accurate than the other because the end result from each must satisfy all known physical laws. Based on this line of reasoning, one would expect only minor differences in the final answer produced by the two codes. These small differences would be attributable to minor differences in control algorithms, solution techniques, and optimizers.

To test the theory that the two codes should predict the same basic answer, two separate vehicles were flown using both POST and OTIS. The first vehicle is a rocket-powered single stage to orbit (SSTO) vehicle named ACRE-92 and the second is a rocket-based combined cycle (RBCC) vehicle named *Hyperion*.

2.0 POST and OTIS Basics

This section presents the basic workings of the two computer programs. The similarities and differences between the two are also emphasized.

2.1 Integration Technique

POST works by integrating the trajectory in time using the direct shooting technique. Throughout the simulation, POST requires all physical laws to be satisfied at every point in the trajectory. In contrast, OTIS has the ability to simulate the trajectory in two distinct ways. In Mode 3, OTIS employs the direct shooting method used in POST while Mode 4 allows OTIS to implicitly solve the trajectory problem using the collocation method. The collocation method does not require the program to incrementally integrate the trajectory with respect to time. Instead, collocation fits polynomials to the state variables based on a user-specified number of nodes and node spacing. During the solution process, OTIS alters the positions of these nodes until all of the user imposed constraints as well as all physical laws are met. The benefit of using collocation is that it provides more flexibility to the program in its attempt to solve the problem by theoretically giving the optimizer an infinite number of degrees of freedom[1]. While this freedom can be helpful in solving the problem, the downside is that there is no guarantee of a physical solution until the problem is completely solved[1]. Since the collocation capability is advertised as the feature in OTIS that really sets it apart from POST, Mode 4 was used exclusively for this project.

Another feature of the implicit solution is that the answer must be checked by explicitly integrating the resulting trajectory. This allows the user to assess the level of defects in the implicit solution by comparing it to the direct solution. The downside is that running the explicit trajectory is an additional step that must be completed before releasing the results.

2.2 User Interface

Both programs use similar namelist input files with each requiring the user to divide the problem up into a series of phases or events. In POST, the program automatically assumes continuity of the majority of variables between phases whereas OTIS requires the user to indicate what, if any, control and quadrature variable continuity there is between phases[†]. The

[†] Both OTIS and POST require that state variables are continuous across a phase boundary

lack of presumed continuity in OTIS allows the user to have more control over the trajectory but it can also create problems for the inexperienced user.

2.3 Optimization

Both programs provide the user with a choice of optimization algorithms. POST allows the user to choose between a projected gradient system based on the Method of Useable and Feasible Directions and a non-linear constrained optimization routine called NPSOL that was written by the Stanford Business Software Optimization Laboratory (SBSOL). OTIS provides the user with the choice of three different optimization algorithms: SLSQP, SNOPT, and NPOPT. SLSQP and NPOPT are sequential quadratic programming optimizers that are used mainly for problems that are explicitly integrated or for relatively small implicitly integrated problems. SNOPT is recommended for the majority of Mode 4 problems in OTIS and is described in more detail in the following paragraph. When the user acquires the OTIS program from NASA-GRC, SLSQP is the only optimization package available. The other two are commercially available and must be purchased from SBSOL and integrated into OTIS by the user. While OTIS does not ship with SNOPT or NPOPT, instructions for compiling them into the code are available and relatively easy to follow.

Since OTIS' strength lies in its ability to implicitly integrate trajectories using collocation, it is computationally more efficient to use a sparse optimizer when running a collocation problem. The reason for this is explained in the following excerpt from pages 130-131 of the OTIS 3.10 user's manual:

> In Mode 4,the independent variables have a much more limited influence on the implicitly integrated trajectory. Influence coefficients are analytically computed. Because of this the calculation of the Objective Gradient and Constraint Jacobian do not require, relatively speaking, a great amount of CPU time. The problem size (number of independent variables and constraints) in Mode 4, however, is relatively large. This large problem size coupled with the fact that each independent variable has a limited influence on the constraints, results in a sparsely populated Jacobian matrix. This "sparse" type of matrix requires a lot of CPU time to be processed by a traditional "dense" optimizer. This is caused by the fact that the sparse matrix has large dimensions while the dense optimizer will process the entire matrix. On the other hand, when using a sparse optimizer, the sparse matrix will be re-mapped taking advantage of its "sparseness ". It is this much smaller re-mapped matrix that is actually used in the optimization process resulting in a significant savings in CPU run time[3].

Based on this reasoning, SNOPT, a sparse optimizer, was used to complete the OTIS simulations performed for the study. As mentioned previously, SNOPT is a commercially available optimizer that is easily integrated into OTIS to take advantage of the reduction in computation time and make use of all of OTIS' advertised strengths.

3.0 Quantitative Comparison

The main goal of this project was to address the question of which trajectory optimization program calculates the "better" answer where the "better" solution is the one that results in the lowest MR at orbit insertion[‡]. To answer this question, two different types of space launch vehicles were simulated with each program and the results compared. The first vehicle is a conceptual design for a rocket powered SSTO named ACRE-92 developed by Boeing Rocketdyne that is based on a shape originated at the NASA-Langley Vehicle Analysis Branch[4]. The second vehicle is an RBCC SSTO vehicle named *Hyperion* that was designed by students in the Space Systems Design Laboratory at the Georgia Institute of Technology[5]. Both vehicles were flown into a 50 nmi x 100 nmi transfer orbit with an inclination of 28.5°. Additionally, both vehicles were launched from Kennedy Space Center (KSC) at 280° East longitude and 28.5° North latitude.

3.1 ACRE-92

The ACRE-92 is a conceptual design for a vertical takeoff horizontal landing (VTHL) SSTO that gets its name from the Advanced Concept Rocket Engine that it uses as its propulsion system. The 92 indicates the engine has a sea-level thrust to weight of 92. Figure 1 shows an artist's conception of a vehicle similar to ACRE-92 landing on a runway.

[‡] For this project, quantitative refers to the actual numerical answers calculated by each program. The two programs were also compared qualitatively in the following section. The qualitative comparison was mainly concerned with how user-friendly the two programs are.

Figure 1: Launch Vehicle Similar to ACRE-92[4]

The objective of the simulation using both codes was to maximize the burnout weight of the vehicle. Figure 2 is a schematic of the trajectory simulation modeled in both programs. The flight begins with a vertical launch of the vehicle from a launch pad at KSC. This is followed by a vertical rise until the vehicle reaches 400 feet of altitude. At this point, the optimizer is given control of the vehicle pitch angles so that it can pick the best trajectory. The trajectory terminates with main engine cutoff (MECO) once the vehicle accelerates to an inertial velocity of an inertial velocity of 25842 ft/sec. The four constraints imposed on the trajectory are as follows:

1. The vehicle must have a geocentric radius of 2,122,950 feet at MECO
2. The vehicle must have a flight path angle (γ) of zero degrees at MECO
3. The vehicle must have an orbital inclination of 28.5° at MECO
4. The maximum acceleration on the vehicle must not exceed three Earth g's.

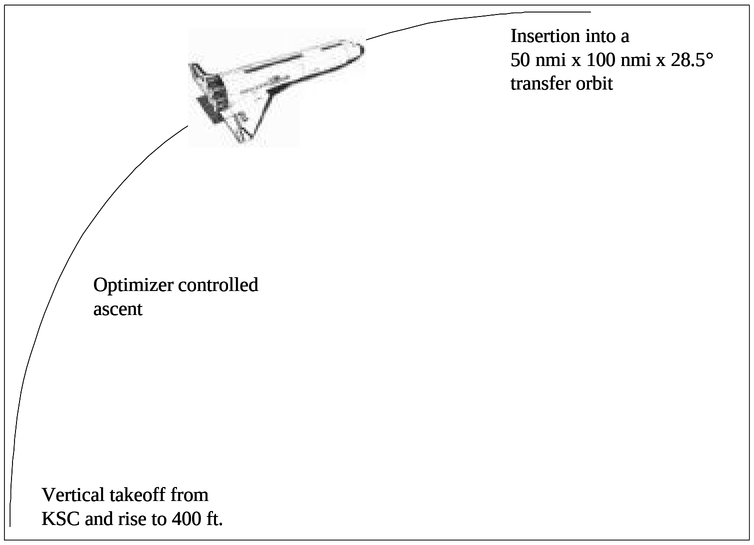

Figure 2: ACRE-92 Simulation Schematic

Table 1 summarizes the parameters used in setting up the problem in the two programs. Looking at the table one sees that both codes required three phases to model the trajectory. POST used five different pitch terms as the independent variables in the explicitly solved problem. On the other hand, the OTIS simulations used a total of fifty-two nodes and four placards to solve the problem implicitly[§]. The complete POST input file is included as Appendix A while the full OTIS input file is included as Appendix B.

Table 1: ACRE-92 Problem Setup Summary

POST Setup	
Number of Phases	3
Number of Pitch Controls	5
OTIS Setup	
Number of Phases	3
Total Number of Nodes	52
Total Number of Placards	4

After flying ACRE-92 using both OTIS and POST, the resulting trajectories were almost exactly the same. Table 2 lists the gross liftoff weight (GLOW), burnout weight, and mass ratio calculated with each program. Looking at the table, one sees that POST calculated a slightly higher burnout weight and correspondingly smaller mass ratio. Considering the two mass ratios differ by only 0.16% one can conclude that there is no significant difference in the results produced by the two codes.

Table 2: ACRE-92 Results

OTIS GLOW	2100000 lbs
OTIS Burnout Weight	277536 lbs
OTIS MR	7.567
POST GLOW	2100000 lbs
POST Burnout Weight	277966 lbs
POST MR	7.555 lbs

A closer look at the results returned by the two different programs show additional similarities. Table 3 lists the values returned for two different integrated velocity loss components in the relative frame as well as the ideal velocity increment that the propulsion

[§] In OTIS, the constraints placed on each phase of the trajectory are called placards.

system needs to produce in the relative frame. Looking at these results, one sees that OTIS calculates slightly higher gravity losses and slightly lower drag losses than POST. However, the overall ideal velocity reported by the two programs is nearly the same.

Table 3: ACRE-92 Velocity Losses in the Relative Frame

OTIS Drag Losses	624 ft/s
OTIS Gravity Losses	3808 ft/s
OTIS Ideal Velocity	29394 ft/s
POST Drag Losses	803 ft/s
POST Gravity Losses	3567 ft/s
POST Ideal Velocity	29371 ft/s

To get a better feel for how the two codes compare to each other, it is helpful to see a graphical representation of the data. Looking at Figure 3, one sees that the overall angle-of-attack as a function of time for the two separate simulations is remarkably similar. One thing to notice is that early in the trajectory (the period up to 60 seconds) when the vehicle is lowest in the atmosphere and subject to drag forces from the higher density air, the POST simulation reports a higher coefficient of drag (C_d) than the OTIS simulation. Looking at Figure 4, one sees that the POST simulation reports a higher angle of attack for most of this same period. These results help explain the slightly higher drag losses in POST. Additional differences can be attributed to the fact that OTIS used a quintic spline fit of the aerodynamic data provided while POST used a simple linear interpolation.

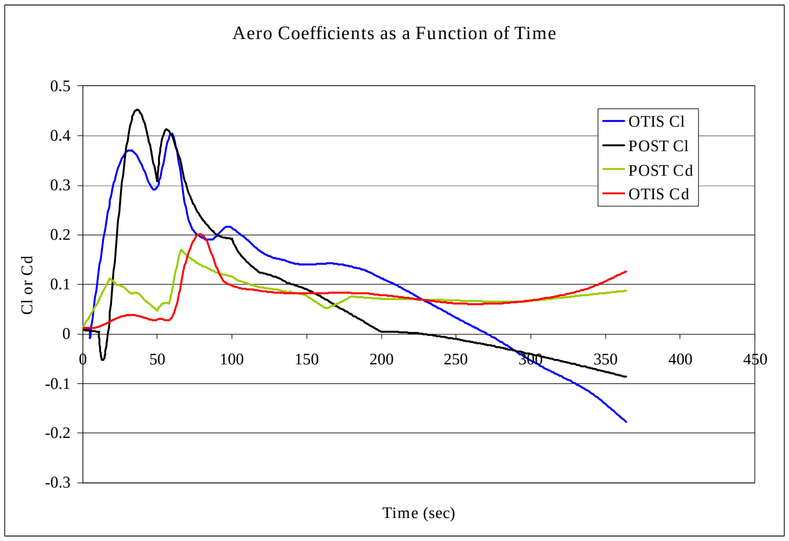

Figure 3: ACRE-92 Aerodynamic Coefficients as a Function of Time

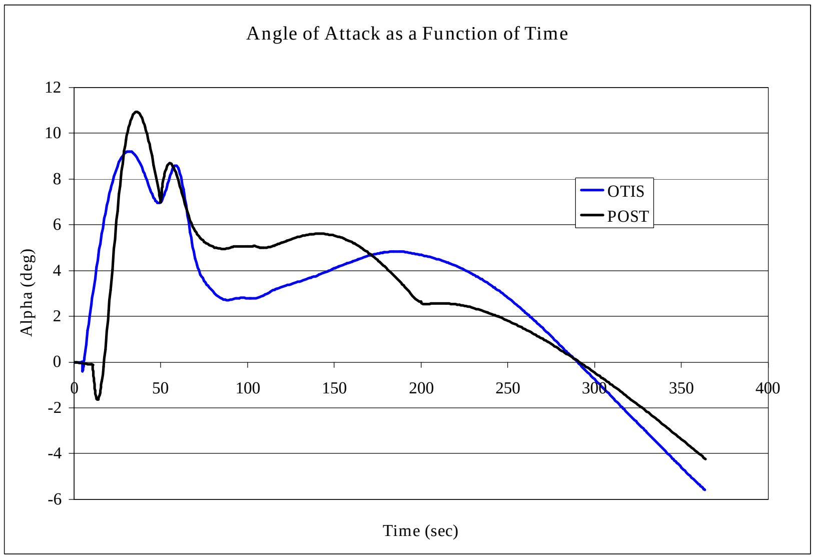

Figure 4: ACRE-92 Angle of Attack as a Function of Time

Figure 5 shows that the slight differences in velocity losses reported in Table 3 have little effect on the overall trajectory. In the figure, one sees that the vehicles have nearly identical altitude versus time histories.

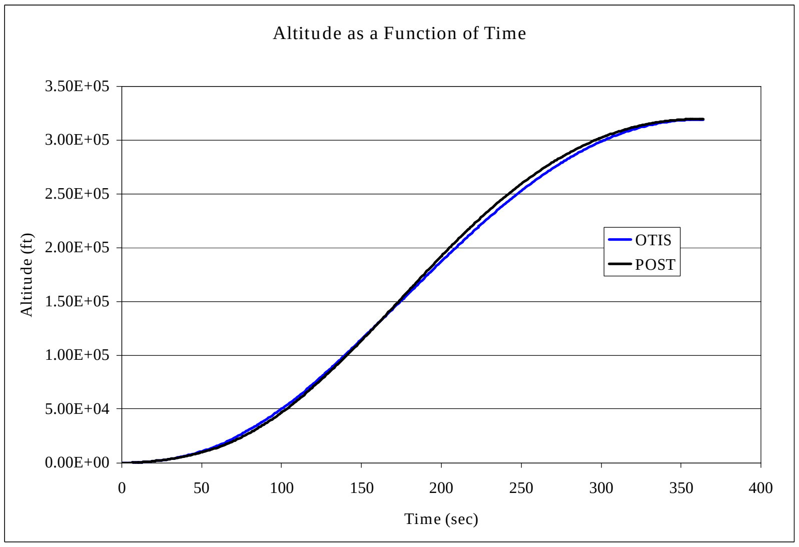

Figure 5: ACRE-92 Altitude as a Function of Time

Figure 6 is a representation of the weight history reported by each program as a function of time. Looking at the figure, one is unable to discern any difference between the two simulations. This indicates that the two programs are both able to correctly calculate the mass flow rate of propellants given vacuum thrust and specific impulse (I_{sp}) values.

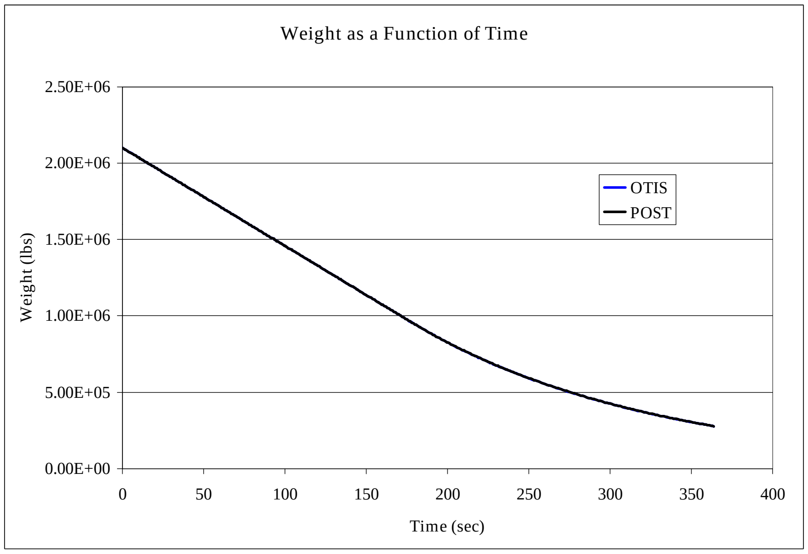

Figure 6: ACRE-92 Weight as a Function of Time

A concern that a user might have when modeling a rocket-powered vehicle as it ascends through the atmosphere is how well the computer program can adjust for the back pressure effects on the rocket engine. Figure 7 shows that both programs accurately adjust for these effects. The slight differences in the thrust values are due to the small differences in altitude over time shown in Figure 5. Additionally, Figure 8 shows that both codes were able to throttle the rocket engine to limit the axial acceleration to the desired limit of 3 Earth g's.

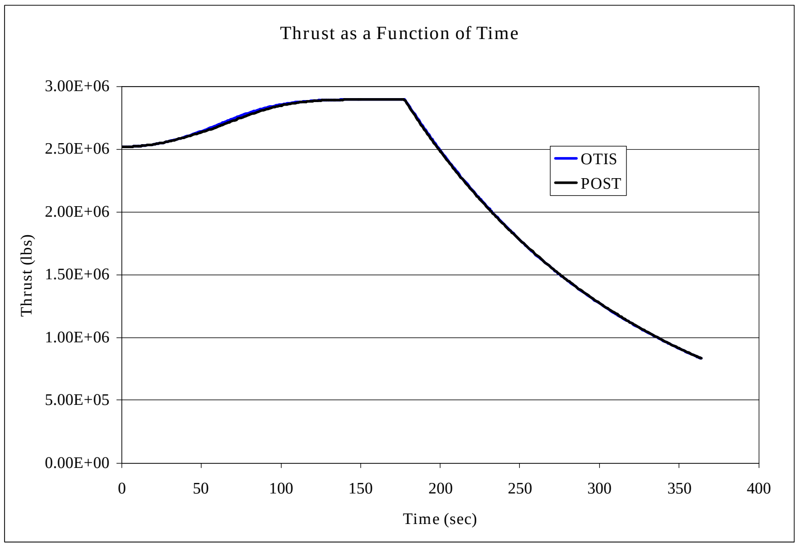

Figure 7: ACRE-92 Thrust as a Function of Time

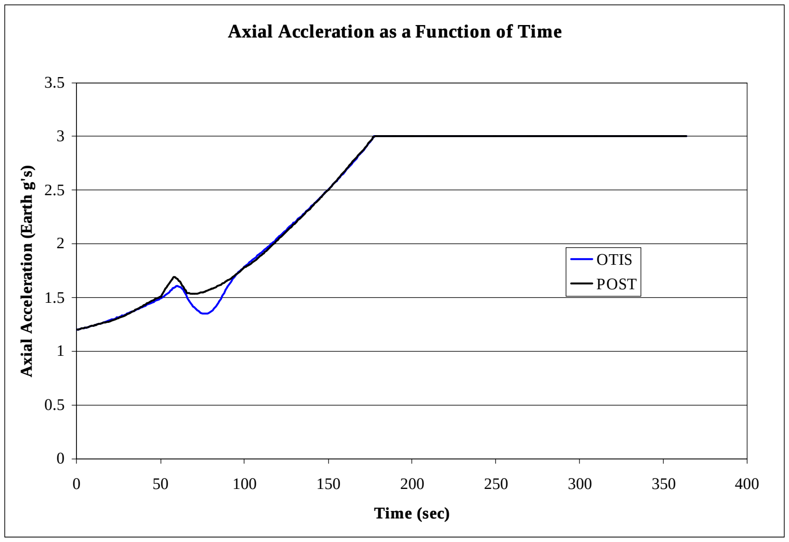

Figure 8: ACRE-92 Axial Acceleration as a Function of Time

3.2 *Hyperion*

Hyperion is a conceptual design for a horizontal takeoff horizontal landing (HTHL) vehicle that uses a RBCC engine as its main propulsion system. Figure 9 is an artist's conception of *Hyperion* deploying its payload.

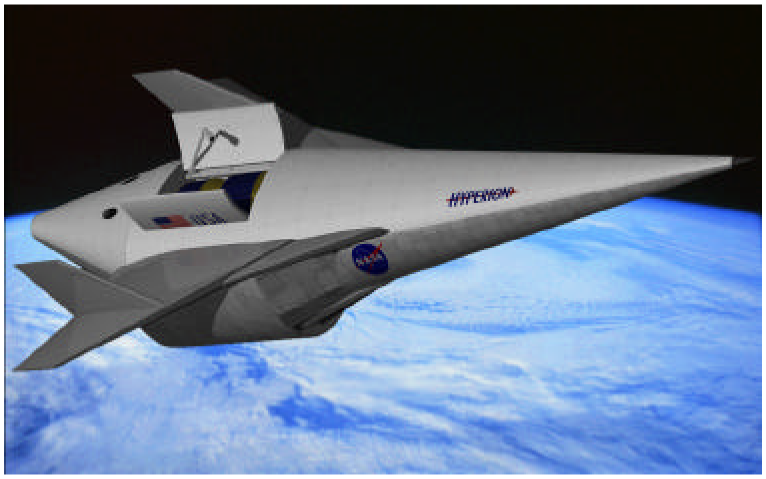

Figure 9: *Hyperion* Launch Vehicle Deploying Payload[6]

As in the ACRE-92 simulation, the objective of the simulation was to maximize the burnout weight of the vehicle. Figure 10 is a schematic of the trajectory simulation modeled in both programs. The flight begins with a horizontal takeoff from a runway at KSC. This is followed by an optimizer-chosen q-boundary during the ejector and air-breathing engine modes. Once the vehicle reaches a Mach number of 10, the rocket engine is turned on and the vehicle follows an optimizer-chosen pitch profile to MECO once the desired 50 nmi x 100 nmi x 28.5° transfer orbit is reached. The optimizers were required to find the best trajectory while meeting the following five constraints:

1. The free stream dynamic pressure must be less than or equal to 2000 psf
2. The wing normal force must be less than or equal to 1.75 times the gross takeoff weight of the vehicle (For *Hyperion*, this limit is 1,968,750 lbs)
3. The flight path angle at orbit insertion must be zero degrees
4. The geocentric radius at orbit insertion must be 2,122,950 feet
5. The maximum static pressure in the engine during ramjet mode must not exceed 250 psi

An additional feature of both models was that the engine mode transition Mach numbers were fixed by the propulsion system so that the optimizer had no control over them. The transition from ejector mode to ramjet operation takes place between Mach 2.9 and 3.1 while the transition from ramjet to scramjet mode occurs between Mach 5.9 and 6.1. The rocket is throttled up over 15 seconds once the vehicle reaches Mach 10 and the scramjet is throttled down linearly between Mach 10 and 11.

One difference between the two simulations is that the OTIS simulation was unable to model the take-off roll of the trajectory. Because of this limitation in the program, the POST simulation was run first. This made it possible to input the initial correct lift equals weight conditions into OTIS as well as allow the OTIS simulation to start with a weight value that has been adjusted to account for fuel burned during the take-off roll. As a result, the OTIS simulation begins at approximately thirty seconds instead of zero.

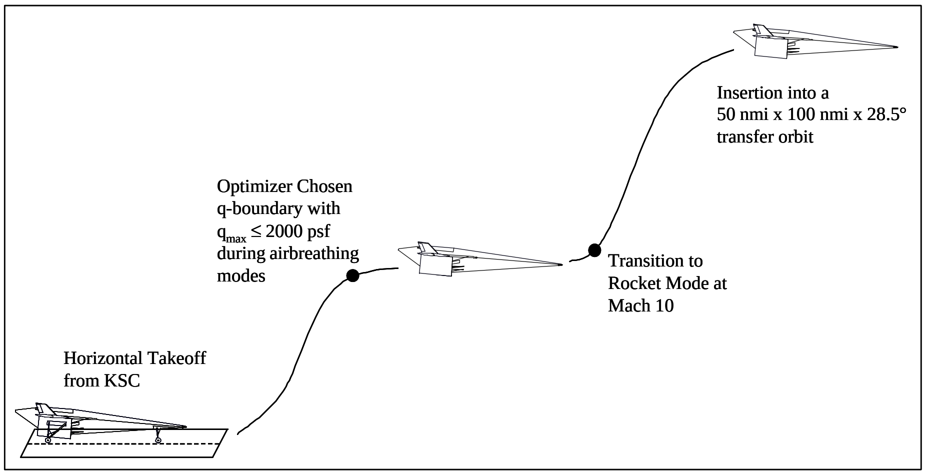

Figure 10: *Hyperion* Flight Simulation Schematic

The pertinent setup parameters for the *Hyperion* simulation are summarized in Table 4. The table shows that the POST simulation used a total of twelve different phases whereas the OTIS simulation used nine. This is somewhat misleading because several of the phases in POST were used to simply change the time increment at which the program wrote to the output file and had no real bearing on the solution. Additionally, the table shows that the POST simulation used ten different pitch values and four q-boundary values for a total of fourteen independent variables. The q-boundary controls include one control that allows POST to determine when the vehicle gets on the q-boundary and three other controls that allow POST to change the q-boundary itself. In contrast, the OTIS simulation used a total of 272 nodes over the entire simulation as well as thirty-one placards. The total number of placards is somewhat misleading because OTIS does not allow a particular placard to apply to more than phases. So if the intent is to limit the value of a particular variable over the course of several phases, the placard for that variable must be repeated in each phase. The full POST input file for the *Hyperion* simulation is included as Appendix C while the corresponding OTIS input file is included as Appendix D.

Table 4: *Hyperion* Problem Setup Summary

POST Setup	
Number of Phases	12
Number of Pitch Controls	10
Number of q-Boundary Controls	4
OTIS Setup	
Number of Phases	9
Total Number of Nodes	272
Total Number of Placards	31

Using OTIS and POST to optimize the trajectory for the *Hyperion* vehicle subject to the requirements listed above resulted in the data presented in Table 5. As with ACRE-92, the end result obtained by the two programs was nearly identical. The final weight calculated by POST is 0.21% higher than that calculated by OTIS and the two mass ratios differ by the same percentage.

Table 5: *Hyperion* Results

OTIS GLOW	1125000 lbs
OTIS Burnout Weight	230525 lbs
OTIS MR	4.880
POST GLOW	1125000 lbs
POST Burnout Weight	231017 lbs
POST MR	4.870 lbs

Looking at the relative drag and gravity losses listed in Table 6, one sees that POST calculated slightly higher losses in both categories. As a result, POST also reports a higher ideal velocity increment required from the propulsion system than OTIS does. Since OTIS calculated the lower ideal velocity, one might expect the OTIS simulation to calculate the lower mass ratio even though Table 5 shows the opposite is true. However, looking at the I_{sp} adjusted for all losses (I*) listed in Table 6 shows that POST calculated an I* that is almost a full second higher than the value calculated by OTIS. This explains why the POST mass ratio is slightly lower than OTIS.

Table 6: *Hyperion* Velocity Losses in the Relative Frame

OTIS Drag Losses	7419 ft/s
OTIS Gravity Losses	2081 ft/s
OTIS Ideal Velocity	33569 ft/s
OTIS I*	479.531 sec
POST Drag Losses	8255 ft/s
POST Gravity Losses	1844 ft/s
POST Ideal Velocity	34724 ft/s
POST I*	480.410 sec

The high degree of similarity between the angle-of-attack histories exhibited in Figure 11 shows that the guidance algorithms used in both programs produce comparable results for a vehicle like *Hyperion*. This fact is emphasized by the close agreement in the calculated aerodynamic coefficients charted in Figure 12.

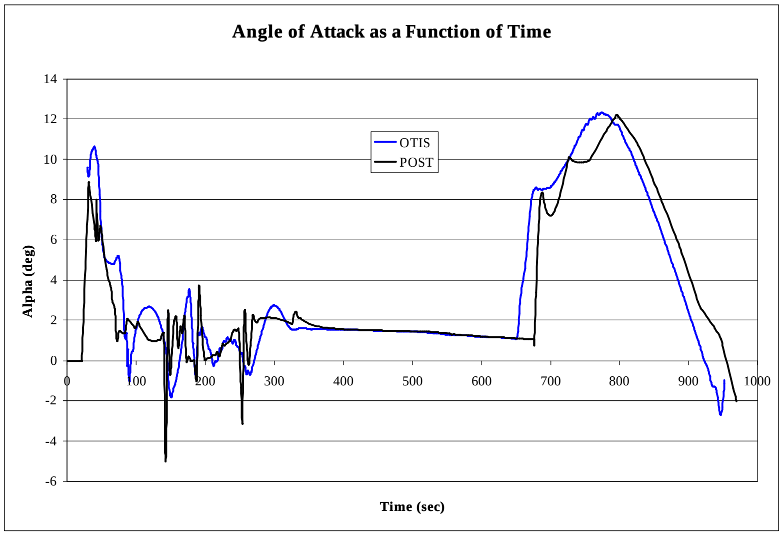

Figure 11: *Hyperion* Angle of Attack as a Function of Time

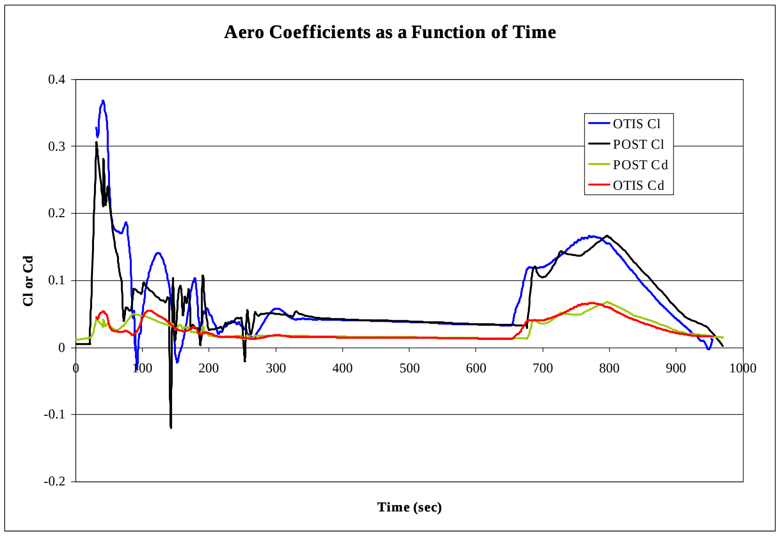

Figure 12: *Hyperion* Aerodynamic Coefficients as a Function of Time

Figure 13 and Figure 14 show that the two programs did not command the *Hyperion* vehicle to fly trajectories as similar as the angle-of-attack history seems to suggest. Figure 13 shows that OTIS is able to accelerate the vehicle to the desired velocity in a shorter amount of time than POST is. Similarly, the weight of the vehicle decreases faster in the OTIS model than in the POST model. While these differences are noticeable early on, they become more pronounced during the all rocket mode at the end of the trajectory. Figure 15 shows one reason for the differences in the weight and altitude histories calculated by the two codes. This figure shows that the two codes interpolated different thrust and thrust coefficient values from the supplied tables. These different values affect both the fuel consumption (change in weight) as well as the acceleration (acceleration will affect the altitude) of the vehicle. Similarly, the thrust profiles will affect the overall length of the trajectories. This helps to explain why the POST trajectory requires slightly more time to reach orbit than the OTIS trajectory.

The reason the thrust profiles calculated by the two programs differ is that the thrust during the air-breathing modes is a function of the dynamic pressure on the vehicle. Looking at Figure 16 one sees that the two optimizers chose different values of dynamic pressure between Mach 0.9 and 5 resulting in a difference in thrust between the two programs.

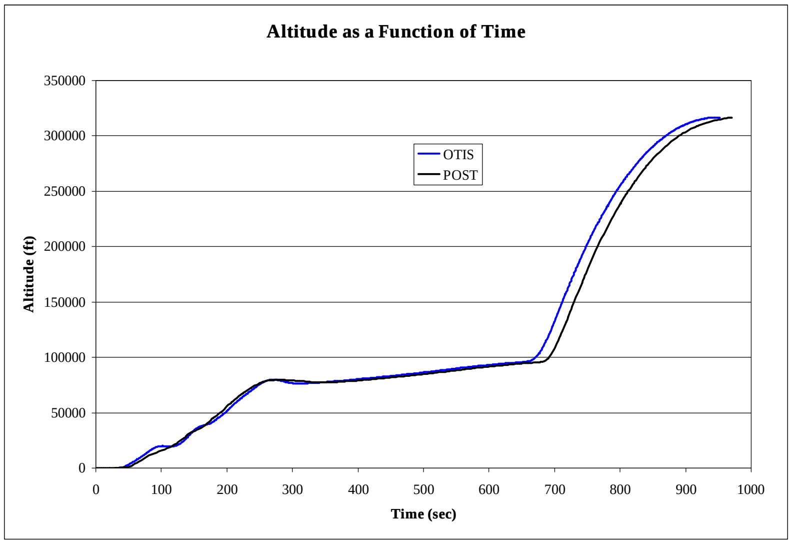

Figure 13: *Hyperion* Altitude as a Function of Time

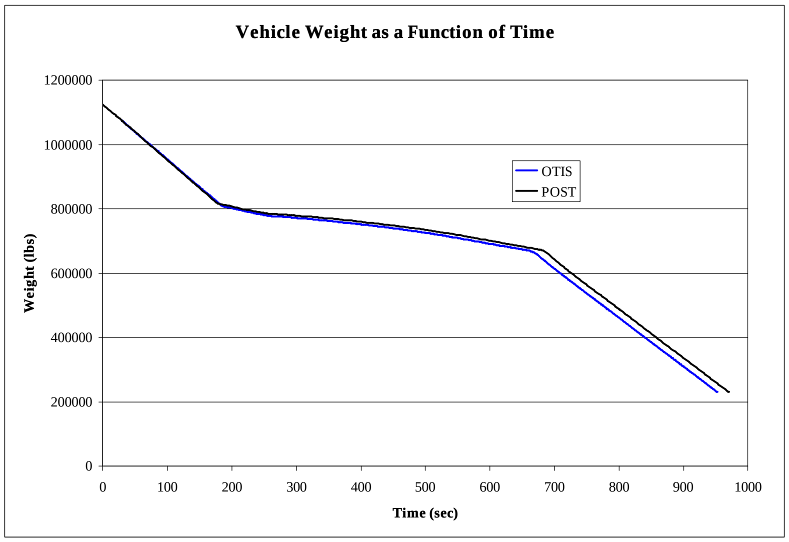

Figure 14: *Hyperion* Weight as a Function of Time

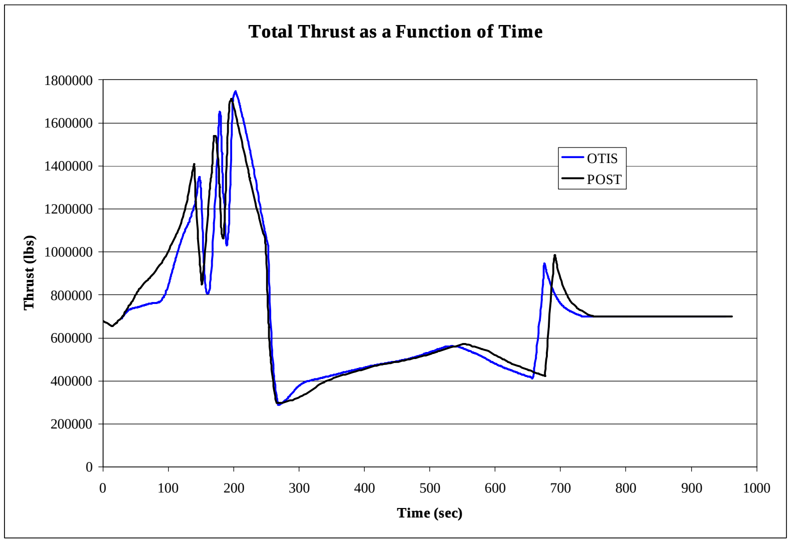

Figure 15: *Hyperion* Thrust as a Function of Time

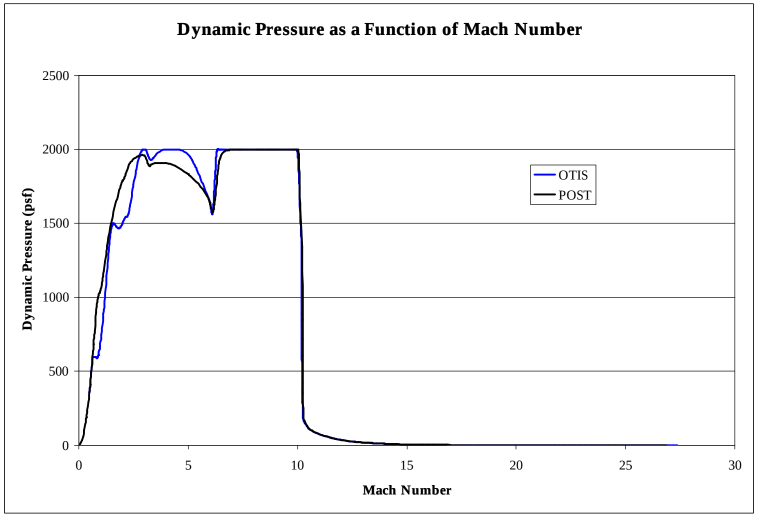

Figure 16: *Hyperion* Dynamic Pressure as a Function of Mach Number

As stated previously, two of the constraints imposed on the simulations were that the static pressure in the engine during ramjet mode could not exceed 250 psi and that the maximum wing normal force could not exceed 1.75 times the gross liftoff weight of the vehicle. Figure 17 is a graph of the engine static pressure calculated during ramjet mode by each program. The red line on the graph is the maximum allowable. The figure shows that both programs were able to successfully limit the value of the engine static pressure within the tolerances set for the constraint. Similarly, Figure 18 is a graph of the wing normal force imposed on the vehicle during the flight with the red line again showing the constraint limit**. Again, both programs were able to successfully meet the constraint.

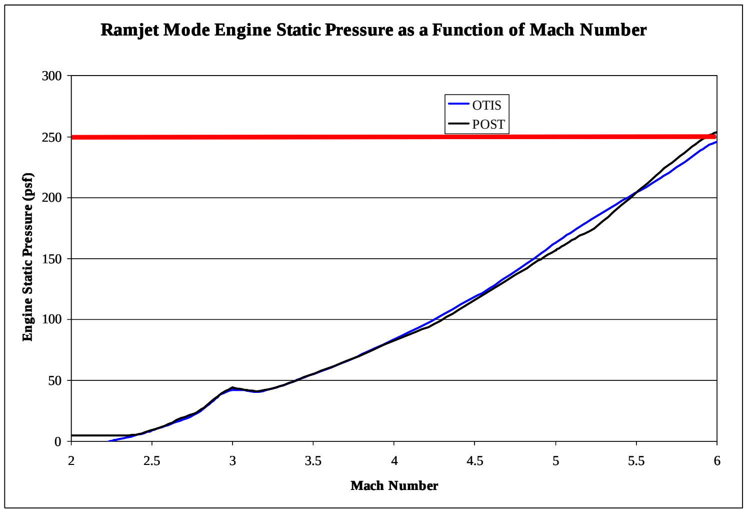

Figure 17: *Hyperion* Ramjet Mode Engine Static Pressure as Function of Mach Number

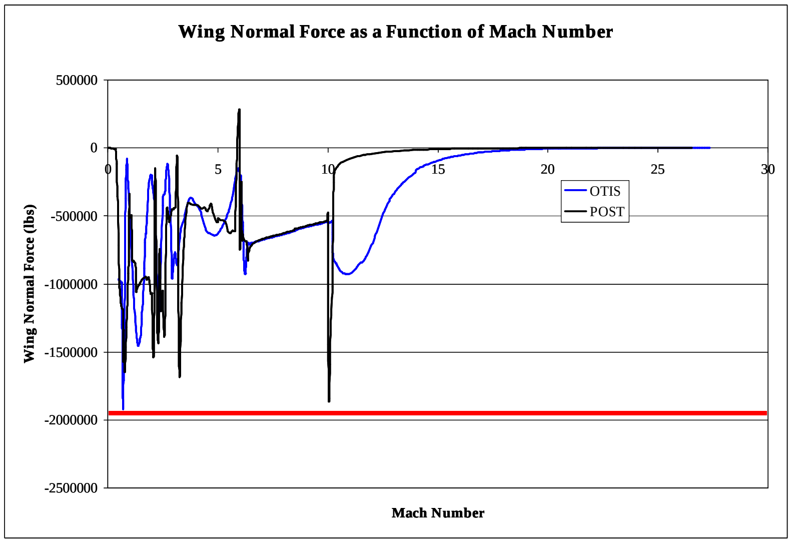

Figure 18: *Hyperion* Wing Normal Force as a Function of Mach Number

4.0 Qualitative Comparison

Both programs have their strengths and their weaknesses when it comes to the qualitative category. That is to say that in certain ways POST is better than OTIS while in other ways OTIS

** OTIS and POST use slightly different vehicle body axis frames. As a result, a positive wing normal force in OTIS corresponds to a negative wing normal force in POST. Consequently, the OTIS wing normal forces were multiplied by -1 to allow for a more direct comparison with the POST data.

is superior to POST. In the following two subsections, these differences are highlighted in the order that they were discovered over the course of the project.

4.1 Lessons Learned from the ACRE-92 Analysis

Since the ACRE-92 analysis was the author's first experience with OTIS, it provided some insight into the general characteristics of the code. The first thing noticed was that OTIS allows the user to create simple plots of any of the available output variables directly in the output file of the program. This makes it simple to see if specific constraints imposed on the trajectory have been met and whether or not the vehicle flew as expected. For example, the ACRE-92 vehicle is required to limit the axial acceleration to 3g because it is designed to carry passengers. To check this, the user can instruct OTIS to create an ASCII plot of the axial acceleration on the vehicle as a function of time so that it is easy to verify that the constraint has been met. To see a similar plot when running POST, the user must have access to a Tektronix emulator window along with the ability to run a program called jplot. Alternatively, one can use an available program to convert the binary output of POST into a file that can be plotted with the mathematical software package MATLAB. A final option is to instruct POST to write an ASCII output file and then port the file to a personal computer to graph with plotting package such as Microsoft Excel. Regardless, the POST user must have access to a separate plotting package to be able to see the results of the simulation graphically.

Another strength of OTIS is the ability to create custom variables within the input file. If one wants to calculate the mixture ratio of a vehicle, the appropriate equation can be entered without having to edit source code. This user-defined variable is then treated like any other variable in OTIS and can be used as a constraint or independent variable. In contrast, POST has the capability to calculate simple ratios and sums but nothing more complex without editing and recompiling the source code.

The most noticeable difference between the two programs manifested itself in the problem setup. With POST, the user must specify initial guesses for all of the independent variables. If care is not taken in choosing these variables, then POST might not be able find a viable solution to the problem. However, the implicit nature of OTIS eliminates the need to set specific initial guesses for the independent variables. The user simply tells OTIS which variables are independent and the program takes it from there. This feature does not mean that the user is relieved of responsibility in setting up a problem in OTIS. This is because unless instructed differently, OTIS will change any variable in an attempt to calculate a better answer. For example, if the user does not correctly set the flag that tells OTIS to fix the initial weight of the vehicle as specified by the user, OTIS will assume that initial weight is something that can be changed! Since most trajectory optimization simulations try to maximize burnout weight, this oversight can have an enormous effect on the final result.

While the collocation method used in OTIS provides many benefits, it also provides challenges to the user. The most important thing that the user needs to check at the conclusion of a simulation is the continuity of the control and quadrature variables across the phases. When initially running ACRE-92, OTIS would instantaneously change the angle of attack of the vehicle or the flight path angle by over 90° at the phase boundary. Physically, such a change is impossible but since OTIS did not know that the variables in question needed to be continuous across the phase boundary, it saw no reason to impose continuity. However, this is not a fatal flaw of the program. Once a user is more familiar with the code, checking for this type of error becomes second nature. It is something that a new user needs to bear in mind when learning how to use the program. Because POST exclusively uses an explicit integration, it does not suffer from this type of problem very often. Depending on the guidance scheme selected, angle of attack (α), sideslip angle (β), and bank angle (ϕ) may have some discontinuities between phases and therefore may need to be watched by the user. Otherwise, the direct shooting technique used in POST automatically enforces continuity of the other variables across all phase boundaries.

Aside from the difficulties involved in using the collocation method, OTIS lacked an additional nicety supplied by the programmers who wrote the POST source code – the ability to specify maximum acceleration limits. When modeling a vehicle that is designed to carry passengers, one usually wants to limit the axial acceleration of the vehicle to keep the ride from being dangerously uncomfortable for the passengers. In POST, limiting the acceleration is as simple as setting a flag and telling the program what the maximum allowable acceleration should be. At the appropriate point in the trajectory, POST throttles down the engine to keep the acceleration within limits. To do the same in OTIS, the user must introduce a multiplier on the thrust data that OTIS can vary to adjust the thrust level of the engine. Additionally, a phase must be added to tell OTIS that it needs to throttle the vehicle to keep from exceeding the maximum allowable acceleration.

4.2 Lessons Learned from the *Hyperion* Analysis

Since the initial differences between OTIS and POST were discovered during the ACRE-92 analysis, there were few new lessons learned during the *Hyperion* analysis. However, one difficulty with OTIS that was discovered during this analysis relates to horizontal take-off vehicles. When modeling horizontal take-off vehicles such as *Hyperion* in POST, the user can easily simulate the take-off roll down the runway by turning on a flag that holds the vehicle at the specified altitude until the flag is turned off. This is not possible in OTIS because no such flag exists. When trying to simulate the take-off roll down the runway, OTIS is unable to maintain the runway altitude because the low speeds of the take-off roll do not generate enough lift to offset the vehicle weight. As a result, by the time the vehicle has enough lift to raise the vehicle, it has fallen to a physically impossible altitude less than zero. If a constraint is placed on the vehicle that requires the altitude to be greater than zero, the problem will never converge because there is no variable for OTIS to adjust that will successfully constrain the altitude. To get around this problem, the simulation had to be started from the point where the vehicle is moving fast enough to generate enough lift to offset the weight of the vehicle. This reduces the accuracy of the simulation because it requires the user to either guess how much fuel is burned during the roll or to use another trajectory program like POST to simulate the roll portion of the flight. The latter solution was used to complete this project.

An additional difficulty discovered in this simulation was the difficulty in converging an air-breathing launch vehicle simulation. In order to get the problem to converge, the trajectory had to be built up step by step. That is, after setting up the input file with the desired sequence of phases, the NP flag in OTIS has to be set to 1 and the name of the first phase needs to be set in the CPI variable array[††]. This tells OTIS to only run the simulation to convergence on the first phase. After the first phase is converged, the input file needs to be edited so that the NP flag is set to 2 and the names of the first two phases need to be entered in the CPI variable array. Additionally, the restart file that resulted from the convergence of the first phase needs to be modified to include the addition of the second phase. Editing the restart file entails changing the total number of phases on the first line of the restart file and adding an additional phase at the end of the file. This new phase is only a "dummy" phase that consists of two nodes. Each

[††] The NP flag in the OTIS input file tells the program how many phases are included in the simulation. The CPI array tells OTIS the names of the phases that are to be run.

of these two nodes has the same set of state variables as the final node from the previous phase. This entire process is repeated until the desired number of phases has been added and the entire simulation converged. More details about the format of the OTIS restart file can be found in Chapter 3.4 of Reference 3. Additionally, it should be noted that the method described here is not the only way to get an air-breathing simulation to converge. It simply represents the procedure that worked best for the author during the comparison study.

The need to edit the OTIS restart file before changing the number of phases in the input file is a complication that is not found in POST. After changing the number of phases in a POST simulation, one only needs to update the initial guesses for the independent variables and does not have to worry about editing the restart file.

One similarity between the two codes is that roughly the same amount of work is required of the user to get the solution to converge. Also, both cases having an input file for a vehicle similar to the one being modeled makes creating a valid input file much easier than when starting from scratch. This point cannot be overemphasized. The ability to set the correct flags as well as format the input files correctly is a skill easily forgotten between projects. Having a file that serves as a template helps ensure the file is set up correctly. However, attempting to use a restart file from a similar OTIS simulation is not as easy as starting a POST simulation with the initial guesses from a previous simulation. As a result, it is necessary to start a new vehicle simulation in OTIS from scratch and build up the restart file phase by phase as described above. Since POST does not rely on a restart file, this type of problem does not apply.

5.0 Conclusions

Based on the results presented above, it is possible to draw three main conclusions:

1. Numerically, there is little difference between the optimized result calculated by OTIS and the optimized result calculated by POST.
2. The main difference between the two programs is in the rules and techniques that the user needs to learn in order to run successful simulations.
3. Use of both programs together helps guarantee that the “true” optimum answer is found.

The first conclusion is supported by the data presented in Tables 1 and 3. Specifically, the fact that the two programs calculated a MR that differs by 0.16% for ACRE-92 and 0.21% for *Hyperion* leads one to conclude that there is no statistical basis for saying that one code calculates “better” answers than the other. Additionally, the high degree of similarity between the POST and OTIS calculated values shown in Figures 3-8 and 13-18 indicate that the two programs compute comparable results. These data support the hypothesis formed prior to beginning the research effort. It also is a logical result considering that the laws of physics are the same whether or not an engineer is using the direct shooting method or the collocation method.

The second conclusion arises from the fact that the results predicted by the two programs are so similar. All other things being equal, it makes sense for an engineer to choose the program that is easiest to use. In most cases, the code that is learned first is the one that feels most comfortable and therefore becomes the program of choice. This is because both programs are complex and require a substantial investment of time to learn how to use well. By the time a user feels comfortable using the program, he or she will have picked up on the idiosyncrasies of the program and will be able to successfully interpret the error messages, help nudge the optimizer towards convergence, etc. After becoming so familiar with one way of doing things, it becomes much more difficult to switch gears and start doing things differently. The end result is that most engineers develop loyalty to the code that they learned first.

On first glance, the third conclusion may seem somewhat contradictory. After all, how can one optimum be better than another? Some optimization problems are sensitive to the initial guesses. That is to say that the optimum answer that results for a particular problem is dependent upon the initial conditions used to set up the problem. As a result, the optimizer will report that it has found the global optimum when it has in fact only found a local optimum. This situation arose during the ACRE-92 and *Hyperion* simulations described earlier. The solution to the problem was to adjust the initial guess values for both the OTIS and POST simulations until a repeatable final optimum was achieved. As a result, the trajectory analyst is able to find a better answer by using both of the codes together and taking clues from each of them to improve the results.

6.0 Future Work

In addition to the conclusions discussed in Section 5.0, another intent of the project was to compare the relative run times of the two different programs. To do this, the goal was to record the number of central processing unit (CPU) seconds required for POST and OTIS to solve the same problems. Unfortunately, the two programs had to be run on different types of computers because of difficulties encountered in installing OTIS on the Georgia Tech computers. As a result, only a qualitative assessment of run time can be made. Based on the experience of dealing with both codes, it seems that there is not much difference in run time between the two programs. This assessment is made based on the fact that the two codes successfully solved the ACRE-92 problem with relative speed and that the majority of time required to solve the *Hyperion* problem was spent in the user-setup stages. In order to make a more accurate assessment of run time, future work should involve getting OTIS to run on the same computer as POST and then rerunning the simulations.

Acknowledgements

I would like to thank all of the people who helped with this research. In particular, John Riehl at NASA-GRC was extremely helpful in getting the OTIS source code installed and running on the SSDL computers at Georgia Tech. Mr. Riehl also provided help with the use of OTIS as well as advice on how to fix specific problems encountered in the use code. Additionally, Mr. Riehl helped set up a computer account for me on the NASA-GRC computer network so that I could use OTIS before I had a working executable installed at Georgia Tech. Without this help, I would not have been able to finish the project on time.

Steve Paris of the Boeing Corp. also provided insight into how to model both rocket and air-breathing vehicles in OTIS and provided significant help in troubleshooting errors with the program as well as several helpful references on OTIS and the collocation method in general. Both Mr. Riehl and Mr. Paris were very generous with their time and enthusiastically supported the work via email and telephone correspondence.

Additional thanks are due Dr. John Olds of the Georgia Institute of Technology for the overall guidance he provided during the research effort. Furthermore, Dr. Olds lent his expertise in the use of POST and helped answer several questions that arose during the simulations. Additionally, Dr. Olds invested a significant amount of time dealing with installation problems encountered during the installation and compilation of the OTIS source code so that I could focus on completing the actual research work.

Finally, I wish to thank my wife, Kelly, for supporting me in this endeavor as she has in everything else that I do. In addition to offering encouragement and support, she served as my proofreader and sounding board while doing an admirable job of enduring my complaints as I struggled to learn how to use the trajectory codes.

References

1. Hargraves, C.R., Paris, S.W., and Vlases, W.G., "OTIS Past, Present, and Future," AIAA 92-4530. 1992.

2. Brauer, G.L., Cornick, D.E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories." NASA CR-2770, Feb. 1977.

3. Paris, S.W., and Hargraves, S.R., "Optimal Trajectories by Implicit Simulation (OTIS 3.10) Volume II – User's Manual." pp. 130-131. Boeing Defense and Space Group. Seattle, WA. 1996.

4. Haney, J.W., "Highly Reusable Space Transportation (HRST) Advanced Concepts Study," Rockwell Aerospace Corporation, project report #43120, November 30, 1995.

5. Olds, J., Bradford, J., Charania, A., Ledsinger, L., McCormick, D., Sorensen, K., "*Hyperion*: An SSTO Vision Vehicle Concept Utilizing Rocket-Based Combined Cycle Propulsion," AIAA 99-4944, 9th International Space Planes and Hypersonic Systems and Technologies Conference, Norfolk, VA, November 1-5, 1999.

6. http://www.ssdl.gatech.edu/main/gallery/vehicles/hyperion/hyperion3.html

Appendix A: POST Input for ACRE-92

```
l$search
c*****************************************************************
c   problem
c       maximize weight
c       subject to:
c                 gcrad  -   2.12294674E+07  =   0
c                 veli   -   25842          =   0
c                 gammai -   0              =   0
c*******************************************************************
c
c  Sample RLV trajectory to a 50 x 220 nmi SSA transfer orbit
c
c  Used for AE6351C design class at Georgia Tech
c  John R. Olds, Feb. 1998
c
c  Changed to 50 nmi x 100 nmi x 28.5 inc on 1/11/01, A. Crocker
c  Modified to represent the ACRE-92 vehicle by Doug Nelson in March 2001
c
 maxitr     =      -1,      / max number of iterations
c maxitr    =      15,
 srchm      =       5,      / optimization mode
 opt        =     1.0,      / maximize
 optvar     = 6hweight,     / final weight
 optph      =    1000,      / optimization phase
 ipro       =      -1,      / only print final trajectory
c
c  Independent Variables
 nindv      =       5,      / no. of controls (independent variables)
 indvr      = 3hazl, 6hpitpc2, 6hpitpc2, 6hpitpc2 , 6hpitpc2,    / independent variables
 indph      = 1,10,20,30,40,    / phases where controls are initated
c
c  Dependent Variables (constraints)
 ndepv      =       3,      / no. of dependent variables (constraints)
 indxd      =       2,3,4,
 depvr      = 5hgdalt, 6hgammai, 3hinc,5hgcrad,   / names of dependent variables
 depval     = 303805.0,0.0, 28.5, 2.12294674E+07,  / target values
 deptl      =  10000.0,.001, .1, 1,    / targeting criteria(allowable errors)
 depph      =  1000,1000,1000,1000,    / targeting phase
c
c  initial guesses for independent variables (last set only)
c
 u=  4.243399409584E+01, -1.215272810754E+00, -1.550217670977E-01, -2.611541158985E-01, -1.273549568738E-01,
 u=  4.243397642161E+01, -1.213811261974E+00, -1.550502569177E-01, -2.620719261863E-01, -1.269318752077E-01,
 u=  4.242795829879E+01, -1.234896416820E+00, -1.333822111109E-01, -2.465411656693E-01, -1.375122726866E-01,
 u=  14.5E+00, -1.20E+00, -1.33E-01, -2.50E-01, -1.40E-01,
 u=  8.820541657940E+01, -1.249490599044E+00, -1.221108194734E-01, -2.508892559407E-01, -1.430563768332E-01,
 u=  8.604275735847E+01, -1.263170686087E+00, -1.043129155945E-01, -2.481305759232E-01, -1.532149678822E-01,
 u=  8.625459728397E+01, -1.281678815143E+00, -7.937265325395E-02, -2.565533973527E-01, -1.495133607287E-01,
 $
c
c  Begin event listing
c
c
l$gendat
 title      = 0h*RLV SSTO to Space Station Transfer Orbit*,
 event      =       1,
c
c  problem setup flags
c
 npc(1)     =       3,  / calculate conics
 npc(2)     =       1,  / runge kutta integration
 npc(3)     =       4,  / input earth relative components(velr,gammar,azvelr)
 npc(4)     =       2,  / input sph coordinates(lat,long,alt)
 npc(7)     =       1,  / limit acceleration to asmax g's
```

```
  asmax    =       3,
 npc(8)    =      2,  / use lift and drag coefs
 npc(9)    =      1,  / rocket propulsion
  iwdf(1)  =      2,  / use Isp vac
  neng     =      1,  / number of engines (multiply Tvac)
 npc(16)   =      0,  / non-spherical earth
 npc(25)   =      3,  / calculate velocity losses
c
c  guidance setup
c
 iguid(1)  =      1,0,  / inertial pitch angles
 iguid(4)  =      1,   / constant pitch term in polynomial
 pitpc(1)  =      0,   / starting inertial pitch angle
c
 maxtim     =   1000.0,   / abort if simulation exceeds 1000 seconds
 fesn       =    1000,    / final event sequence no.
 dt         =     1.0,    / integration step size
 pinc    =     50.0,
c pinc      =     20.0,    / print frequency
 prnc       =       0,    / make file for plotting
 prnca      =       1,
 time       =     0.0,    / initial time
 prnt(97)   =  6hnetisp, 'altito',    /  additional print block variables
c
c   Launch from KSC
c
 gdlat      =    28.5,   / initial latitude
 long       =   279.4,   / initial longitude
 azl        =     0.0,   / azimuth of launch centered inertial frame (set above)
c
c        *************
c
c   HRST Advanced LOX / LH2 Engine from Rocketdyne (Levack) 12 / 96
c
 ispv       =   451.43,  / vac isp
 wgtsg      =    2.1e6,  / initial weight
 sref       =   4000.0,  / aero ref area
c
c        *************
c
 $
l$tblmlt
 $
c*include ' / home / asdl2 / olds / post / wb001.aero'
c*include ' / usr / people / olds / classes / design_98 / wb001.aero'
*include ' / home / asdl2 / dnelson / AE8900 / ACRE92 / POST / wb001.aero'
l$tab
c
c   here, tvc1 is the total thrust for all engines
c
 table      = 6htvc1t ,0,2.9006e6,
 $
l$tab
c
c   total exit area approx = (Tvac / 2.423e6)*150.25 ft2
c
 table      = 6hae1t  ,0,179.87,
 endphs     = 1,
 $
l$gendat
c   change to pitch polynomial using linear term
 event      = 10,
 critr      = 5hgdalt,
 value      = 400,
 iguid(4)   = 0,
 endphs     = 1,
 pinc    =50.,
 $
l$gendat
 event      = 20,
 critr      = 6htime  ,
 value      = 50,
 endphs     = 1,
 $
l$gendat
 event      = 30,
 critr      = 6htime  ,
 value      = 100,
 endphs     = 1,
 $
l$gendat
 event      = 40,
 critr      = 6htime  ,
 value      = 200,
 endphs     = 1,
 $
l$gendat
 event      = 1000,
 critr      = 6hveli  ,
 value      = 25842.,
 endphs     = 1,
 endprb     = 1,
 endjob     = 1,
 $
```

Appendix B: OTIS Input for ACRE-92

```
  ACRE92 SSTO Simulation
  Uses MODE 4
  MARCH 2001
 $OTISIN
*
********** GENERAL INPUTS
*
  NIT=300,LM4=T,NP=3,LM1=F,
  NUWZ=2,NURZ=3,
*
  RESTART=F,
  RESTART=T,
*
  INP=2,
  AUTOSCALE = T,
  AUTO4_SV = F,
*
  REDIST = T,
  NZOPT = 3,
*
*
*
  CHOTFG=F,
  EFLAG=F,
  DEBUG=F,
  ORBFLG=T,
  APCOPY=F,
  RANGFG=F,
  CARTFG=F,
*  ZHARFG=F,
  TDELFG=F,
  ATMSFG=F,
  AUX2FG=F,
  LZOUT=T,
  GGPIFG=T,GGPXFG=T,
*
********** LINKING CONDITIONS
C  CPI='ONE', 'THREE','FOUR',
*
********** OBJECTIVE FUNCTION
*
   NOL=1,
***NPOL=15,
   WOL=1,SOB=-1.,
C  CPOL='FOUR',
C  CMOL='WEIGHT',
*
*********** VEHICLE MODEL
*
   NST=1,
   SREF=4000.0,
   NENG=1,
C  CLDSN='RLVCL',CDDSN='RLVCD',
C  THDSN='RLVTHR',
C  MDDSN='RLVISP',
C  ARDSN='RLVAREA',
*   DVISP=451.43,
*
*********** OUTPUT
*
*     MSGLVL=5,
      NPPS=10,
C     CXPP='TIME','TIME','TIME','TIME','TIME','TIME','TIME','TIME','TIME', 'TIME',
C     CYPP='ALT','Q','THRUST','DRAG','LIFT','WEIGHT','ALPHAD','VEL','GAMD', 'ACCA',
     NPTO=24,
C    CPTO='TIME','TPHASE','ALT','MACH','VEL','ALPHAD','GAMD','WEIGHT',
C         'AZMD','THRUST','LATD','LOND','ACCA','RADI','Q','VELI'
C         'CL','LIFT','CD','DRAG','HA','HP', 'INCD', 'XPAR(1)','ECC', 'SMA',
      NLNPLT=22,
      NHBPLT=4,
 $END

 $PHASIN
****************** PHASE 1 DEFINITION *****************
C  PTITLE='Vertical Rise - Core plus first 6 solids'
*
  NNPP=12,SEGL=20*1.,IBODY=399,IATMOS=1,ISTAGE=1,
C PTYPE='C',PNAME='ONE',
*
*********** EARTH MODEL
*
*  EFLAT=0.,
*
*********** INITIAL CONDITIONS
*
  IENTYPE=2,2,IEQTYPE=2,2,ITG=1,
  ZI=        0.,  90,    90.,   1.e-6, -81. , 28.5, 2.1e6, 0.,
  ZI(1)= 1.E-6,
  ZF=    72, 90, 89.6, 348, 279.3, 0.2850E+02, 2.0357E+06,
  TI= 0.,
  TP=10.,
*
********* BOUNDARY CONDITIONS
*
   NEQI=6,IEQI=1,3,4,5,6,7,
*
************ BOUNDS AND SCALE FACTORS
*
  ZMAX= 1343.,  90.,    90., 85., -81. , 28.5, 2.1e6, 3.5e+04,
*  ZMIN= 1.00E+00,-3.14E+00,-1.57E+00, 0.00E+00,-3.14E+00,-1.57E+00,1.19E+04,-2.36E+04,
  SDEF=100000.,  10., 10., 1000000., 10., 10., 1000., 1000.,
*  NQV=1,
*C CQV='WD(1)',
  ST=  10.,
  TUB= 10.,
  TLB= 1.,
  TIUB= 0.,
  TILB= 0.,
*
*    Special phase - map out gamma dot
*
  MAPFPC(2)=0,
  MAPFPC(3)=0,
  VPCDOT(3)=-.002,
*
********** CONTROL VARIABLE
*
  ICONST=2,
  NCT=0,
  NCVF=2,ICONST=2,
C CCVF='ALPHA','XPAR(1)',
  FCV=0., 1.0,
*
  DTPRINT=50.,
 $END
*
****************** PHASE 3 DEFINITION *****************
 $PHASIN
```

```
C  PTITLE='Third stage'
*
  NNPP=20,SEGL=20*1.,IBODY=399,IATMOS=1,ISTAGE=1,
C PTYPE='C',PNAME='THREE',
*
*
*********** INITIAL CONDITIONS
*
  IENTYPE=2,2,IEQTYPE=2,2,ITG=1,
  ZI=      72,  0.9004E+02,  0.4255E+02,  348, -0.8091E+02,  0.2850E+02,  0.1699E+07,
  ZF=      2530,  0.9004E+02,  0.4255E+02,  0.5878E+05, -0.8091E+02,  0.2850E+02,  0.1699E+07,
  TI=60.,
  TP=47.4,
*
************ BOUNDS AND SCALE FACTORS
*
  ZMAX= 25842.,  90.,    88.,  10000., -81. , 28.3, 2.79e5, 1.28E+05, 1.000000, 1.000000,
*  ZMAX= 5.75E+04, 3.14E+00, 1.57E+00, 4.67E+06, 3.14E+00, 1.57E+00,3.87E+03, 1.28E+05, 1.000000, 1.000000,
*  ZMIN= 1.00E+00,-3.14E+00,-1.57E+00, 0.00E+00,-3.14E+00,-1.57E+00,1.00E+00,-6.70E+04,-1.000000,-1.000000,
  SDEF=100000.,  10., 10., 1000000., 10., 10., 1000., 1000., 1., 1.,
*  NQV=1,
*C CQV='WD(1)',
  ST=  100.,
  TUB= 550.,
  TLB= 1.,
  TIUB=600.,
  TILB=  5.,
*
*
********** CONTROL VARIABLE
*
  ICONST=2,
  NCT=1,
C CCV='ALPHA',
  CUL= 1.0,
  CLL=-0.349,
*  CLLD=-0.1745329,
*  CULD=0.1745329,
*
  NCVF=1,ICONST=2,
C CCVF='XPAR(1)',
  FCV= 1.0,
*
  NPC=3,
C CPC= 'ACCA','ALTDOT','GAMDOT',
  PLB= 1.2, 0., -0.01745329,
  PUB=  3.02, 100000.,0.01745329,
  SCON= 3., 100000., 0.01745329,
*
  DTPRINT=50.,
  NNFC=1,
C CNFC=     'ACCA',
  YFCUB=  3.0,
  YFCLB=  3.0,
  SNFC=   3.,
 $END
****************** PHASE 4 DEFINITION *****************
 $PHASIN
C  PTITLE='Fourth stage'
*
  NNPP=20,SEGL=20*1.,IBODY=399,IATMOS=1,ISTAGE=1,
C PTYPE='C',PNAME='FOUR',
C CC = '1_THREE'
C CC = '1_THREE'
```

```
*
*C CQ = '1_Fairing',
*
*********** INITIAL CONDITIONS
*
  IENTYPE=2,2,IEQTYPE=2,2,ITG=1,
  ZI=    2530,  0.9004E+02,  0.4255E+02,  0.5878E+05, -0.8091E+02,  0.2850E+02,  0.1699E+07,
  ZF=    24482,  0.9620E+02,  0.1706E-01,  0.3039E+06, -0.6792E+02,  0.2787E+02,  0.2572E+06,
  TI=110.,
  TP=339.7,
*
************ BOUNDS AND SCALE FACTORS
*
  ZMAX= 25842.,  90.,    88.,  10000., -81. , 28.3, 2.79e5, 1.28E+05, 1.000000, 1.000000,
*  ZMAX= 5.75E+04, 3.14E+00, 1.57E+00, 4.67E+06, 3.14E+00, 1.57E+00,3.87E+03, 1.28E+05, 1.000000, 1.000000,
*  ZMIN= 1.00E+00,-3.14E+00,-1.57E+00, 0.00E+00,-3.14E+00,-1.57E+00,1.00E+00,-6.70E+04,-1.000000,-1.000000,
  SDEF=100000.,  10., 10., 1000000., 10., 10., 1000., 1000., 1., 1.,
*  NQV=1,
*C CQV='WD(1)',
  ST=  100.,
  TUB= 550.,
  TLB= 5.,
  TIUB=600.,
  TILB=  5.,
*
*
********** CONTROL VARIABLE
*
  ICONST=2,
*
  NCT=2,
C CCV='ALPHA', 'XPAR(1)'
  CUL= 1.0, 1.0,
  CLL=-.349, 0.2,
*  CLLD=-0.1745329,
*  CULD=0.1745329,
*
  NPC=1,
C CPC= 'ACCA',
  PLB= 3.0,
  PUB=  3.01,
  SCON= 3.01,
*
  DTPRINT=50.,
  NNFC=4,
C CNFC=   'VELI',   'GAMD', 'INCD', 'RADI','HA', 'HP',
  YFCUB=  25845.,     .0001,    28.53, 2.12294674E+07,110.,60.,
  YFCLB=   25842.,    0.,    28.5,   2.12294674E+07, 100., 50.,
  SNFC=   25845.,   100.,    1000.,  2.12294674E+07, 110, 50.
 $END
 $SPC
Begin Mode4Tst

*  Major iterations are set via OTIS' NIT parameter
*  Minor iterations            5000
*  (Minor Iter Default=max{1000,5m}; where m=tot# of constr)
*  Iterations limit            xxxxxx
*  (Total Minor Iter Default=max{10000,20m}; where m=tot# of constr)

   Derivative line search          * The default

   Major Print level          000001
*                        (JFDXbs)
   Minor print level              0
   Major Optimality  Tolerance     1.0e-6
   Major Feasibility Tolerance     1.0e-6
       Feasibility Tolerance     1.0e-6
   Minor Optimality  Tolerance     1.0e-8
   Minor Feasibility Tolerance     1.0e-8
   Crash option                   0

   Print frequency               1
   Summary frequency              1
   Solution                    yes

   Verify Level                 -1

   Check frequency               1
   Function precision            1.0E-8
   Linesearch tolerance           0.10
   Major step limit              0.50

End Mode4Tst
   $END
```

Appendix C: POST Input for *Hyperion*

```
l$search
c
c  hyperion_release.inp
c  SSTO vehicle takes off horizotally, then accelerates onto a dynamic
c  pressure boundary until a maximum airbreathing Mach number. The
c  booster accelerates to the staging point under rocket mode power.
c  The target orbit is 50 x 100 nmi x 28.5 inclination (LEO).
c
c  The ESJ RBCC engine operates in several modes on the way to orbit
c  including ejector, ramjet, scramjet, and pure rocket.
c  The dynamic pressure during the airbreathing segment can be varied
c  through different flight regimes and engine operating modes.
c
c  created by Dr. John R. Olds, Georgia Tech, January 2001.
c  uses new postq5 version with variable q tables created by Doug
c  Nelson, Georgia Tech, January 2001
c
c  modified by Doug Nelson to intercept q boundary condition after
c  takeoff rotation and climb out, Georgia Tech, March 2001.
c
c  modified by Doug Nelson to allow optimizer to choose when the vehicle gets
c  on the q-boundary and also to allow it to choose the low mach number q
c  values that optimize final weight, Georgia Tech, April 2001.
c
 maxitr  = -1,
c maxitr  = 15,
 ipro    = -1,            / print first and last trajectories
c
 opt     = 1,             / maximize (i.e. use optimizer)
c opt    = 0,             / target only
 srchm   = 5,             / projected gradient optimizer
 optvar  = 6hweight,      / final weight
 optph   = 2000,
 wopt    = 200000,
 pctcc   =.5,

c
c  ** setup independent variables **
c
 nindv   = 14,
 tabl(1) = 4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,4hpitt,
 tabl(11)= 6hgenv1t,6hgenv1t,6hgenv1t,6hgenv1t,
 tably(1)= 2,3,4,5,6,7,8,9,10,11,2,3,4,

 indvr   = 5htabl1,5htabl2,5htabl3,5htabl4,5htabl5,5htabl6,5htabl7,5htabl8,5htabl9,6htabl10,
 indvr(11)=6htabl11,6htabl12,6htabl13,5hcritr,
 indph   = 10*1000,4*100,
 pert(1) = 10*1.e-6,4*1.e-2,
c
 ndepv   = 4,
 indxd   = 1,2,4,5,
c
c  the targets are a 50 x 100 nmi circular orbit, max q < 2000 psf
c  (where xmax1 is the monitor variable for max q),
c  fazb < 1.75 * gross weight (where xmin5 monitors fazb in pull up)
c  orbital inclination = 28.5 (due east orbit), and max static
c  engine pressure in ramjet mode (< 250 psi).
c
 depvr   = 5hxmax1,   5hxmin5,  6hgammai,  5hgcrad,        6hgdaltp,6hgdalta,  5hxmax6,
 depph   =    2000,      2000,      2000,  2000,           2000,    2000,      2000,
 depval  =    2050,  -1968750,         0,  2.12294674E+07, 50.,     100.,       250,
 idepvr  =       1,        -1,         0,  0,              0,       0,          1,
 deptl   =      10,     10000,        .1,  1000,           1,       1,          5,
c
```

```
c   guesses for independent variables (u's)

 u=  1.123985568414e+01,   1.767650305295e+01,  1.636292695490e+01,  1.476666801845e+01,  1.354729146167e+01,
  1.073911672022e+01,  7.477952074683e+00,  3.548471741921e+00,  1.571735106832e+00, -2.76846982574e+00,810,810,810,810,
 u=   1.063951360736E+01,  1.860394717156E+01,  1.678905449937E+01,  1.495920954332E+01,  1.337254857653E+01,
  1.043523997018E+01,  7.225198893314E+00,  3.463833275771E+00,  1.547027849858E+00, -2.794229193539E+00,
  7.100000000000E+02,  9.145247548254E+02,  1.011469990817E+03,  7.369948923619E+02,
 u=   1.119435549038E+01,  1.793765056436E+01,  1.641902925407E+01,  1.657230496396E+01,  1.390860867248E+01,
  1.047537226204E+01,  7.073537271665E+00,  3.389623262516E+00,  1.502411782310E+00, -2.839353792889E+00,
  7.099945836064E+02,  1.328193144549E+03,  1.059893921383E+03,  6.849088126348E+02,
c
 $
c
c   ** begin simulation **
c
c
c  begin events with horizontal take off roll
c
l$gendat
 event  = 1,0,
 title  = 0h*Hyperion Release Version -- LEO Mission*,
c
 fesn   = 3000,      / final event sequence number
 maxtim = 6000,
c
 monx(1)= 4hdynp,    / monitor dynamic pressure
 mony(1)= 4hmach,
 monx(2)= 4hasmg,    / monitor max g's
 mony(2)= 4hmach,
 monx(3)= 5halpha,   / monitor alpha
 mony(3)= 4hmach,
 monx(4)= 4hfazb,    / monitor wing normal force through end of airbreathing modes
 mony(4)= 4hmach,
 monx(5)= 4hfazb,    / monitor second wing normal force from rocket pullup to MECO
 mony(5)= 4hmach,
 monx(6)= 6hgenv13,  / monitor engine pressure in ramjet mode
 mony(6)= 4hmach,
c
c  print additional calculated values (other than standard set)
 prnt(97)= 6hxmax1 ,6hyxmx1 ,6hxmax2 ,6hyxmx2 ,6hxmax3 ,6hyxmx3 ,5hxmin4,5hyxmn4,
        5hxmin5,5hyxmn5,5hxmax6,5hyxmx6,6hnetisp,6hdynpdt,6hheat2d,6htimrf1,6htimrf2,
        5hgenv3,5hgenv4,5hgint6,4hthr1,4hthr3,4hthr4,4hthr6,4hisp1,4hisp3,4hisp4,4hisp6,
        5hgenv1,6hgenv13,6hgenv14,
c
c  set profile print, integration time step, and print block options
 prnc  = 3,
 prnca = 3,        / print to ASCII file also
 dt    = 1.0,
 pinc  = 50.,
c
c    ***  set appropriate npc flags  ***
c
 npc(3)  = 4,        / use earth relative initial velocity (G frame)
   velr  = 1,gammar=0,azvelr=90.0,
 npc(4)  = 2,        / use spherical initial position coordinates
   gdlat = 28.5,long=280,gdalt=0,
 npc(5)  = 5,        / use 1976 standard atmosphere
 npc(7)  = 1,        / limit acceleration to 3 g's
   asmax = 4,
 npc(8)  = 2,        / use cd and cl as aerodynamic coefficients
 npc(9)  = 1,        / use rocket engine equations with vacuum thrust and Isp
 npc(12) = 2,        / calculate downrange and crossrange based on inertial great circles
 npc(14) = 2,        / set hold down option for HTO
 npc(15) = 1,        / calculate heating using Chapman's equation
 npc(16) = 0,       / spherical earth
 npc(22) = 3,       / calculate the throttling parameter, eta, by a table lookup
 npc(24) = 1,       / calculate user defined integrals
 npc(25) = 2,       / calculate and print velocity losses
 npc(30) = 3,       / use enhanced weight model
c
c  set up 4 propulsion systems
c
  ispv   = 0,    0,     0,     0,     0,    462,
  iengmf = 1,    0,     3,     3,     0,     0,    / which engines are on and what type
  ienga  = 1,   -1,    -1,    -1,    -1,    -1,    / throttle which engines
  iwdf   = 3,    3,     3,     3,     3,     2,    / flowrate calculation (isp table or const)
  iwpf   = 1,    1,     1,     1,     1,     1,    / include engines in flowrate calculations
  nengl  = 1,
  nengh  = 6,
  nstpl  = 1,
  nstph  = 1,
  menstp = 1,1,1,1,1,1,        / map each engine to a specific step
  mentnk = 1,1,1,1,1,1,        / map each engine to a specific tank
c
c  set up initial weight
c
c  wstpd is the initial gross weight.
 wstpd(1) = 1125000,
 istepf  = 1,
c
c  reference area for aerodynamic coefficients is sref
 sref=8068.4,
c
c  Start integration of LOX prop flow for subsequent calculation of mixture ratio
 gderv(6)=6hgenv6 ,
 $
c
c
c  include engine and aerodynamic tables calculated elsewhere
c  note that rbcc.dat table includes required tblmlt line
c
cl$tblmlt $
*include '/home/asdl2/dnelson/AE8900/HYPERION/POST/temp/rbcc_hyperion_release.dat'
*include '/home/asdl2/dnelson/AE8900/HYPERION/POST/temp/hyperion_release.aero'
c
c  table to tie lox flow to total throttle for ejector phase
cl$tblmlt $
l$tab table=6hgenv6t,1,4hetal,2,3*1,
 0,  0,
 10, 10,
 $
c
c  ** set appropriate throttles for ejector and ramjet cutovers here **
c
c  begin additional control for ejector throttle
l$tab table=4hetat,1,6hmach  ,4,1,1,1,
 0,    1,
 2.9,  1,
 3.1,  0,
 10,   0,
 $
c
c  table of throttle control for ramjet thottle (engine 3)
l$tab table=6hgenv3t,1,4hmach,6,1,1,1,
 0.0,  0,
 2.9,  0,
 3.1,  1,
 5.9,  1,
 6.1,   0,
 20.0,  0,
 $
c  table of throttle control for scramjet throttle (engine 4)
l$tab table=6hgenv4t,1,4hmach,6,1,1,1,
 0.0,   0,
 5.9,   0,
 6.1,   1,
 10.0,  1,
 11.0,  0,
 20.0,  0,
 endphs = 1,
 $
c
c  horizontal take-off roll to about 230 knots (400 fps).
c  then begin rotation with alpha control
c
l$gendat
 event    = 20,0,critr=4hvela,value=400,
 iguid    = 0,0,3,   /  use 'desired alpha' control starting at zero degrees
 dalpha   = 25,
 desn     = 100,100,100,  /  target a high enough alpha for event 200
 dtimr(1) = 1,
 endphs   = 1,
 $
c
c**********************
c  release horizontal hold down when wing lift reaches prescribed value
c  note: change lift value manually as gross weight changes
c**********************
c
l$gendat
 event    = 21,0,tol=1,critr=4hlift,value= 1012500,
 npc(14)  = 0,
 dalpha   = 6,     /  bring alpha down before next event
 endphs   = 1,
 $
c
c
c  change to special LFC steering to hold dynamic pressure boundary 20 seconds
c  after the takeoff rotation begins
c
l$gendat
 event    = 100,tol=1e-6,critr=4hdynp,value=810,
 dt       = .1,        /  decrease integration time step for constant q phase LFC
 pinc     = 50.,        /  decrease print increment
c
c  establish required constants and setting for LFC and calspe
c
 iguid(1)  = 0,1,
 iguid(6)  = 6,1,1,      /  use special LFC steering subroutine written at GT
 kdg(1)    = 0.005,      /  displacement gain on q
 krg(1)    = 0.1,       /  rate gain on q dot
 us(1)     = 8,         /  initial guess for alpha
 $
l$tblmlt genv1m = 1,
c
c  table for q boundary control
c  note some values can be changed by the optimizer
 $
l$tab table=6hgenv1t,1,4hmach,12,1,1,1,
 0.25,    820,
 0.50,    810,
 0.69,    810,
 0.95,    850,
 1.25,   1570,
 1.88,   2000,
 2.86,   2000,
 3.90,   1900,
 5.50,   1650,
 6.07,   1470,
 6.35,   2000,
 10.0,   2000,
 endphs   = 1,
 $
c
l$gendat
 event = 200,critr = 4hmach,value = 2.0,
 pinc     = 50.,         / change print increment
 endphs   = 1,
 $
c
l$gendat
 event = 300,critr = 4hmach,value = 3.0,
 kdg(1)   = 0.005,
 krg(1)   = 0.1,
 pinc     = 50.,         / change print increment
 endphs   = 1,
 $
l$gendat
 event = 500,critr = 4hmach,value = 5.0,
 kdg(1)   = 0.005,
 krg(1)   = 0.1,
 pinc     = 50.,         / change print increment
 endphs   = 1,
 $
l$gendat
 event = 600,critr = 4hmach,value = 6.0,
 kdg(1)   = 0.005,
 krg(1)   = 0.1,
 monx(6)  = 6hgenv14,   / set monx(6) to monitor max static pressure for scramjet mode
 pinc     = 50.,         / change print increment
 endphs   = 1,
 $
c
c  leave dynamic pressure boundary as rocket mode transitions in
c
l$gendat
 event = 1000, critr=4hmach,value=10,
 dt     = 1,
 prnc   = 3,
c prnca = 3,
c
 iengmf(1) = 0,0,0,3,0,1,      / turn on rocket engine
 ienga     = -1,-1,-1,-1,-1,1, / let only the rocket engine be controlled by the throttle
 iguid(1)  = 2,0,             / return to relative pitch angle control for rocket mode
 iguid(4)  = 2,               / use table lookup for pitch angles, pitt
 iguid(13) = 2,
 timrf(1)  = 0,               / reset timer1
c
 npc(1)   = 2,               / begin to calculate Keplerian elements
 npc(7)   = 1,               / return to acceleration limit
   asmax = 4.0,
 pinc     = 50.,              / change time between print blocks
 xmax(2)  = 0,                / reset monitor for heat rate
 xmin(5)  = 0,                / reset second wing normal force monitor
 npc(22)  = 3,                / use table lookup for eta calculation
 $
```

```
c  rocket engine operates at a mixture ratio of 7/1.
cl$tblmlt genv6m = 1327.7 $
l$tblmlt $
c
c  track use of oxygen
l$tab table=6hgenv6t,1,4hetal,2,3*1,
 0,0,
 1000000,1000000,
 $
c  ramp up rocket thrust gradually over 10 seconds
l$tab table=4hetat,1,6htimrf1,3,3*1,
 0,  0,
 15, 1,
 100,1,
 $
c  pitch angle table - zero values changed by optimizer.
l$tab table=4hpitt,1,6htimrf1,11,1,1,1,
 0,   1,
 10,  0,
 50,  0,
 80,  0,
 120, 0,
 150, 0,
 180, 0,
 210, 0,
 240, 0,
 270, 0,
 300, 0,
 endphs=1,
 $
c
l$gendat
 event = 1100,mdl = -1,critr = 4hdynp,value = 10.0,
 iengmf(1) = 0,0,0,0,0,1,    / turn off ramjet/scramjet and leave on rocket engine
 npc(22)  = 2,               / return to cubic eta calculation
 etapc    = 1,
 endphs   = 1,
 $
l$gendat
c  MECO conditions are velocity, gammai, and gdalt
c event=2000, mdl = 1,critr=4hveli,value=25838.,
 event=2000, mdl=1,critr=6hgdalta,value=100.,
c
 iengmf(1) = 0,0,0,0,0,0,    / turn all engines off
 iguid(1)  = 0,0,            / begin flying at zero angle of attack for minimum drag
 iguid(3)  = 1,
 alppc(1)  = 0,
 $
l$tblmlt genv6m=0,
 endphs = 1 $
c
l$gendat
c  end of simulation
 event=3000,critr=6htdurp ,value=0,
 endphs=1,
 endprb=1,
 endjob=1,
 $
```

Appendix D: OTIS Input for ***Hyperion***

```
$OTISIN
*
*Hyperion Airbreathing RBCC Launch Vehicle
*
*Modified from ABLV input file created by  W K Sjauw of NASA-GRC, 02-20-01
*by Doug Nelson
*
* ---------- Table 1 - General Inputs ----------
*
  LM1=F,
  LM4=T,
  NP=9,
  NIT= 100,
  RESTART=T,
  AUTOSCALE = T,
*   AUTO4_SV = T,
*   GUESS_SV = T,
*   PROB = 'ABLV-7c',
  INP= 2,
*   TIMEFLG = T,
*
* ---------- Table 2 - Auxillary Calculations ----------
*
  ORBFLG=T
  VMINO=   0.0,
  ZHARFG=T,
  TDELFG=T,
*
*   LUSEDV = T,
*   HAFINAL = 220.0,
*   HPFINAL = 220.0,
*
* ---------- Table 3 - Linkage Variables ----------
*
*C  CPI = 'Ph1', 'Ph1_5',
C  CPI = 'Ph2', 'Ph3', 'Ph4', 'Ph5',
C  CPI(5) = 'Ph6',
C  CPI(6) = 'Ph7','Ph8','Ph9','Ph10',
*
* ---------- Table 4 - Vehicle Model ----------
*
  NST=1,
*
  SREF= 8068.4,
  XPAR(1) = 5.0,
  XPAR(2) = 350.,
*   XPAR(3) = 1.0,
*
C  CLDSN = 'HYPCL'
C  CDDSN = 'HYPCD'
C  CYDSN = 'NOCY'
*
  NENG= 4,
*
C  THDSN(1,1)= 'EJEC_THR',     MDDSN(1,1)= 'EJEC_ISP',
*
C  THDSN(2,1)= 'RAM_THR',      MDDSN(2,1)= 'RAM_ISP',
*
C  THDSN(3,1)= 'SCRM_THR',     MDDSN(3,1)= 'SCRM_ISP',
*
C  THDSN(4,1)= 'ROCKTHR',      MDDSN(4,1)= 'ROCKISP',
*
*
* ---------- Table 5 - Output ----------
```

```
*
  LZOUT=T,
  GGPIFG=T,
*
     NPPS=15,
C    CXPP='TIME','TIME','TIME','TIME','TIME','TIME','TIME','TIME','TIME', 'TIME',
C    CXPP(11) = 'MACH', 'MACH', 'MACH', 'MACH','MACH',
C    CYPP='ALT','Q','THRUST','DRAG','LIFT','WEIGHT','ALPHAD','VEL','GAMD', 'ACCA',
C    CYPP(11) = 'TCOMP(1)','TCOMP(2)','TCOMP(3)','TCOMP(4)', 'PCAL(7)',
      NLNPLT=22,
      NHBPLT=4,
*
  NPTO=51,
CCPTO='TIME'      , 'TPHASE'    , 'WEIGHT'    , 'THRUST'    , 'WDOT'     , 'ACCT'      , 'DRAG'    ,
C    'VEL'       , 'AZMD'      , 'GAMD'      , 'ALT'       , 'LOND'     , 'LATD'       , 'MACH'      , 'LIFT'    ,
C    'VELI'      , 'AZMID'     , 'GAMID'     , 'GAMDOT'     , 'PRES'    , 'Q'         ,
C    'SMA'       , 'ECC'       , 'INCD'      , 'CL'         , 'CD'      , 'PCAL(7)'     ,
C    'ALPHAD'    , 'ACCN'       , 'ACCA'    ,  'HA'        , 'HP'       , 'ALTDOT'    , 'VDOT'      ,
*
* For diagnostics:
*
C    'PCAL(1)'   , 'PCAL(2)'   , 'PCAL(3)'   , 'PCAL(4)'   ,'PCAL(5)'   , 'ISP(4)'    ,'TCOMP(1)'   , 'TCOMP(2)'    ,
C    'TCOMP(3)'  , 'TCOMP(4)'  , 'TCOMP(5)',  'XPAR(1)'   , 'XPAR(2)'   , 'XPAR(3)'   , 'DVATM'      , 'DVG', 'DVI',
*
* ---------- Table 6 - Reference Parameters --------------------
*
* Launch from 28.5 degree latitude, Greenwich meridian
  AZML= 90.0,
  LONL= 0.0,
  LATL= 28.5,
*
* ---------- Table 7 - Objective Function Definition ----------
*
  NOL=1,
  SOB = -1.0,
C  CPOL= 'Ph10',
C  CMOL= 'WEIGHT',
  WOL= 1.0E-5,
*
* ---------- Table 8 - Build It-Yourself Parameters ----------
*
*  Calculator Parameters
  NCALP= 7,
  NCAL      = 7*5,
*20, 20, 20, 20, 20, 20,
*
*   NCAL(11:20)= 10, 10, 10, 25, 10, 10, 10, 25, 10, 10,
*   NCAL(21:30)= 30, 30, 30, 30, 25, 25, 25, 30, 30, 30,
*
CAL1 (TAB('THROT1'))
CAL2 (TAB('THROT2'))
CAL3 (TAB('THROT3'))
CAL4 (TAB('THROT4'))
CAL5 (701020 - ('PRES' * 184.77))
CAL6 (('PCAL(5)' / 701020))
CAL7 (TAB('CHAMPRES'))
*
*
* ---------- Table 10 - Node Distribution ----------
*
 REDIST=T,
 NZOPT=3,
*
* ---------- Table 11 - Numerical Methods ----------
```

```
*
* ---------- Table 12 - I/O Units ----------
*
 $END
 $PHASIN
*********************************************
*                    PHASE 1                  *
*                                             *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME       = 'Ph1',
C  PTITLE      = 'Phase 1',
C  PTYPE       = 'C',
   NNPP=12,
*
*   NQV=1,
*      Tot   DR  Fuel  DR  Oxid
*C  CQV= 'WDOT',
*
   ISTAGE=1,
   SEGL=12*1,
   DTPRINT= 50.0,
*
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 0.0,    TP  = 0.20861E+02,
   TILB= 0.0,    TIUB= 0.0,
   TLB = 1.0,    TUB = 100.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
*   IEQTYPE=2,2  IENTYPE=2,2
*         Vel       Azi       Gamd   Alt       Lond       Latd       Weight
*         Alpha      Dalpha
*
   ZI     = 1.,       90.0,     0.0,   0.0,       280.0,        28.5,       1125000.00,
   ZF     = 400.,  90., 0.0,   0.0,  280.,  28.5,  1.08948612E+06,
   ZF(8)   = 0.08726646,
*
*
   NEQI=6,
   IEQI=  1,3,4,5,6,7,
*
*
   NNFC= 1,
C  CNFC = 'VEL',
   YFCLB=  400,
   YFCUB=  400,
   SNFC =  400E1,
*
* ---------- Table 21 - Control Inputs ----------
*
  NCT=1,  ICONST=2,
C CCV='ALPHA',
  CUL= 1.0,
  CLL=-0.349,
*
  MAPFPC(2) = 0,
  VPCDOT(2) = 0.,
*
```

```
* ---------- Table 22 - Placards ----------
*
  NPC=0,
C  CPC=  'GAMD',  'ALT',
  PLB=   0.0,     0.0,
  PUB=   89.0,    10000,
  SCON=  89.0E2,  10000,
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
*********************************************
*                  PHASE 1.5                *
*                                           *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph1_5',
C  PTITLE       = 'Phase 1.5',
C  PTYPE        = 'C',
  NNPP=22,
*
*   NQV=1,
*      Tot      DR  Fuel   DR  Oxid
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph1',
*
  ISTAGE=1,
  SEGL=22*1,
  DTPRINT= 50.0,
*
* ---------- Table 19 - Time Inputs ----------
*
  TI  = 0.208611543E+02,    TP  = 0.9651747101E+01,
  TILB= 1.0,                TIUB= 100.0,
  TLB = 1.0,                TUB = 200.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
*   IEQTYPE=2,2  IENTYPE=2,2
*         Vel       Azi       Gamd   Alt       Lond       Latd       Weight
*         Alpha      Dalpha
*
  ZI     = 400.,  90., 0.0,   0.0,  280.,  28.5,  1.08948612E+06,
         0.08726646,
  ZF     = 548.816,  90.89,  0,  0.,  280,  28.158135,  1.07551743E+06,
         0.18325957,,
*
*
  NNFC= 1,
C  CNFC = 'LIFT',  'ALPHAD'
  YFCLB=  1012400, 5.0,
  YFCUB=  1012600, 11.0,
  SNFC =  1012600E1, 110.,
*
```

```
* ---------- Table 21 - Control Inputs ----------
*
  NCT=1,
C CCV='ALPHA',
  CUL= 1.0,
  CLL=-0.349,
*  CLLD=0.022112,
*  CULD=0.022112,
*
  MAPFPC(2) = 0,
  VPCDOT(2) = 0.,
*
* ---------- Table 22 - Placards ----------
*
   NPC=0,
C  CPC=  'ALPHAD',  'GAMD',
   PLB=  -3.0,      0.0 ,
   PUB=   15.0,     89.0,
   SCON=  10.0E2,   89.0E2,
*
* ---------- Table 24 - Atmospheric Model ----------
*
   IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
*
* ---------- Table 32 - Addl Trim Input ------------
*
*
 $END
 $PHASIN
*********************************************
*                    PHASE 2                *
*                                           *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME       = 'Ph2',
C  PTITLE      = 'Phase 2',
C  PTYPE       = 'C',
*C  CC= '1_Ph1_5',
*C  CR= '1_Ph1_5',
   NNPP=22,
*
*   NQV=1,
*      Tot     DR  Fuel   DR  Oxid
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph1',
*
   ISTAGE=1,
   SEGL=22*1,
   DTPRINT= 50.0,
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 0.28991E+02,   TP  = 0.9776E+01,
   TILB= 0.28991E+02,   TIUB=0.28991E+02,
   TLB = 1.0,           TUB = 200.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
   NEQI=7,
  IEQI= 1,3,4,5,6,7,8,
*         Vel       Azi      Gamd   Alt       Lond       Latd       Weight
*         Alpha      Dalpha
*
  ZI     = 548.816,  90.89,  0,  0.,  280,  28.158135,  1.07551743E+06,
*          0.18325957,
  ZF     = 711.57,  90.98,  .96729, 99.797, 280.4,  28.499,  1.05860489E+06,
          0.0872265
*
  NNFC= 1,
C  CNFC = 'Q',
  YFCLB=   0.,
  YFCUB=  600.,
  SNFC =  610E1,
*
* ---------- Table 21 - Control Inputs ----------
*
  NCT= 1,
C  CCV=  'ALPHA',
  CUL=   1.0,
  CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
  NPC=6,
C  CPC=  'Q',   'ALPHAD',  'GAMD', 'ALT',   'AZMD','FNORM'
  PLB=   0.,    -3.0,       0.0 ,   0.,      0.0,  -1968750
  PUB=   600.,  12.0,      89.0,   100000., 180,    1968750
  SCON=  600.,  10.0E2,    89.0E2, 100000., 180,    1968750
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Central Body --------------------------
* ---------- Table 17 - Central Body ---------
*
  EFLAT= 0.3352824D-2,
  GCON = 32.17405,
  MUCB = 1.4076466E16,
  OMCB = 4.17807413224E-3,
  RECB = 2.092566E7,
*
  ZJ2= 0.108264E-2,
  ZJ3= -0.2541E-5,
  ZJ4= -0.1618E-5,
*
* ---------- Table 27 - Trim Model -----------------
*
*
* ---------- Table 32 - Addl Trim Input ------------
*
*
 $END
 $PHASIN
*********************************************
*                   PHASE 3                 *
*                                           *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
```

```
C  PNAME        = 'Ph3',
C  PTITLE       = 'Phase 3',
C  PTYPE        = 'C',
C  CC= '1_Ph2',
C  CR= '1_Ph2',
   NNPP=20,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
   ISTAGE=1,
   SEGL=32*1,
   DTPRINT= 50.0,
   DTINT = 0.5,
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 0.387673122E+02,    TP  = 51.233,
   TILB= 2.0,                TIUB= 300.0,
   TLB = 1.0,                TUB = 1000.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
*          Vel       Azi      Gamd      Alt       Lond       Latd       Weight
*          Alpha      Dalpha
*
   ZI     = 711.57,  90.98,  .96729, 99.797, 280.4,  28.499,  1.05860489E+06,
*           0.0872665,  .044224
   ZF     = 9878.96,  99.6565, .25269, 95456., 291.68,   27.439,  654949.8,
*
*
   NNFC= 1,
C  CNFC = 'MACH',
   YFCLB= .9,
   YFCUB=  .9,
   SNFC =  .9E1,
*
* ---------- Table 21 - Control Inputs ----------
*
   NCT= 1,
C  CCV=  'ALPHA',
   CUL=   1.0,
   CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
   NPC=4,
C  CPC=  'ALPHAD',  'GAMD', 'Q',  'FNORM'
   PLB=  -5.0,       0.0,   600., -1968750
   PUB=   10.0,      89.0,  610.,  1968750
   SCON=  10.0E1,    90.0E1, 6100., 1968750
*
* ---------- Table 24 - Atmospheric Model ----------
*
   IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
```

```
*
 $END
 $PHASIN
*********************************************
*                    PHASE 4                   *
*                                              *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME       = 'Ph4',
C  PTITLE      = 'Phase 4',
C  PTYPE       = 'C',
C  CC= '1_Ph3',
C  CR= '1_Ph3',
   NNPP=25,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
   ISTAGE=1,
   SEGL=32*1,
   DTPRINT= 50.0,
   DTINT = 0.5,
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 0.387673122E+02,    TP  = 51.233,
   TILB= 2.0,               TIUB= 300.0,
   TLB = 1.0,               TUB = 1000.0,
*
* ---------- Table 20 - State / Boundary Condition Specification ----------
*
*         Vel      Azi      Gamd      Alt      Lond      Latd      Weight
*         Alpha      Dalpha
*
   ZI     = 711.57,  90.98,  .96729, 99.797, 280.4,  28.499,  1.05860489E+06,
*           0.0872665,  .044224
   ZF     = 9878.96,  99.6565, .25269, 95456.,  291.68,   27.439,  654949.8,
*
*
   NNFC= 1,
C  CNFC = 'MACH',
   YFCLB=  2.0,
   YFCUB=  2.0,
   SNFC =  2.0E1,
*
* ---------- Table 21 - Control Inputs ----------
*
   NCT= 1,
C  CCV=  'ALPHA',
   CUL=   1.0,
   CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
   NPC=3,
C  CPC=  'ALPHAD', 'Q',  'FNORM'
   PLB=  -5.0,    600., -1968750
```

```
  PUB=   10.0,   1500., 1968750
  SCON=  10.0E1,  1500., 1968750
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
********************************************
*                    PHASE 5                    *
*                                               *
********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph5',
C  PTITLE       = 'Phase 5',
C  PTYPE        = 'C',
C  CC= '1_Ph4',
C  CR= '1_Ph4',
  NNPP=25,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
  ISTAGE=1,
  SEGL=32*1,
  DTPRINT= 50.0,
  DTINT = 0.5,
*
*
* ---------- Table 19 - Time Inputs ----------
*
  TI  = 149.30602,    TP  = 551.233,
  TILB= 2.0,               TIUB= 1000.0,
  TLB = 1.0,               TUB = 1000.0,
*
* ---------- Table 20 - State / Boundary Condition Specification ----------
*
*         Vel       Azi       Gamd      Alt       Lond       Latd       Weight
*         Alpha      Dalpha
*
  ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
  ZF      = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
  NNFC= 1,
C  CNFC = 'MACH',
  YFCLB=  2.9,
  YFCUB=  2.9,
  SNFC =  2.9E1,
*
* ---------- Table 21 - Control Inputs ----------
*
  NCT= 1,
C  CCV=  'ALPHA',
  CUL=   1.0,
  CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
*   NPC=2,
*C  CPC=  'ALPHAD','Q',
*   PLB=  -15.0,   600.,
*   PUB=   15.0,   2000.,
*   SCON=  15.0E1, 2000.,
  NPC=3
C  CPC='ALPHAD', 'Q' 'FNORM'
  PLB=-5.0,   600, -1968750
  PUB= 10.0,  2000, 1968750
  SCON=10.0,  2000,1968750
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
*********************************************
*                    PHASE 6                    *
*                                               *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph6',
C  PTITLE       = 'Phase 6',
C  PTYPE        = 'C',
C  CC= '1_Ph5',
C  CR= '1_Ph5',
  NNPP=40,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
  ISTAGE=1,
  SEGL=32*1,
  DTPRINT= 50.0,
  DTINT = 0.5,
*
*
* ---------- Table 19 - Time Inputs ----------
*
  TI  = 149.30602,    TP  = 551.233,
  TILB= 2.0,               TIUB= 1000.0,
  TLB = 1.0,               TUB = 1000.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
*        Vel       Azi       Gamd      Alt       Lond       Latd       Weight
*        Alpha      Dalpha
*
  ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
  ZF     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
  NNFC= 1,
C  CNFC = 'MACH',
  YFCLB=  6.09,
  YFCUB=  6.1,
  SNFC =  6.1E1,
*
* ---------- Table 21 - Control Inputs ----------
*
  NCT= 1,
C  CCV=  'ALPHA',
  CUL=   1.0,
  CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
*   NPC=2,
*C  CPC=  'ALPHAD','Q',
*   PLB=  -15.0,   600.,
*   PUB=   15.0,   2000.,
*   SCON=  15.0E1, 2000.,
  NPC=4
C  CPC='ALPHAD', 'Q' 'FNORM', 'PCAL(7)'
  PLB=-5.0,   600, -1968750, 0.,
  PUB= 10.0,  2000, 1968750,  250.,
  SCON=10.0,  2000,1968750,   250.,
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
**********************************************
*                    PHASE 7                    *
*                                               *
**********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph7',
C  PTITLE       = 'Phase 7',
C  PTYPE        = 'C',
C  CC= '1_Ph6',
C  CR= '1_Ph6',
  NNPP=20,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
  ISTAGE=1,
  SEGL=32*1,
  DTPRINT= 50.0,
  DTINT = 0.5,
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 149.30602,    TP  = 551.233,
   TILB= 2.0,               TIUB= 1000.0,
   TLB = 1.0,               TUB = 1000.0,
*
* ---------- Table 20 - State / Boundary Condition Specification ----------
*
*          Vel       Azi      Gamd      Alt       Lond       Latd       Weight
*          Alpha      Dalpha
*
   ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
   ZF      = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
   NNFC= 1,
C  CNFC = 'MACH',
   YFCLB=  8.5,
   YFCUB=  8.5,
   SNFC =  8.5,
*
  NCVF=1,ICONST=2,
C CCVF='XPAR(3)',
  FCV=1.0,
* ---------- Table 21 - Control Inputs ----------
*
   NCT= 1,
C  CCV=  'ALPHA',
   CUL=   1.0,
   CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
   NPC=3
C  CPC='ALPHAD', 'Q' 'FNORM',
   PLB=-5.0,   600, -1968750,
   PUB= 10.0,  2000, 1968750,
   SCON=10.0,  2000,1968750,
*
* ---------- Table 24 - Atmospheric Model ----------
*
   IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
*********************************************
*                      PHASE 8                      *
*                                                   *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph8',
C  PTITLE       = 'Phase 8',
C  PTYPE        = 'C',
C  CC= '1_Ph7',
C  CR= '1_Ph7',
  NNPP=50,
*
*   NQV=1,
*      Tot      RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
  ISTAGE=1,
  SEGL=32*1,
  DTPRINT= 50.0,
  DTINT = 1.,
*
*
* ---------- Table 19 - Time Inputs ----------
*
  TI  = 149.30602,    TP  = 551.233,
  TILB= 2.0,               TIUB= 1000.0,
  TLB = 1.0,               TUB = 1500.0,
*
* ---------- Table 20 - State / Boundary Condition Specification ----------
*
*          Vel       Azi       Gamd      Alt       Lond       Latd       Weight
*          Alpha      Dalpha
*
  ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
  ZF      = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
  NNFC= 1,
C  CNFC = 'MACH',
  YFCLB=  10.,
  YFCUB=  10.5,
  SNFC =  10.5,
*
* ---------- Table 21 - Control Inputs ----------
*
  NCT= 1,
C  CCV=  'ALPHA',
  CUL=   1.0,
  CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
  NPC=3
C  CPC='ALPHAD', 'FNORM','Q'
  PLB=-0.0,   -1968750, 0.
  PUB= 10.0,  1968750, 2000.,
  SCON=10.0, 1968750, 2000.,
*
* ---------- Table 24 - Atmospheric Model ----------
*
  IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
*********************************************
*                     PHASE 9                       *
*                                              *
**********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph9',
C  PTITLE       = 'Phase 9',
C  PTYPE        = 'C',
C  CC= '1_Ph8',
C  CR= '1_Ph8',
   NNPP=40,
*
*   NQV=1,
*      Tot      RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
   ISTAGE=1,
   SEGL=32*1,
   DTPRINT= 50.0,
   DTINT = 1.,
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 149.30602,    TP  = 551.233,
   TILB= 2.0,               TIUB= 1000.0,
   TLB = 1.0,               TUB = 1500.0,
*
* ---------- Table 20 - State / Boundary Condition Specification ----------
*
*          Vel       Azi       Gamd      Alt       Lond       Latd       Weight
*          Alpha      Dalpha
*
   ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
   ZF      = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
   NNFC= 1,
C  CNFC = 'MACH',
   YFCLB=  14.,
   YFCUB=  14.05,
   SNFC =  14.0,
*
* ---------- Table 21 - Control Inputs ----------
*
   NCT= 1,
C  CCV=  'ALPHA',
   CUL=   1.0,
   CLL=  -1.0,
*   CULD=  0.1,
*   CLLD= -0.1,
*
* ---------- Table 22 - Placards ----------
*
   NPC=2
C  CPC='ALPHAD', 'FNORM','Q'
   PLB=-15.0,   -1968750, 0.
   PUB= 15.0,  1968750, 2000.,
   SCON=15.0, 1968750, 2000.,
*
* ---------- Table 24 - Atmospheric Model ----------
*
   IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $PHASIN
*********************************************
*                  PHASE 10                 *
*                                          *
*********************************************
*
* ---------- Table 16 - Phase Definition ----------
*
C  PNAME        = 'Ph10',
C  PTITLE       = 'Phase 10',
C  PTYPE        = 'C',
C  CC= '1_Ph9',
C  CR= '1_Ph9',
   NNPP=30,
*
*   NQV=1,
*      Tot     RJ Fuel
*C  CQV= 'WDOT' ,
*C  CQ = '1_Ph2',
*
   ISTAGE=1,
   SEGL=32*1,
   DTPRINT= 50.0,
   DTINT = 1.,
*
*
* ---------- Table 19 - Time Inputs ----------
*
   TI  = 149.30602,    TP  = 551.233,
   TILB= 2.0,               TIUB= 1500.0,
   TLB = 1.0,               TUB = 2500.0,
*
* ---------- Table 20 - State/Boundary Condition Specification ----------
*
*          Vel       Azi      Gamd      Alt       Lond       Latd       Weight
*          Alpha      Dalpha
*
   ZI     = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
   ZF      = 1959.3468,  86.427,  18.218, 33619.852, 280.35,  28.18,  569157.27,
*
*
   NNFC= 4,
C  CNFC = 'VELI','GAM', 'INCD', 'RADI'
   YFCUB=  25845.,  0.0001, 28.53, 2.12294674E+07,
   YFCLB=  25842., 0.0,    28.5,  2.12294674E+07,
   SNFC =  25845., 0.001,   28.53, 2.12294674+07,
*
* ---------- Table 21 - Control Inputs ----------
*
   NCT= 1,
C  CCV=  'ALPHA','XPAR(3)',
   CUL=   1.0, 1.0,
   CLL=  -1.0, 0.1,
*   CULD=  0.1,
*
* ---------- Table 22 - Placards ----------
*
   NPC=3
```

```
C  CPC='ALPHAD', 'FNORM','ACCT',
   PLB=-15.0,   -1968750,1.0,
   PUB= 15.0,  1968750,3.05,
   SCON=15.0, 1968750,3.05,
*
* ---------- Table 24 - Atmospheric Model ----------
*
   IATMOS=2,
*
* ---------- Table 27 - Trim Model -----------------
*
* ---------- Table 32 - Addl Trim Input ------------
*
 $END
 $SPC
Begin SNOPTInt
*
*  Major iterations are set via OTIS' NIT parameter
*  Minor iterations            5000
*  (Minor Iter Default=max{1000,5m}; where m=tot# of constr)
*  Iterations limit            xxxxxx
*  (Total Minor Iter Default=max{10000,20m}; where m=tot# of constr)

   Derivative line search          * The default

   Major Print level          000001
*                        (JFDXbs)
   Minor print level              0

   Major Optimality  Tolerance      5.0e-5
   Major Feasibility Tolerance      1.0e-5
        Feasibility Tolerance      1.0e-5
   Minor Optimality  Tolerance      1.0e-8
   Minor Feasibility Tolerance      1.0e-8
   Crash option                   0

   Print frequency                1
   Summary frequency              1
   Solution                  yes

   Verify Level                  -1

   Check frequency                1
   Function precision            1.0E-9
   Linesearch tolerance           0.10
   Major step limit              0.50

End Mode4Tst
  $END
```