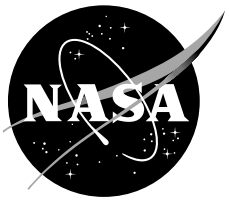
NASA/TM–2005-213449



Traj_opt User's Guide

David A. Saunders

March 2005

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

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

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

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

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

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

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

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

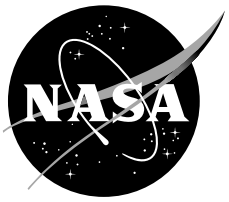
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM–2005-213449



Traj_opt User's Guide

David A. Saunders
ELORET
Ames Research Center
Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

March 2005

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

Contents

Traj_opt User's Guide

David Saunders*

Ames Research Center

Summary

This document describes a new trajectory optimization program, *Traj_opt*, and how to use it for calculating several kinds of optimal trajectory not involving propulsion. *Traj_opt* version *January 22, 2003* is covered. This version is a slight revision of *Traj_opt* version August 16, 2002, which was intended for participation in the Systems Requirements Review (SRR) of next-generation Reusable Launch Vehicles (RLVs) under NASA's Strategic Launch Initiative. SRR evaluations of industry proposals were scheduled to start in November 2002 but were postponed indefinitely pending maturation of NASA's Advanced Engineering Environment (AEE). AEE integrates numerous multi-disciplinary tools necessary for analyzing RLV performance and assessing safety and life cycle costs. *Traj_opt* is an AEE tool suited to calculating optimized ascent abort trajectories for the CRV/CTV (Crew Rescue/Crew Transfer Vehicle) and CEM (Crew Escape Module) components of a launch system.

The software is introduced to the new user, with further details available from indicated references for the underlying trajectory analysis and constrained optimization packages. Aspects of the loosely coupled optimization approach, the choice of optimization variables, and the implementation of certain linear and nonlinear constraints are discussed in an extended introduction. Addressed as much to the newcomer to numerical optimization as to the would-be user of *Traj_opt*, this introduction covers most of the standard issues associated with applications of gradient-based optimization. The *Traj_opt* specifics that typically follow each discussion serve as illustrations but also supplement the detailed *Traj_opt* input descriptions of a later section.

A general information section provides an overview of all input and output files involved in a *Traj_opt* run, followed by a discussion of the standard trajectory types for which *Traj_opt* is suitable, and concludes with performance figures for a representative workstation.

* Senior Research Scientist, ELORET, Ames Research Center, M.S. 230-2, Moffett Field, CA 94035/Sunnyvale, CA.

Detailed control input descriptions have been extracted from the source code—where they belong for timely maintenance—and preparation of other input files is described. These descriptions are at the level below any automation via graphical user interfaces that is also part of preparing for AEE applications. Likewise, interpreting and displaying results are outlined at the level employed by the author prior to AEE integration. Multiple appendices show the *Traj_opt* control files recommended for the standard series of reentry and ascent abort trajectories.

1. Introduction

Unpowered trajectory optimization provides a way to quantify the aerodynamic performance of an aerospace vehicle, either during atmospheric entry from orbit or during an emergency if the ascent to orbit is aborted after launch. For a vehicle with known aerodynamic, material and mass properties, control schedules can be tuned to optimize a performance objective such as cross-range (distance reachable away from the plane of the orbit) or time of abort from launch (earliest or latest for which a target landing site is reachable) while satisfying constraints on surface temperature, dynamic pressure (or structural integrity) and acceleration (or crew survivability). The relative performance of alternative vehicle shapes can thus be compared, and the sensitivity of each vehicle to the various constraints can also be estimated.

Traj_opt Origins: *Traj_opt* was developed at NASA Ames Research Center (ARC) in 2000-2001 as a straightforward means of evaluating the benefits of the sharp leading edges that newly developed ultra-high temperature ceramics promised to make viable. The Crew Transfer Vehicle application avoided propulsion issues, allowing the in-house 3-degrees-of-freedom *Traj* (ref. 1) analysis package to serve in conjunction with a constrained optimizer (*NPSOL* (ref. 2)) familiar to the CTV team from earlier shape optimization applications. Using aerodynamic databases prepared at ARC with a combination of hypersonic impact pressure methods and low-speed panel methods, preliminary *Traj_opt* results comparing the blunt HL-20 and the SHARP-V5 configuration were obtained and published in mid-2001 (ref. 3). Since that time, numerous refinements have been implemented, the most important being the option to include distributed heating constraints.

Loosely Coupled Approach: The two packages are loosely coupled in the sense that *Traj* is treated as a "black box" called many times to evaluate objective functions and constraints and to estimate their derivatives. Among other virtues involving separation of functionality and moderate numbers of variables and constraints, loose coupling has the advantage that every function evaluation produces a legitimate trajectory. In particular, even if some constraints are not satisfied at the end of a run, the result is still meaningful. In contrast, the tightly coupled implicit optimization methods must converge or the solution is nonphysical. Lacking experience with collocation (ref. 4) methods, the author understands that the more elaborate implementations can be impressively fast and robust, while the *Traj_opt* approach is quite robust but relatively slow (spending most of its time calculating gradients via finite differencing, which in turn require high-precision analyses). More on gradients and computational performance appears below.

Traj Refinements for Optimization: *Traj* was a thoroughly validated analysis tool when the task of linking it with an optimizer was undertaken. Its 4th-5th-order Runge-Kutta-Fehlberg differential equation algorithm is efficient and accurate, typically requiring well under a second on a contemporary workstation.[*] Following its initial remodularization into fully-reentrant function form, refinements to *Traj* prompted by the present application have included options to perform spline interpolation of angle of attack and bank angle with respect to time, more careful control of the

[*] For example, a 300 MHz Silicon Graphics Octane

differential equation time steps and the stored time history, and precise capture of the specified end-of-trajectory condition.

Traj↔Traj_opt Communication: *Traj* is written in C; *Traj_opt* is in Fortran 90. Two-way communication is performed through an argument-driven C interface function, trajectory.c. Copying of data between equivalent data structures is unavoidable. Initially, end-of-trajectory quantities served for most likely objective functions and constraints, but more thorough implementations of certain constraints have led to transfer of more and more quantities from the stored time history.

Traj's original maneuver script scheme is avoided here. Instead, spline control points for angle of attack and bank angle are manipulated as the main variables during optimization. *Traj* interpolates these control points or knots at every Runge-Kutta time step. The available "monotonic" spline option ensures that the piecewise cubics do not exceed the relevant control point range between knots. This is probably most relevant for trajectories involving major bank reversals.

Splined Control Schedules and Other Practicalities: Such continuous variation of Alpha and bank implied by spline interpolation may not be achievable in practice, amounting as it does to continuous expenditure of energy. Results therefore tend to represent the best that is theoretically possible. Actually, *Traj_opt* can now be run in step-function mode as needed for simulating closed-loop control of entry probes with small numbers of variables during planetary aero-capture. The angle steps may be of variable duration in this case. Normally, though, 40-80 control points are used for each of Alpha and bank, and their locations are fixed in time.

Likewise, no account is taken of possible uncertainties in the data—atmospheric, aerodynamic, or aerothermal—apart from a limited option for perturbing the vehicle's lift-to-drag ratio (L/D). Thus, absolute numbers from *Traj_opt* should be interpreted with likely dispersions in mind (perhaps estimated through sensitivity studies). Nevertheless, comparisons of optimized trajectories among vehicles under comparable conditions still provide valid measures of relative vehicle performance.

Another practical issue associated with spline control points for Alpha and bank is the choice of spacing (in time) and the total duration. While the spacing is completely arbitrary (and the spacing for Alpha is independent of that for bank), these choices can clearly affect the optimized solution. Normally, the ***recommended practice*** is to use the same uniform distribution for the two sets of control points (knots) and to extend beyond the expected end-of-trajectory in time by a modest fraction of that time. Falling short of the end of a trajectory means the angles are held constant at the last knot values. Some loss of optimality in such a solution is likely. "Wasting" a few optimization variables off the end of the trajectory is preferable, and costs little because gradient elements that are clearly zero are readily set without explicit calculation.

Objective Function(s): Any optimization process minimizes or maximizes some quantity or combination of quantities named the *objective* function or the *cost* function. In the case of *Traj_opt*, a quantity commonly specified to be minimized during an optimal trajectory calculation is the accumulated heat load at the stagnation point, while maximizing cross-range during reentry from orbit is another common objective. A likely further contribution to the objective function, tending to promote smooth solutions for the main variables, would be a (relatively small) multiple of a measure of the *lack* of smoothness in the time variation of the angle of attack and/or bank angle. Such a

contribution is termed a *penalty function* added to the objective. More direct ways of constraining the solution are discussed below under the headings ***Linear Constraints*** and ***Nonlinear Constraints***.

Interfacing with the Optimizer: Apart from numerous inputs such as the starting guesses for the variables, the widely used *NPSOL* sequential quadratic programming (SQP) package for nonlinear constrained optimization requires two application-specific subroutines for its OBJFUN and CONFUN arguments.* Simple-minded implementation of these two routines for evaluating objective functions and nonlinear constraint functions would lead to duplication of most trajectory calculations—once for the objective, and once for the constraints. Therefore, given that CONFUN is known to be called by the optimizer *before* OBJFUN at all times except when the number of nonlinear constraints is zero, *Traj_opt* has the option to evaluate the objective within CONFUN and then avoid its immediate reevaluation within OBJFUN.

Gradient calculations are also performed within these two routines (preferably both for the constraints and the objective within CONFUN), via finite differencing and use of Derivative Level 3 among the optional *NPOPT* parameters. Thus, *NPOPT*'s own finite differencing option is not used because it would normally lead to duplicate trajectories for objective and constraint evaluations. *Traj_opt* permits 2- or 3-point differencing, or both (more on which below).

Starting Guesses: Ideally, solutions from previous similar cases will be available to start a new application of *Traj_opt*. Good starting guesses are not normally crucial, but naturally they help. In the absence of similar past results, good *enough* starting guesses may take multiple tries, because even the most powerful optimizer can all-too-easily encounter infeasibilities that are not surmountable without input adjustments. For instance, trying to start a large-down-range reentry case using just angle of attack and zero bank everywhere invites trouble with skipping in and out of the atmosphere. [Maximizing down-range is actually an ill-defined problem, as explained further below.] Even a cross-range case can skip or overheat badly with poor initial choices for Alpha, and not recover.

Huge constraint violations mean large associated gradient elements, and hence poor initial scaling and likely ill-conditioning. Yet large magnitudes are quite possible when "path constraints" (see below) are violated, because of the way they are evaluated as integrals of time-history quantities. Dynamic pressure (pascals x seconds) and surface temperature (°K x seconds) are particularly prone to apparently massive violations that still have to be reduced to zero. Nevertheless, with reasonably sensible data, *SNOPT* generally manages to produce a solution in 200-300 iterations in the presence of half that many variables, which is better performance than we have a right to expect from gradient-based methods. Moreover, if a run is going to fail, it usually does so fairly early. Thus, trial-

* Actually, the same authors' *NPOPT* has an identical calling sequence and is the form employed at ARC. Most recently, the *NPOPT* interface to *SNOPT* (ref. 5) has been adopted for *Traj_opt*—***not*** for the ability of *SNOPT* to handle large problems via sparse techniques (quite unnecessary here), but rather to benefit from a few refinements made to this implementation, the dense-matrix forms having been frozen for some years. Retaining the *NPOPT* call avoids dealing with sparse matrices at the *Traj_opt* level. Conceptually, SQP methods perform quasi-Newton minimization by linearizing the nonlinear constraints and calculating each search direction by solving a "quadratic program"—that is, by minimizing a quadratic approximation to the objective function subject to linear constraints.

and-error repeats with intelligent adjustments in between are normally not too painful. Experience helps with the choice of starting guesses.

A related aspect, even with good initial estimates, is a tendency to take too large a *first* step that reduces the objective significantly but turns small constraint violations into large ones. Use of STEPLIM = 0.01 (i.e., limiting per-step changes to 1%) or smaller can control the tendency to some extent. This is particularly common when restarting to clean up a solution with minor violations: we don't particularly care about the precise value of the objective, but we do care about not accepting solutions that don't really satisfy the constraints, particularly when we're performing parametric sensitivity studies. Ideally, the optimizer would focus on the (small) violations more than on the objective. *SNOPT* does indeed have a ***Violation limit*** among its optional inputs as another way to perform the balancing act. Having only belatedly discovered this option, the author has little experience with it and can merely refer the reader to the *SNOPT* User Guide. Also, it might be thought that adjusting the relative scaling between objective and constraints could help, but all input scale factors should be carefully chosen to within an order of magnitude in order to equilibrate largest gradient elements (roughly), so there should not be much maneuvering room here.

As colleague Peter Gage has noted, an interesting research topic would be the application of genetic algorithms to starting guesses in the presence of constraints handled in some way other than as penalty contributions to the objective. Meanwhile, *Traj_opt* depends upon semi-educated guesses and the power of *SNOPT*, sine qua non.

Terminating the Optimization: Particularly in the presence of finite difference estimates for the first derivatives, terminating an optimization iteration at a point where any further improvement would be negligible and wasteful is a practical problem in general with no guaranteed solution. Like any gradient-based method, *SNOPT* behavior is highly dependent upon the accuracy of the gradients, and it is fairly sensitive to their scaling. Some of its input tolerances can also affect the path taken towards a minimum, even early in the iteration. Poor gradients and/or poor scaling can lead to early termination with constraint violations at worst, or painfully slow progress with abnormally short step lengths at best. Fortunately, the inputs that should produce good *Traj_opt* gradients are well understood.

Factors affecting the gradient accuracy include the finite differencing intervals, the cap on the Runge-Kutta time steps, and the resolution specified for the stored time history, which is used for certain constraint calculations. Those calculations must be performed smoothly, in the sense that small changes in the data (the variables) lead to smooth changes in the constraint values, and hence in their derivatives. As described further below, *Traj_opt* employs spline quadrature of time-history quantities for several of the constraints to help with these smooth variations.

An illustration of what ***not*** to do is provided by the *two* time-step-size-related inputs to *Traj* where there was formerly just one: The stored time history was originally intended for plotting purposes only, being written to 'traj.out' at the end of a *Traj* analysis at precise intervals (1 second, 2 seconds, or whatever was specified). If these output evaluations of the differential equations were sufficiently close to the natural steps suggested by the local step control algorithm, they would be substituted for those steps. Thus, the *plotting* resolution could actually affect the sequence of Runge-Kutta time [steps] taken—an intolerable situation when it comes to finite difference gradient estimates. Now,

Traj has the option to keep the end-of-trajectory time step completely independent of the plottable resolution, with the side effect that the stored steps are no longer at the precisely specified intervals—rather, they are the natural steps taken that are nearest to the indicated resolution. Given that these stored steps enter into some constraint calculations, we see there still remains a subtle source of possible noise. Finer resolution improves gradient accuracy.

Experience shows that satisfying reasonable nonlinear constraints to within a small tolerance (that is, satisfying the feasibility convergence test) is much more likely than satisfying the test for optimality (roughly, a measure of the number of significant digits in the objective at the solution). One ***suggested strategy*** is to start by specifying fairly loose feasibility and optimality tolerances along with 2-point differencing. *Traj_opt* then has the option to tighten the tolerances and perform a "warm" restart (via a second call to *NPOPT* in the same run) with *central* differencing specified. See section 3 for details about control inputs TIGHTEN, TOLNLIN and TOLOPT, and also finite differencing intervals H_ALPHA and H_BANK. However, as workstations become more and more powerful, it may well be more convenient to specify three-point finite differencing from the start, thereby trading double the computational cost per major iteration for greater reliability and (most likely) fewer iterations.

A *Traj_opt* convergence history is illustrated in figure 1. The run maximized cross-range during normal return from orbit for the HL-20 vehicle. The standard dynamic pressure limit was never reached, and G load was low throughout, so the feasibility curve refers to the distributed surface heating constraint (which consists of a single point near the blunt nose in this test case).

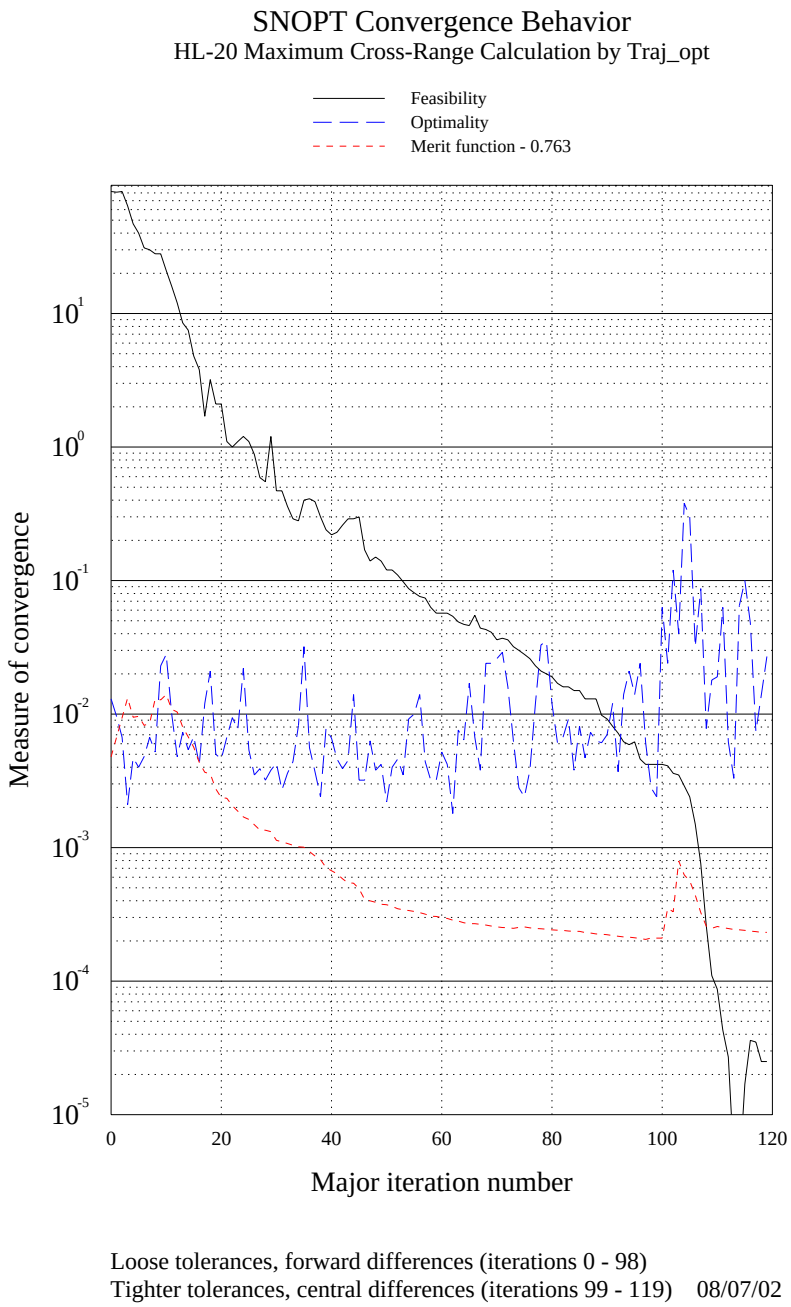


Figure 1. Sample *SNOPT* convergence history.

The above strategy of starting with loose tolerances then switching to central differencing with tighter tolerances is employed. Following the (automatic) switch at iteration 99 (in this case), the constraint violation is nicely reduced further but the optimality measure actually worsens, with no sign of converging. The merit function shown reflects larger values of *SNOPT*'s internal penalty parameter, used as a multiple of the squared constraint violations to help ensure descent directions. The cross-range-related objective function (not plotted) actually stayed close to flat during the last 50-odd iterations. At termination, the optimizer reported "EXIT -- optimal, but the requested

accuracy could not be achieved." This is a far more common occurrence than the occasional "EXIT -- optimal solution found."

Bounds on the Variables: By definition, *un*constrained optimization cannot confine the variables to some space, let alone impose more general constraints, except crudely via barrier or penalty functions. Constrained optimization requires vastly different strategies, starting with simple bounds placed on the variables. These can range from equal upper and lower bounds (to fix the variable) to finite distinct bounds to no bound in either or both of the directions. Variables are "free" if they are not on one of their bounds.

In the case of *Traj_opt*, the aerodynamic database normally indicates the upper and lower limits on Alpha, although lowering the upper bound may be desirable to avoid excessive body flap deflections if the vehicle is poorly trimmed at high angles of attack. Bank angle limits are somewhat arbitrary unless the full ±180° range is allowed. Indeed, allowing the vehicle to be fully inverted at high Alpha has been found beneficial for trajectories needing to reduce altitude as rapidly as possible without much change in heading—as in the case of certain ISS ascent aborts to Gander, Newfoundland, and minimum heat load trajectories with zero cross-range.

Linear Constraints: A linear constraint represents a linear equation or inequality involving one or more of the optimization variables. If it is an equality constraint, equal upper and lower bounds are specified for the linear combination. Otherwise, distinct bounds are specified, possibly as values that are interpreted as $-\infty$ or $+\infty$. An optimization typically begins with a feasibility iteration that adjusts the starting-guess variables as necessary to satisfy the specified linear constraints. These remain satisfied thereafter, to within the linear feasibility tolerance.

The only linear constraints implemented so far in *Traj_opt* are first-order approximations to limiting pitch rate and roll rate. The ordinates of adjacent pairs of angle control points (spline knots) are constrained to be close enough together that the indicated rate is not exceeded by the slope of the straight line joining them. A roll rate constraint is appropriate if large bank reversals are indicated (as in minimum heat load/zero cross-range cases, or certain abort trajectories).

Traj_opt translates the named linear constraint(s) into the appropriate number of constraints as seen by the optimizer. For instance, a 'ROLL' input leads to the set up of NDV_BANK – 1 linear constraints, where NDV_BANK is the number of bank control points allowed to vary.

Nonlinear Constraints: The optimization variables often do not enter explicitly into the equations representing imposition of bounds on key quantities. In the case of *Traj_opt*, for instance, the terminal latitude and longitude and the peak dynamic pressure and surface temperatures cannot be expressed as functions of the Alpha and bank schedule control variables—and certainly not as linear functions. Such nonlinear constraints on key quantities are implemented simply by evaluating the quantities from the current data in the CONFUN routine, along with their partial derivatives when indicated by the optimizer.

In the case of *NPOPT/SNOPT*, nonlinear constraints need not be satisfied by the starting-guess solution. Indeed, the optimization process typically amounts to gradually reducing initial violations until they fall below tolerance, while reducing the objective function as much as possible (although

the objective can commonly *increase* from the starting value). Imagine starting from a result obtained with some constraint omitted (and not satisfied). Adding the constraint means something has to "give," and typically it is the objective function that cannot be as small as it had been. (Otherwise, a *local* minimum must be involved, and because ***local minima*** are *always* a possibility with these gradient-based methods—yet seldom cause difficulties in practice—no more shall be said here on that subject except to note that relative flatness of the "design" spaces, meaning ***ill-defined minima*** in the presence of 80-150 "design" variables, are all-too-common in this application, contributing no doubt to the above convergence issues.)

In the trajectory world, constraints are imposed either at "events" (or boundaries, start- and end-of-trajectory being the only possibilities for *Traj_opt*) or between events, where the term "path constraint" is commonly used. In the present case, ***initial conditions*** can vary only for abort trajectories, via the TABORT time-from-launch variable, which has simple bounds. ***Final conditions*** may be constrained in numerous ways, including target (latitude, longitude) coordinates, flight path angle, velocity, and so on. These are all implemented trivially. Two ways of implementing path constraints are discussed next.

Path Constraints: The simplest way to constrain a quantity such as G load throughout a trajectory is to require that its *peak* value be below a specified level. Typically, a trajectory analysis tool tracks such peaks, so their use as constraint values is as trivial as for event constraints. This approach is workable, but not ideal, because peaks are prone to moving about in time, (a) if the quantity is at its limit for extended periods, or (b) if it exhibits two or more spikes. Small changes in the data can lead to big changes in the location of a peak—the very sort of situation that can lead to confusing gradient estimates during finite differencing.

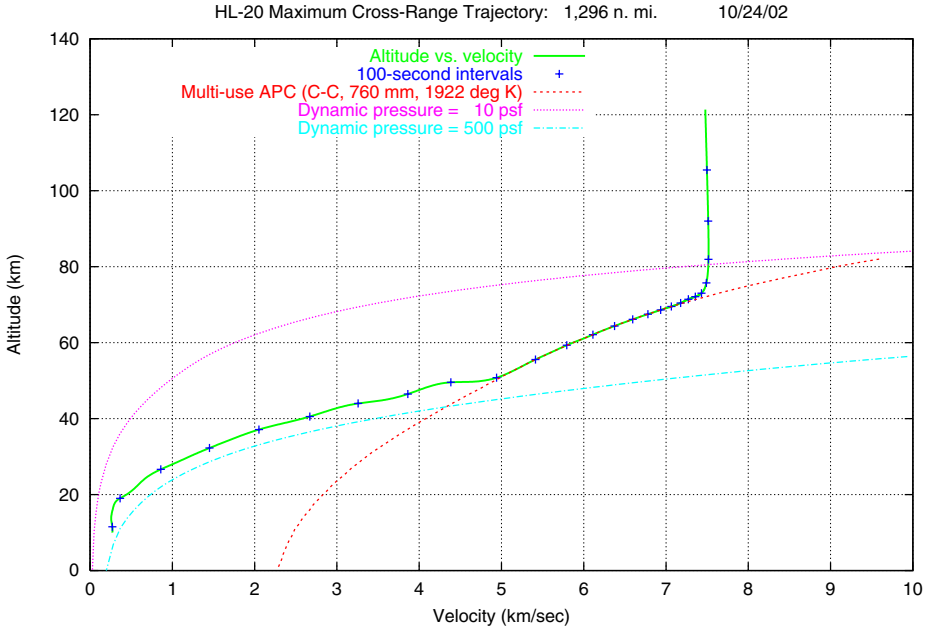


Figure 2. Aerothermal performance constraint (APC) illustration. Note that APCs are more appropriate for sharp vehicles. This one for HL-20 would be better handled among the distributed heating constraints discussed below.

Short of imposing constraints at every time step—possibly thousands of them—as collocation methods allow, a method that combines all the violations in time can be used. Ten of the nonlinear constraint options in *Traj_opt* employ spline quadrature of the violations found in the stored time history. This strategy works well for both simple upper bounds (as on dynamic pressure or surface point temperature, with time as the abscissa) and for more general constraints specified as curves in a different space, namely altitude vs. velocity as shown in figure 2. In either case, a "curve" of differences between the current data values and the constraint is established, suitably zeroed where there is no violation. The area under this curve is then integrated via a quadrature variant of the local cubic spline utility used for interpolating Alpha and bank. Its monotonic option (disallowing spline excursions between data points) handles spiky data particularly well.

Originally, violations of the so-called aerothermal performance constraint (APC) (ref. 3) were calculated as "areas" in (velocity, altitude) space, but now *Traj_opt* always uses time as the abscissa to be certain that "x" is strictly ascending or descending—not always true with velocity.

When the curve of differences contains no constraint violations, some nonzero measure for the constraint value is considered helpful (in the printed output as well as to the optimizer). The choice in *Traj_opt* is to switch from an "area" to a "shortest distance" when there is no violation. The constraint remains continuous but its derivative does not. Note that if a constraint is active, this means even at convergence it is probably slightly violated to within tolerance, so the derivative discontinuity should seldom be detected by the optimizer. *SNOPT* appears to handle such constraints well, and they are certainly preferable to the use of simple peak violations.

More on Distributed Heating Constraints: In the case of the distributed heating constraints (SURF1[R] for temperature, and SURF2[*] for heat flux), the question arises as to whether to implement a separate constraint for each surface point being constrained, or instead to combine the violations of all surface points. Given that optimized solutions almost invariably contain no more than one or two of the points at a limit, the latter choice is taken in *Traj_opt*, simplifying constraint nomenclature considerably. For instance, point 14 in figure 3 is the only point at its limit. Note how well the constraint is satisfied (yet active) for more than 1,500 seconds. A cockpit canopy point (not shown) would also typically be at its temperature limit.

[*] Heat flux is now understood to be *not* independent of wall temperature, so the use of SURF2 should be redundant if the appropriate limits are consistent. See also the SURF1R form (p. 25), which is more accurate than SURF1. (Here, R ⇒ heat *rate*-based temperatures.) Heat flux can still be tabulated and plotted, but need not be constrained.

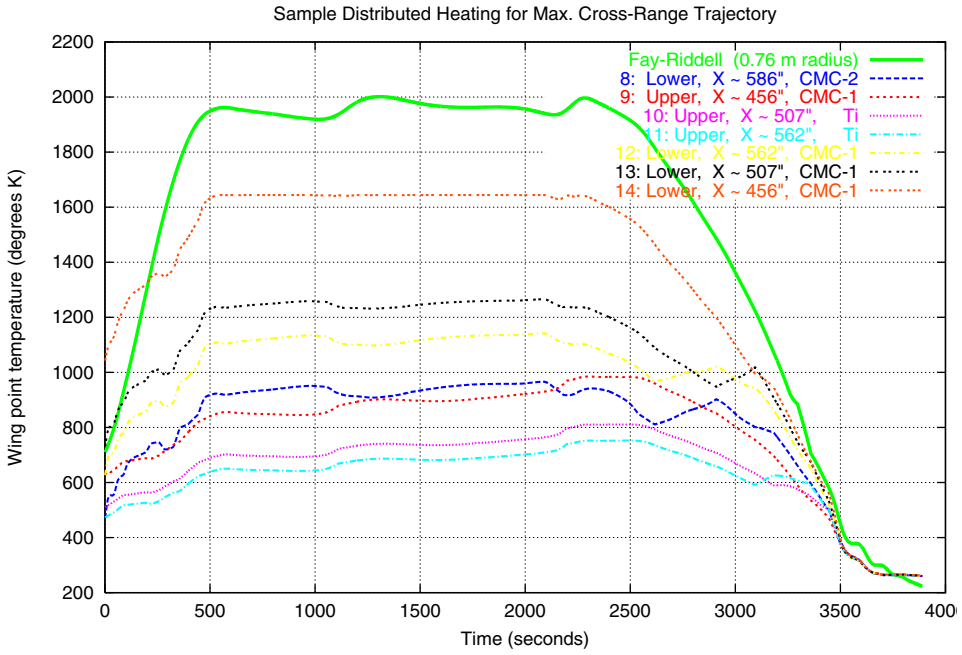


Figure 3. Distributed heating example, with just one surface point at its limit (second curve from top).

This plot also illustrates both forms of surface heating calculation performed with *Traj_opt*. The time histories of the numbered surface point temperatures are interpolated in (Mach, Alpha, Q) space, where Q is dynamic pressure. These interpolations are performed at the *Traj_opt* level (outside *Traj*), using the stored time history from *Traj* and the aerothermal database. The curve labeled Fay-Riddell refers to *Traj*'s own stagnation-point heating option (-q at the command line when *Traj_opt* is invoked, or "Heating Rad_equil_only" instead of "Heating No_model" in 'traj.in'). This calculation for some reference nose radius such as that of the HL-20 may be redundant in the presence of an aerothermal database, but it provides the only measure of heat load that is usable for optimization at present. (The distributed heat flux histories are actually integrated, but the mechanism intended for some sort of eventual TPS sizing capability still lacks an appropriate means of defining the necessary area-based weights.)

Note also that the radiative heat flux predicted by Fay-Riddell is normally insignificant for reentry from Earth orbit, so the values reported during typical *Traj_opt* calculations are effectively those from the Tauber-Sutton heating relation (ref. 1).

Path Constraint Bounds: Use of quadrature as described for evaluating a path constraint means one of the bounds is *zero* and the other is some reasonable bound on the measure for the case of no violation. This means that for something like a G load constraint, the desired limit (say 3 G) cannot be entered as a normal constraint bound input. Here, an adjacent input (XNLCON) intended for just such a situation is employed instead.

For instance, the ACCEL constraint penalizes *positive* areas of the acceleration magnitude time history above the indicated limit, so the *upper* bound should be zero. The lower bound, being on

shortest “distance” in units of G, should properly be –3.0 if XNLCON = 3.0, but the round number –10.0 serves just as well.

The UPPERC (upper corridor) analogue of the APC behaves similarly, with an *upper* bound of zero to prevent excursions *above* the (velocity, altitude) curve corresponding to minimal control surface effectiveness (normally the curve representing 10 psf dynamic pressure). The lower bound is therefore a negative value such as –50 km, representing the *largest* altitude possible for the *nearest*-distance measure of *no* violation. (This can be confusing at first.)

Conversely, the APC implementation penalizes excursions *below* the constraint curve (treated as *negative* areas), so the *lower* bound is zero and the upper bound is +50 km or so.

Finally, lumping all the surface points into one distributed heating constraint means the distinct limits for each point must be entered by another mechanism, namely an ancillary file—see section 4.

Scaling: The magnitudes of the optimization variables, the objective function and the nonlinear constraints, and their first derivatives, are all expected to be “reasonable” for good performance by an optimizer (i.e., < 100 or so). The application has to employ scale factors, and sometimes shifts, to cope with awkward units. Magnitudes of various quantities will change as the point in the optimization space changes, so there is no perfect choice for scaling.

Recommended scaling practice: Scale the variables to be $O(1)$, and the objective and nonlinear constraints to be such that their (largest) first derivatives are of comparable magnitude. In other words, the rows of the Jacobian matrix $[\partial c_i / \partial x_j]$ (printed as *columns* in ‘traj_opt.out’) should be roughly equilibrated. Any odd large element is likely to be a sign of sporadic noise that will very likely cause the optimization to fail on the next step (no longer likely with good *Traj_opt* inputs).

Normalizing the expected range of each variable to [0, 1] or to [-1, 1] gives two common ways of determining scales and shifts. The convention adopted in *Traj_opt* is that the (scaled) variable seen by the optimizer times the scale factor is the value used outside the optimizer when the variable is applied, although others define the scale as the value used for the normalization.

Traj_opt automatically sets up and scales the main optimization variables (Alpha and bank control points) to be $O(1)$. To be precise, all Alpha knots have a scale factor of 20, and all bank knots are scaled by 100. This simplified scheme facilitates occasional recovery of intermediate iterations from the printout of the *scaled* variables in ‘‘traj_opt.out’.

The single other variable known to *Traj_opt* at present is TABORT. With its physically meaningful value being a few hundred seconds from launch, this is typically initialized between 1 and 5 and scaled by 100 when used to interpolate the trajectory initial conditions from the ascent data. Note that the initial value and the bounds (and the finite difference interval) are *scaled* values in the input file.

In contrast, the linear constraints (PITCH, ROLL) and simpler nonlinear constraints such as ENDLAT are entered with bounds in *real* units. The scale factor is applied to the bounds upon input, and to each nonlinear constraint whenever it is evaluated.

For constraints calculated via quadrature from time history data, the scale factor must be seen as being applied to an *area*, and for long trajectories, such areas (of violation) can be large, depending on the units. Suggested scales are as follows: 0.1 – 0.01 for APC and UPPERC (km) and for ACCEL (Gs), and 0.001 – 0.0001 for DYN_PR (pascals) and SURF1[R] (°K). The smaller values may be more appropriate a long way from the solution, where large violations are possible; the larger values should help refine the solution better once most of the violations have been eliminated. The scale factor can also be smaller where slight violations are not a concern (as with dynamic pressure), but remember that the constraint scale factors are key to achieving roughly equilibrated columns in the (transposed) Jacobian.

Finally, for the various possible contributions to the objective, the multiplier that turns them on or off also serves as the scale factor. For instance, a minimum heat load trajectory case would use RHO_HEAT_LOAD = 0.0001 if the expected solution is about 50,000 J/cm^2. Again, to *max*imize the time from launch of an ascent abort for which a target landing site is reachable, using RHO_TABORT = -0.01 should be appropriate.

Finite Difference Intervals: Assuming the variables are $O(1)$, conventional wisdom for the choice of interval for 2-point difference approximations to first derivatives is $\underline{h} = \sqrt{\varepsilon_{machine}}$ or $\sqrt{\varepsilon_{obj}}$ for full precision or limited precision functions, respectively, where ε refers to the smallest value added to 1 (or to the function) that makes a meaningful change. For central differencing, the simple choice is $h_c = (\underline{h})^{2/3} = \underline{h}\,(\varepsilon)^{-1/6}$. However, the careful would-be optimizer performs a study of the effect of $\underline{h}$ and h_c on the first derivatives of the objective and the nonlinear constraints. Some negative power of 10 for $\underline{h}$ should be shown to produce derivative estimates that are imperceptibly different (when plotted) from those obtained with the next smaller power of 10. The corresponding h_c on the same plot helps confirm the choice.

Of course, optimal choices may vary with the point in optimization space, and common values of h for many variables may not be viable (although scaling them to $O(1)$ promotes this possibility). For *Traj_opt*, the recommended inputs for Alpha and bank are 10^{-6} and 10^{-6} degrees (before scaling). 10^{-5} degrees for bank is also satisfactory. For TABORT, a physical interval of 10^{-4} or 10^{-5} seconds is suggested, meaning an input of 10^{-6} or 10^{-7} if the scale factor used is 100.*

Lift and Drag Calculations: For certain analytic shapes, *Traj* can calculate lift and drag explicitly, and the reader is referred to the *Traj* User Guide[1] for the details. Arbitrary geometries require an external aerodynamic database containing coefficients of lift, drag and pitching moment (or possibly flap deflection in place of C_M if the aerodynamics are trimmed), as functions of Mach, angle of attack, and Reynolds number or $\log_{10}$(Reynolds number). In hindsight, use of dynamic pressure as the third independent variable may have been preferable. (Indeed, this is the case for the distributed surface heating tables.)

The aerodynamic coefficients are prepared as rectangular tables, typically 18 x 10 x 9, although denser is better. At ARC, a combination of (inexpensive) hypersonic impact methods and low-speed

* Incidentally, TABORT is both a variable and the objective (or its negative), so no matter what the choice of h, the derivative with respect to this variable is always ±1 for the objective, and it is 0 for all other variables. Derivatives of the nonlinear constraints with respect to the TABORT variable, however, are affected by the choice of h.

panel methods—either *HAVOC*[6] + *VORVIEW*[6] or the more recent *CBAERO*[7,8]—is employed to generate the aerodynamics tables and analogous tables of surface temperatures and heat fluxes. The present author's ancillary programs *CALC_RE* and *COMBINE_TABLES* + *CONVERTQ* respectively perform the conversion from dynamic pressure to Reynolds number and interpolation to uniform intervals in the $\log_{10}$(Re) direction. Further details apart from the table formats in section 4 are beyond the scope of this document. However, one issue involving C_D interpolation warrants mention.

At a given Mach number, while C_L vs. α is essentially linear over the middle angle of attack range, C_D is not: C_D typically increases quadratically with Alpha. This means the familiar "trilinear" formulation (a misnomer, because there are cross terms such as *pqr*) errs on the high side for C_D between table entries:

```
            p = (MNUM –  MACH(I_M)) / (MACH (I_M+1) -  MACH(I_M))
            q = (ALFA - ALPHA(J_A)) / (ALPHA(J_A+1) - ALPHA(J_A))
            r = (QBAR –  QBAR(K_Q)) / (QBAR (K_Q+1) -  QBAR(K_Q))

CD_INTERP =
 (1-r) ((1-q) ((1-p) CD(I_M, J_A,  K_Q)   + p CD(I_M+1,J_A,  K_Q))   +
          q   ((1-p) CD(I_M, J_A+1,K_Q)   + p CD(I_M+1,J_A+1,K_Q)))  +
   r   ((1-q) ((1-p) CD(I_M, J_A,  K_Q+1) + p CD(I_M+1,J_A,  K_Q+1)) +
          q   ((1-p) CD(I_M, J_A+1,K_Q+1) + p CD(I_M+1,J_A+1,K_Q+1)))
```

This has the demonstrable effect that, during maximization of cross-range, the angle of attack control points tend to cling to the table entries where the high-drag error goes to zero. *Traj* overcomes this tendency by interpolating drag nonlinearly in the Alpha direction. (It actually uses a weighted combination of the two possible quadratics defined by four points; use of the spline utility it already used for interpolating Alpha and bank vs. time would have been simpler.)

Further Reading: User Guides are available for *NPSOL* and *SNOPT* (see **References**, section 7). For a more thorough discussion of the many aspects of numerical optimization, the reader is urged to consult the excellent text, Practical Optimization (ref. 9), by the *NPSOL* authors.

2. General Information

Input File Summary: Most of the following files are required as input for every *Traj_opt* run. One applies to abort trajectories only; two others apply if a distributed heating constraint is specified; others depend on other constraints. File format details appear in sections 3 and 4.

traj.in	Control file for *Traj*
traj_opt.inp	Control file for *Traj_opt*
traj.alp	Initial control points for the angle of attack schedule, read by *Traj*
traj.bnk	Initial control points for the bank angle schedule, read by *Traj*
.aer	Aerodynamic database as specified in 'traj.in'
thrm_*.bin	Mollier diagram ↔ Planet in traj.in, if stag. pt. calcs. requested
apc.dat	Aerothermal performance constraint data (APC constraint only)
upper_corridor.dat	Upper corridor boundary data (UPPERC constraint only)
aerothermal.database	Distributed heating database (SURF* constraint only)
traj_opt.surface_bounds	Corresponding limits for temperature, heat flux[, …]
traj_opt.ascent	Launch trajectory data (abort calculations only)
traj_opt.steps	Alternative to traj.alp & traj.bnk for aero-capture trajectories

Output File Summary: Most of the following files are produced by every *Traj_opt* run. Two apply only if a distributed heating constraint is specified. File format details appear in section 5.

traj_opt.out	Printable log of the run (all except output from *NPOPT*)
npopt.*	Standard output (redirected to user-specified file) from *NPOPT*
traj_opt.qplot	Plottable form of optimized Alpha and bank control points
traj.out	Printable time history for optimized trajectory
traj.plt	Plottable time history for optimized trajectory
traj_opt.temperatures	Plottable time history of surface point temperatures (SURF* only)
traj_opt.fluxes	Plottable time history of surface point heat fluxes (SURF* only)
traj_opt.steps_opt	Optimized Alpha & bank as step functions (MODE_VAR = 3 or 4)

Derived File Summary: The following files may be derived from the above outputs using ancillary tools. They facilitate interpretation of plotted results.

alpha.opt	Optimized control points in 'traj.alp' and 'traj.bnk' format;
bank.opt	use *GETANGLES* to extract these two files from 'traj_opt.qplot'
subset.plt	Subset of 'traj.plt' (e.g., every 100 seconds); use *REGULARIZE*
traj.qalpha	Time history of Q * Alpha from 'traj.plt'; use *QALPHA*

Standard plotting can be performed readily with *GNUPLOT*, as indicated in section 6. See the web site http://www.ucc.ie/gnuplot/gnuplot.html for *GNUPLOT* information. Ask the *Traj_opt* author (dsaunders@mail.arc.nasa.gov) for a sample plotting script and for the indicated ancillary programs.

Preparing for and Monitoring a *Traj_opt* Run: Recommended practice is to start a new optimization in a clean directory containing the above files (or symbolic links to them, in some cases, to avoid wasteful duplication). Two of the output files will *not* be overwritten, and therefore must be deleted or renamed if they are present prior to a rerun: 'traj_opt.out' and 'traj_opt.qplot'. As more than one run may well be required to solve a particular problem, numbering the outputs is recommended, starting with the redirected standard output file containing (only) *NPOPT* outputs:

% traj_opt > npopt.1 &

Here, 'traj_opt' refers to the executable file, and '>' redirects standard output. That output (mostly from *NPOPT*, the rest being *Traj* diagnostics and final summary) goes to 'npopt.1' here. The '1' sensibly indicates the run number. The '&' sets the job running in the background on a Unix or Unix-like system, to avoid tying up the terminal session. Then,

% tail snopt.1 and tail –60 traj_opt.out or grep SURF1R traj_opt.out, say,

may be used to monitor *NPOPT* progress. Multiple "grep"s can be combined in a "look" script, with redirection and "tail"s, to monitor all relevant constraints and the objective from the current state of 'traj_opt.out', which is flushed at the end of each optimization iteration for this reason (at least on SGI systems):

```
rm obj.out iter.out
grep -F OBJECTIVE traj_opt.out > obj.out
grep -F ITERATION traj_opt.out > iter.out
tail -40 obj.out
tail -3 iter.out
grep -F 'APC        1' traj_opt.out
grep -F 'ENDLAT     1' traj_opt.out
grep -F 'ENDLON     1' traj_opt.out
grep -F 'DYN_PR     1' traj_opt.out
grep -F 'SURF1R     1' traj_opt.out
grep -F acceleration traj_opt.out
grep -F dynamic traj_opt.out
grep -F 'Ascent abort time' traj_opt.out
grep -F 'Shortfall' traj_opt.out
```

Good starting guesses, good scaling, and good gradients are all key to success, along with attainable bounds for the constraints. Issues relevant to the various trajectory cases are discussed next, starting with recommended limits for single-use (abort trajectories) and multi-use (all other trajectories):

	Single-use	Multi-use
Acceleration (total magnitude)	3 G	3 G
Dynamic pressure	800 psf = 38,304 Pa	500 psf = 23,940 Pa

Maximum Cross-Range Trajectories: This extreme case determines how far away from the plane of the orbit the vehicle can turn during a gliding reentry—an indication of its flexibility in terms of landing site and deorbit point in time. As *Traj* does not have a generalized measure of cross-range, *Traj_opt* requires that the orbit be equatorial in order for (a scaled form of) the terminal latitude to serve as the objective function. Inertial entry conditions used at ARC for comparisons with HL-20 are 7.950444 km/sec velocity, -1.2876° entry angle, and 121.92 km entry altitude (all those digits reflecting translation from fps units).

For a due-East orbit, positive bank carries the vehicle into the southern hemisphere. Broadly speaking, bank will vary steadily from 90° to 0°, while angle of attack will be near the $(L/D)_{max}$ value for as long as possible. Typically, Alpha has to be higher for the first half of the trajectory to avoid overheating, then the multi-use dynamic pressure limit may become active, at least for sharp vehicles, with Alpha and L/D optimal for best range except where a canopy temperature constraint inhibits low Alpha.

Less-than-Maximum Cross-Range Trajectories: Given that maximizing cross-range prolongs the surface heating stresses, and is hardly representative of multiple reuse conditions, an alternative form of cross-range calculation would be to minimize the (stagnation point) heat load while achieving a specified cross-range (or some percentage of maximum such as 75%). Studies at ARC show that the total heat load can be roughly halved while achieving three-quarters of the maximum cross-range. More interestingly, the time spent on the APC (or at peak temperature) can be drastically reduced. The following figures should be representative for the 75% case: peak heating time is reduced from ~2,000 seconds to ~300 seconds for a sharp vehicle, and from ~1,150 seconds to ~500 seconds for a blunt vehicle. See **Minimum Heat Load Trajectories** for a little more on this subject.

Down-Range Trajectories: The word Maximum is avoided here, because the problem is then ill-defined: the vehicle tends to stay as high as possible, for more than a whole orbit. Instead, a large down-range should be specified and some other quantity should be optimized, the obvious candidate being (stagnation point) heat load. Even so, an upper-corridor boundary is desirable to help coax the vehicle down and avoid skipping. Large bank reversals are also unavoidable as part of terminating straight ahead, so unique solutions cannot be expected.[*] Down-range constraints suggested are 35,000 km for sharp vehicles (almost a full revolution) and 18,000 km for blunt (about half a revolution).

[*] Use of the available "**total-variation-diminishing**" terms in the objective is recommended for smoother results in the angle of attack and bank angle schedules. This applies to any trajectory where bank reversals are unavoidable, but even cross-range cases can benefit. Entering RHO_ALPHA_TVD = RHO_BANK_TVD = 0.01 is suggested to help isolate smooth solutions from among the many possibilities where bank must oscillate.

Minimum Heat Load Trajectories: These resemble ***minimum-duration*** reentries, because the heat load is roughly proportional to the flight duration. The solution for one might be used to start the other. At ARC, a zero-cross-range constraint has traditionally been specified, for more thorough definition of the problem. In retrospect, it might make more sense not to constrain the cross-range at all because otherwise the large bank reversals that mean ill-conditioning are inevitable. Descending as rapidly as possible requires high angles of attack, but zero banking would eventually mean skipping. Rather, allowing bank to reach ±180° is beneficial, as this permits inverting the lift vector for maximum descent rate early in the trajectory. As with the down-range case, though, terminating cleanly at zero latitude is a tall order for the optimizer. (A tight spiral in the final stages is occasionally obtained, with likely gradient difficulties. Possibly, further definition would help; such as by constraining final heading. This would still be arbitrary, however (0°? 90°? 180°?).

Ascent Abort Trajectories: Studies at ARC have focused on calculating the earliest possible trans-Atlantic emergency landing (TAL) and the latest possible return to an East Coast site. Insufficient overlap in these two times from launch means a window for which ditching in the ocean is an undesirable possibility. RTLS (return-to-launch-site) trajectories have yet to be investigated with *Traj_opt* because the high-fidelity low-speed aerodynamic databases necessary for plausible results require major computational effort and have not been available.

Due East launches from Cape Canaveral and ascents to the International Space Station have both been treated at length in ARC studies of earliest and latest aborts. Cape Verde and San Juan, and Shannon and Gander, are the standard choices for landing sites, respectively. Two ascent trajectories in each case indicate significant effects on the results, but broadly speaking the two TAL cases are similar (with sharp vehicles performing better), while Gander is about equally reachable by sharp and blunt—they behave similarly at high Alpha—and San Juan is not reachable at all by blunt vehicles.

For earliest aborts to Shannon, blunt vehicles with lower glide range have to start later and hence at higher velocities than sharp vehicles. They therefore require higher Alpha initially to slow down more while staying higher. An early spike in the surface heating is followed by a tendency for the canopy temperature to be the active constraint. The dynamic pressure and deceleration limits tend not to be approached. Sharp vehicles experience similar heating while also encountering the Q and G limits. Both types of vehicle experience altitude oscillations. It may be beneficial to stretch the almost-straight-ahead glide by trading kinetic and potential energy. However, ***low-altitude oscillations can occasionally cause grief*** during optimization if a small perturbation causes the termination altitude to be encountered much earlier than previously. No guaranteed way of preventing this situation is known to the author, except perhaps to constrain the terminal flight path angle well away from zero.

In the case of latest time of abort to Gander, passing East of the target then turning back after slowing sufficiently appears to be optimal (see figure 4). This may be facilitated by maximum angle of attack and full inversion (bank = ±180°). The G limit is sorely stressed during the tight turn back. A more direct approach has to begin at lower speed and hence from an earlier abort which is less optimal. If an APC is present, it tends to be touched barely or not at all.

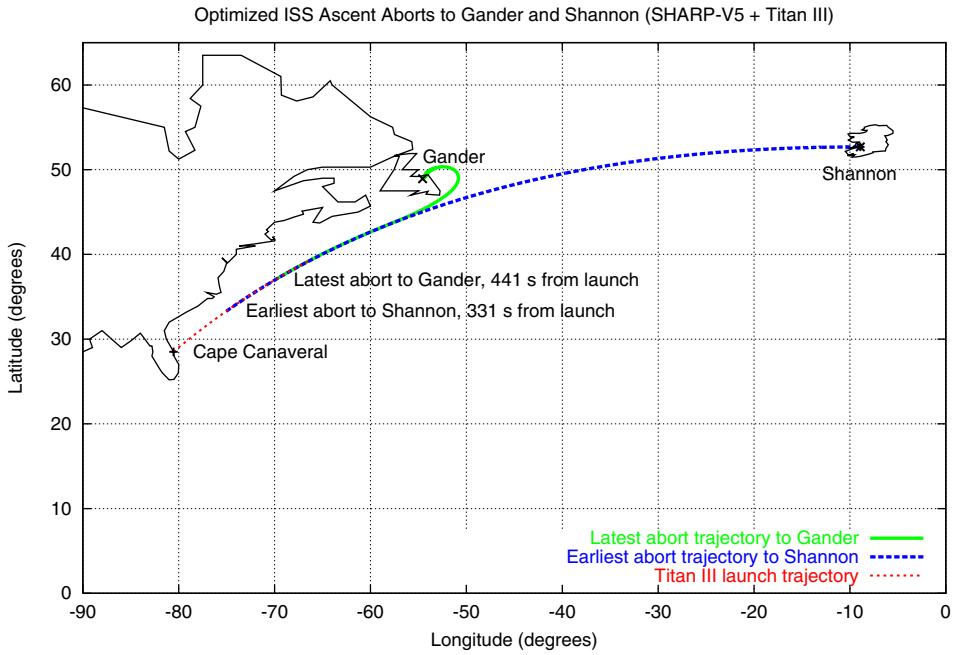


Figure 4. Ascent abort trajectories optimized for Gander and Shannon.

Depending on the ascent trajectory, aborts to San Juan may not be possible even for sharp vehicles without relaxing the standard G and Q limits. If the limits are below what is physically possible, the optimizer struggles greatly. Thus finding higher, feasible limits can be difficult. For spiky violations, small increases in the limit hardly help. Some way of permitting modest violations for short periods would be a blessing. Any ideas are welcomed.

Aero-Capture Trajectories: The on-board fuel needed for an interplanetary vehicle to enter orbit following deceleration and capture by a planet is reduced if the planet's atmosphere can be used to help slow the vehicle down. An optimal trajectory will require a minimum fuel burn to adjust the orbit following aero-capture. An equivalent objective is to minimize the orbit's eccentricity at the target apoapsis. With appropriate initial conditions, and constraints on apoapsis, eccentricity, and orbital inclination, *Traj_opt* can drive *Traj* to perform such optimizations using its "termination at apoapsis" option. Actually, *Traj*'s "Run_to Entry_altitude" option is equivalent and likely to be much more efficient.*

Traj_opt's step function mode for Alpha and bank was introduced for aero-capture applications. Very few steps are intended, in anticipation of closed-loop control schemes where the number of variables to update in real time to account for dispersions should be as small as possible. Allowing the control points to move in time (i.e., variable length steps) appears preferable to adding further control points fixed in time. Further studies of nominal robust aero-capture trajectories at ARC require from *Traj* some means of adding noise to the planetary atmosphere density, in addition to the

* Recent versions of *Traj* itself also have a built-in option to calculate the overshoot/undershoot aero-capture corridor for a specified apoapsis, via safeguarded bisection iterations.

existing scheme implemented at the *Traj_opt* level for perturbing L/D by some percentage. See MODE_VAR = 3 and 4 in section 3, and the 'traj_opt.steps' file specification in section 4 for further details.

Computational Performance: *Traj* performs a trajectory analysis very quickly, even with the recommended half-second cap on ODE time steps and the stagnation point heating calculations turned on. However, even with cheap function evaluations in the 0.3 – 1.5 CPU seconds range, unparallelized finite difference gradient calculations for 80 – 150 variables and 100 – 400 optimization iterations make for run times in the hours on contemporary workstations, such as the SGI Octane (300 MHz). A 2.0 GHz Intel/Linux system using Intel compilers executes *Traj_opt* about 2.25 time faster than this SGI system.

The longer trajectories (maximum cross-range, large down-range) naturally take more time steps and hence run more slowly. Use of 1 s for both 'traj.in' time-step inputs rather than 0.5 s is tempting, and will often succeed for an initial run. Then, restarting with more reliable gradients may or may not be necessary. On the other hand, it may be more convenient to err on the safe side and stay with 0.5 s limits from the start.

If the stagnation point heat load is not really needed, turning it off saves about one-third of the run time. Then, to obtain the reference temperature and heat flux curves on the plots as is commonly preferred using the HL-20 nose radius, the optimal solution can be reanalyzed with NITMAX = 0 and lower resolution to reduce the bulk of final results (2 seconds for OptiStp and H_max is suggested). For abort trajectories, be sure to transfer the optimized and suitably scaled abort time to the TABORT input variable.

3. Control Inputs

traj.in: The *Traj* user guide (ref. 1) should be consulted for details of these control parameters (specified in the *Traj*-specific Trajectory Programming Language (TPL)), and also for the command line switches available to override them. Also, "traj_opt -h" activates *Traj*'s on-line help switch, but be sure that 'traj_opt.out' and 'traj_opt.qplot' are removed first. A sample control file appropriate for an ascent abort to Gander is shown below. Keyword case is significant. Tabs or blanks may be used as separators. The trailing comments are ignored by *Traj*. The indicated entry conditions are, of course, overridden from the ascent trajectory during the optimization.

```
CTV Configuration
PLANET  Earth       Destination planet
Atmosph US_1976     US Standard Atmosphere
Frame   Rotating    Coordinate frame of reference
Gravity Oblate      Gravitational model (dipole, oblate)
Veloc   7.950444    Inertial entry velocity in km/sec
Gamma   -1.2876     Inertial entry angle in degrees
Heading 90.         Entry head angle (psi) in degrees
Altitud 121.92      Entry altitude in kilometers
LatGrph 0.          Entry latitude in degrees
Longtud 0.          Entry longitude in degrees
Step_mx 2500        Limit on number of time steps in stored time history
OptiStp 0.5         [Approximate] resolution of time history saved
H_max   0.5         Maximum Runge-Kutta step size (seconds)
PltType Show_Cm     28-column outputs in the plot file
Run_to  Impact      Termination method
Floor   10.         Altitude at which to terminate (km; 9.144 = 30,000 ft)
Aerodyn Ma_Alpha_logRe External aerodynamic database type …
File    ctv-1d.aero_data  … and file name (must be *.aer[*])
Mass    38551.723   Mass in kg (84,992 lb)
Area    59.7871     Reference area in m^2 (643.543 ft^2)
Length  23.83536    Characteristic length in meters (78.2 ft)
Nose    0.76        Nose radius, meters, for Fay-Riddell (HL-20 reference)
Schedul Fnct_alpha_mono  Control pts are interpolated via monotonic spline
Schedul Fnct_bank_mono
Action  Start       Initialize the aerodynamic model
Action  End
```

The only command line switch other than "-h" likely to be used is "-q", which activates the Fay-Riddell stagnation point heating calculations. Alternatively, use the TPL entries 'Heating Rad_equil_only' and 'Heating No_model' to activate and suppress the heating calculations explicitly. *Traj* presently performs the integrations of convective and radiative heat fluxes by solving additional ODEs, increasing the computational cost of an analysis by more than one third. [Note that all the *Traj* command line switches are available on SGI systems, but other systems lack the necessary Fortran-callable system utilities.]

traj_opt.inp: The main *Traj_opt* control inputs are illustrated in the following example for a latest ascent abort to Gander:

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV  NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
111     55      55      2         CTV ascent abort to Gander
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
2       7       300     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       3       0       100000  2500

-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
2.0     0.1     0.0001  1.E+30  1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.99    10.           1.E-5   -180.   180.    10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
0.          0.          0.        0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
-0.01       0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE          TARGET_TRIM
0.               48.933           -54.567                     0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES -----------------------------------------------
#  VTYPE    V           VSCALE    H          BL       BU
1  TABORT  4.83          100.     0.0000001 4.60     4.95
------- LINEAR CONSTRAINTS ---------------------------------------------
# LCTYPE       BL         BU     TLCON    ILCON
1  ROLL     -10.0      10.0      999.      999
------- NONLINEAR CONSTRAINTS ------------------------------------------
# NLCTYPE      BL         BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC          0.     50.0         0.       1       1     0.1
2  ENDLAT  48.933    48.933         0.       1       1     1.0
3  ENDLON -54.567   -54.567         0.       1       1     1.0
4  ENDFPA    -30.      -2.0         0.       1       1     0.1
5  ACCEL      -3.       0.0         3.       1       1     0.1
6  DYN_PR  -38000.      0.0     38304.       1       1  0.0001
7  SURF1R   -1000.      0.0         0.       1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.01,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
2  UPPERC     -50.      0.0         0.       1       1     0.1
5  ENDBNK     -10.      10.         0.       1       1     1.0
2  OSCILL    -10.       0.0         0.       1       1     1.0
```

Note that some inactive nonlinear constraints are kept handy below the *NPOPTIONS* namelist inputs.

Descriptions for the main *Traj_opt* control inputs are extracted from the source code with minimal reformatting as follows:

```
NDV         Number of "design" (optimization) variables (total);
            alpha variables normally precede bank variables;
            others may follow (see MODE_VAR)
NDV_ALPHA   Number of variables affecting angle of attack vs. time;
            use with "Schedul Fnct_alpha_..." in traj.in;
            if NDV_ALPHA = 0 remove "Schedul Fnct_alpha_*" from traj.in;
            if MODE_VAR = 3 or 4 (step-function variation), enter NSTEPS
NDV_BANK    Number of variables affecting bank angle vs. time;
            use with "Schedul Fnct_bank_..."  in traj.in;
            if NDV_BANK = 0 remove "Schedul Fnct_bank_..." from traj.in;
            if MODE_VAR = 3 or 4 (step-function variation), enter NSTEPS
MODE_VAR    Controls optimization variable choice:
               0 = CLs, CDs of aero. database, for sensitivities only;
                   NDV = 2 * N_MACH * N_ALPHA * N_RE; see also H_ALPHA;
               1 = Alpha and/or bank only (fixed control point times);
                   NDV = NDV_ALPHA + NDV_BANK;
               2 = As for 1 but provides for variable entry conditions,
                   which are required to be inertial;
                   NDV = NDV_(ALPHA + BANK) allows use of ENTRY_DV > 0;
                   NDV = NDV_(ALPHA + BANK) + 1 means an ascent abort;
               3 = Alpha & bank are step functions (fixed durations):
                   alpha & bank are assumed to step at the same times;
                   NDV = 2 * NSTEPS in "traj_opt.steps";
               4 = Alpha & bank are step functions (variable durations);
                   NDV = 3 * NSTEPS in "traj_opt.steps" - 1

            The aero. database dimensions should be outputs from Traj
            initialization, but are instead inputs to Traj_opt:

N_MACH      Dimensions of ...
N_ALPHA     ... the aerodynamic database ...
N_RE        ... which is a rectangular table in (M,AOA,log(Re)) space
PERCENT_LOVERD  Mechanism for checking sensitivities to L/D; enter a
                positive or negative value p in order for the CDs in
                the aero. database to be scaled in a way that changes
                L/D by p%.  E.g., -3.0 lowers L/D by 3%.  Enter 0. to
                suppress any scaling of the drag coefficients.

LINCON      Control for specifying linear constraint TYPES (NPOPT only);
               0 means no linear constraints (NCLIN = 0);
               1 means pitch rate is limited for all alpha control pnts;
                 NCLIN is set to NDV_ALPHA - 1;
               2 means roll rate is limited for all bank control points;
                 NCLIN is set to NDV_BANK - 1;
```

3 means both pitch and roll rate are constrained;
NCLIN is set to NDV_ALPHA + NDV_BANK - 2

NCNLN *Number of nonlinear constraints (NPOPT only)*
NITMAX *Number of optimization iterations, >= 0*
KNOT1_A *First alpha knot to optimize (if NDV_ALPHA > 0)*
KNOT1_B *First bank " " " " (if NDV_BANK > 0)*

NNOPT *Optimizer switch:*
1 = QNMDIF2 (unconstrained [CENDIF2 part only now]);
2 = NPOPT (constrained)
NNSPLINE *Controls splining of control points for alpha and bank;*
Traj interpolates the angle control points it finds in alpha.dat and/or bank.dat at every time step;
NNSPLINE > 0 in traj_opt.inp should match use of "Schedul Fnct_alpha_mono" etc. in the Traj control file, in order for similar interpolations to be indicated by Traj_opt in traj_opt.qplot:
1 = piecewise linear interpolation;
2 = monotonic cubic spline;
3 = non-monotonic ("Bessel") cubic spline
NNGRAD *Controls gradient scheme:*
0 = conventional finite differencing;
1 = adjoint method;
2 = finite differencing, but CONFUN does OBJFUN F & g;
3 = adjoint method, with CONFUN used as for NNGRAD = 2
NNDIFF *Controls finite differencing for objective & constraints:*
2 = 2-point forward differencing with given “h”;
*3 = 3-point central differencing with h * EPSOBJ**(-1/6)*
NNTIME *Controls time step to use from trajectory results:*
0 = last time step;
N > 0 is the Traj journal block time step to use;
N will be bounded by the actual last step
NALLOW *Upper limit on number of trajectory calculations*
MAX_STEP *This should be at least as large as Step_mx in the Traj control file if APC or SURF* constraints are present; it is a limit on the number of uniform steps in the time history maintained by Traj, some items of which are copied in trajectory.c for use by Traj_opt.*

OPTIMIZER INPUTS:

OBJBND *Lower bound on objective function*
ETA *Controls line search's acceptance of a sufficiently lower objective. 0 < ETA < 1.0; try 0.2 for expensive objectives*
OPTOL *Minimum stepsize of QNMDIF2 line search; see reference to optional NPOPT/SNOPT inputs below*
STEPMX *Maximum stepsize of line search; units are obscure; use a smaller value to help avoid a big initial step (but too small can easily make NPOPT think it can't move)*

```
EPSOBJ      Minimum absolute value of a significant difference in OBJ
ZETA        Rescales divisor of STEPMX if OBJ fails in line search
ENTRY_DV    Velocity increment added to entry velocity found in traj.in
            and also to velocities calculated from ascent data if a
            TABORT design variable is present

H_ALPHA     Differencing interval for alpha variables (degrees);
            adjusted larger if central differencing specified (NNDIFF);
            also used for coefficient perturbations, forward & back-
            ward) when estimating aero. sensitivities (MODE_VAR = 0)
BL_ALPHA,   Bounds on alpha spline knots (degrees); note that
BU_ALPHA    alpha and bank variables are scaled for NPOPT purposes
PITCH_RATE  Finite rate used when expanding alpha steps to spline form
H_BANK      Differencing interval for bank variables (degrees)
BL_BANK,    Bounds on bank angle spline knots (degrees)
BU_BANK
ROLL_RATE   Finite rate used when expanding bank steps to spline form

UNITL,      These are two logicals carried along with QNMDIF2, which
QNMPR       is unlikely to be employed any more:
            UNITL controls initial finite diff. gradient information:
               TRUE  = use forward or central differences in QNMDIF2;
                       Hessian is set to I;
               FALSE = use central differences from CENDIF2;
            QNMPR = TRUE means QNMDIF2 prints verbose output

            OBJECTIVE FUNCTION INPUTS:

RHO_xx is a multiplier for the corresponding contribution of xx
to the (possibly composite) objective function being minimized.
Multipliers are also used to scale the objective function to be
O(1) at the minimum.

Be sure to have at least one RHO* input nonzero.

RHO_CROSS_RANGE   [latitude is optimized for now, for equatorial orbit]
                  N.B.:  Bank > 0 rolls the vehicle into the southern
                  hemisphere from the equator.  Use either bank < 0 or
                  RHO_CROSS_RANGE = -1. (but not both) to maximize the
                  northern hemisphere latitude.  RHO * (-CROSS_RANGE)
                  is always the quantity minimized.
RHO_DOWN_RANGE    [-(surface arc) is the quantity minimized]
RHO_HEAT_LOAD     Stagnation point heat load (joules/cm^2)
                  [-(heat load) is minimized; see also RHO_HEATLD]
RHO_MAX_ACCEL     Max. acceleration magnitude in Gs
RHO_MAX_W_TEMP    Max. temperature            deg K  (stagnation point)
RHO_MAX_DYN_PR    Max. dynamic pressure       pascals
RHO_MAX_H_FLUX    Max. total heat flux        watts/cm^2  (stagn. pt.)

RHO_TABORT        If a TABORT variable is present after variable #
```

	NDV_ANGLES, RHO_TABORT > 0 minimizes the ascent abort time for which a target landing site can be reached; RHO_TABORT < 0 maximizes the abort time; use in conjunction with ENDLAT & ENDLON or ENDIST equality constraints, or with RHO_TARGET_POINT > 0
RHO_END_ALTITUDE	*Use this multiplier in conjunction with terminating trajectories at (say) Mach 2, in order to maximize altitude over a target landing site (see TARGET_*).*
RHO_ECCENTRICITY	*Minimizing the orbit eccentricity means minimizing the velocity increment needed to circularize an orbit (later). Use this for aero-capture problems (run to apoapsis), with constraints on apoapsis and orbital inclination.*
RHO_HEATLD	*Integrated heat load using an aerothermal database for multiple surface points with area-based weighting*
RHO_TRIM	*Time-integrated squared deviation of pitching moment (or flap deflection) from TARGET_TRIM.*
RHO_TARGET_POINT	*This multiple of the (squared) distance from target end point (TARGET_LATITUDE, TARGET_LONGITUDE) is minimized; the units are degrees ** 2; the appropriate ending altitude should appear in the traj.in control file via the Floor keyword*
TARGET_LATITUDE	*Target end-of-trajectory pt. (degrees), ...*
TARGET_LONGITUDE	*... if RHO_TARGET_POINT > 0*
TARGET_TRIM	*Desirable bound on |Cm| or |flap deflection|. Use the value 0. if RHO_TRIM > 0; use some small positive value if the corresponding constraint is being used instead (because zero everywhere is unlikely to be feasible).*
RHO_ALPHA_TVD	*Total-variation-diminishing contribution to the objective. Adds a (small) multiple of the sum of (Alpha(i+1) - Alpha(i))**2 / TK_ALPHA(NK_ALPHA), where i is a control pt., not a time history value.*
RHO_BANK_TVD	*Total-variation-diminishing contribution to the objective, as for RHO_ALPHA_TVD but for the bank control pts. Rule of thumb: Use RHO_*_TVD = 1.E-m where m is the decimal place of the total objective function intended to be affected by this contribution.*
RHO_DURATION	*Duration (total time) of the trajectory - probably in conjunction with reasonable constraints on the terminal velocity and flight path angle*

SUPPLEMENTARY OPTIMIZATION VARIABLE INPUTS

If NDV = NDV_ANGLES = NDV_ALPHA + NDV_BANK, meaning no additional variables, just include two header lines in traj_opt.inp. Likewise for MODE_VAR = 3 or 4 (step fns.).

*Ordinal number of design variable added to NDV_ANGLES*

VTYPE *6-character design variable type or name:*

TABORT *Time at which to initiate an ascent abort; if present, a launch trajectory dataset is read from "traj_opt.ascent" in this format, using INERTIAL coordinates:*

```
Title of ascent trajectory dataset
N_ASCENT (# pts. defining the ascent trajectory)
T (sec) V (kps) GAMMA HEADING ALT (km) LAT LONG
T       V       GAMMA HEADING ALT      LAT LONG
T       V       GAMMA HEADING ALT      LAT LONG
:       :       :     :       :        :   :
```

[Allow entry conditions as variables here some day?]

V *Optimization variable value as seen by the optimizer; see VSCALE*

VSCALE *Scale factor for the optimization variable to keep it ~O(1); V * VSCALE is the actual quantity used by Traj_opt & Traj*

AITCH *Step size used for estimating the gradient of the objective w.r.t. the variable by forward differencing (NNGRAD = 0, 2); the actual perturbation used is AITCH * VSCALE; if AITCH < 0 and UNITL = F and ISTART <= 1, an optimal "h" will be estimated by CENDIF2; AITCH is also used for forward derivatives of any nonlinear constraints; see NNDIFF above for central differencing*

BL *Lower bound on the design variable as seen by the optimizer; BL = -999. means the variable has no lower bound, else BL scaling should match that of the variable - e.g., for a variable with VSCALE = 0.1, BL = -10. means the effective values are >= -10.*0.1 = -1., consistent with AITCH usage*

BU *Upper bound on the design variable as seen by the optimizer; BU = 999. means the variable has no upper bound, else BU scaling should match that of the variable*

LINEAR CONSTRAINTS:

If LINCON = 0, just include two header lines.
If LINCON > 0, further linear constraint description lines should be entered as indicated by the LINCON description above.

Now that use of perturbing shape functions for the initial alpha and bank schedules has been eliminated, the only linear constraints are on the alpha knots (to limit pitch rate) and/or the bank knots (to limit roll rate). Rather than requiring a constraint input for each pair of adjacent control points, a single input line here activates each TYPE of linear constraint.
If NDV_ALPHA = 0 but LINCON = 1 or 3, the PITCH inputs here will be ignored; likewise for the NDV_BANK = 0 case.

*Ordinal number of linear constraint type - not used*
LCTYPE *6-character linear constraint type:*

PITCH *Pitch rate applied to alpha control point pairs; both BL and BU are used (see below)*
ROLL *Roll rate applied to bank control point pairs; BL is not used; see BU below*

BL *Lower bound for linear constraint [type]; -999. = -BIGBND; if LCTYPE = PITCH, BL and BU both apply (degrees per second)*
BU *Upper bound for linear constraint [type]; 999. = BIGBND; if LCTYPE = ROLL, BU is the upper bound on the roll rate magnitude in degrees per second for all adjacent pairs of bank control points; BL is ignored*
TLCON *Time at which linear constraint applies (seconds); if LCTYPE = PITCH or ROLL, TLCON is ignored*
ILCON *Integer control for the linear constraint; if LCTYPE = PITCH or ROLL, ILCON is ignored*

NONLINEAR CONSTRAINTS:

If NCNLN = 0, just include two header lines.

*Ordinal number of nonlinear constraint - not used*
NLCTYPE *6-character nonlinear constraint type of name:*

ACCEL *Peak acceleration magnitude; use XNLCON to specify the desired upper bound (Gs); BU should be zero; BL is either zero (for exact peak G) or -XNLCON value if lower is OK, e.g., -3.*
CRANGE *Cross range <degrees latitude for now>*
DRANGE *Down range (surface arc, km)*
DYN_PR *Dynamic pressure; use XNLCON to specify the desired upper bound (pascals, e.g., 23940. for 500 psf); BU should be zero; BL is either zero (for exact peak Q) or -XNLCON value if lower is OK*
ENDALP *Ending alpha (deg)*
ENDALT *Ending altitude (km)*
ENDBNK *Ending bank angle (deg)*
ENDFPA *Ending flight path angle (gamma, degrees)*
ENDHED *Ending heading angle (psi, degrees; 0 = N, 90 = E)*

```
ENDLAT  Ending latitude (deg) ! ENDLAT & ENDLON seem to do
ENDLON  Ending longitude  "   ! better than ENDIST
ENDVEL  Ending velocity (km/sec, relative)
ENDIST  Ending distance from (TARGET_LATITUDE, -_LONGITUDE)
        degrees (not deg ** 2); use zero for both bounds
HEATLD  Integrated heat load over multiple surface points
H_LOAD  Stagnation point heat load (joules/cm^2)
H_FLUX  Peak total heat flux at stagnation pt. (watts/cm^2)
QALPHA  Dynamic pressure * alpha;
        use XNLCON (psf * degrees) for the upper bound;
        15,000 is a likely value; set BU = 0. & BL = -XNLCON
STG_PR  Peak stagnation pressure (pascals)
W_TEMP  Peak wall temperature at stagnation pt. (deg K)

DESCNT  Descent rate at end of trajectory, probably for
        ascent abort trajectories (km/sec)
PK_ALT  Peak altitude (km); use XNLCON for the upper bound
        (km); if this is lower than the entry altitude,
        start measuring from where it first drops below the
        bound; set BU = 0. and BL = -XNLCON

APOAPS  Target apoapsis for aero-capture (km from planet CG)
ECCENT  Ending orbital eccentricity, probably for aero-
        capture
INCLIN  Ending orbital inclination (degrees), probably for
        aero-capture:  0 = equatorial due East; 90 = polar

APC     Aerothermal performance constraint:
        file "apc.dat" should contain points as follows:

           Title
           n      ! # points
           V1  H1 ! High to low, km/sec and km;
           V2  H2 ! Traj_opt reverses the order if necessary
           ::  ::
           Vn  Hn

        The constraint tends to force the minimum of
        "current altitude minus interpolated APC altitude"
        to be >= BL = 0;
        set BU = max. likely altitude difference

UPPERC  Upper corridor constraint analogous to the APC,
        forming an upper boundary in (velocity, altitude)
        space (probably corresponding to Q = 10 psf).
        File "upper_corridor.dat" should match the APC file
        format.  Program CONSTANT_Q is available for
        constructing the initial curve, which may need
        padding via spline interpolation.  The suggested
        choice is 478.8 pascals (10 psf).  The abscissas
```

should be redistributed ~ uniformly to 50 - 100 points in the range ~ 0 - 12 km/sec; set BU = zero and BL ~ -50.

OSCILL An upper bound of 0. means any increase in altitude during reentry is penalized; oscillations may thus be damped; severe skipping may also be overcome. (All altitude increases at each step in the journal history are summed; try scale = 1 & BL = -10.)

SURF1 Distributed heating constraint type 1 (temperature): the total violation of the temperature limits over all surface points is constrained to the upper bound of 0. Use BL ~ -1000. (largest likely undershoot in degrees K). See further description above.

SURF1R Variant of SURF1 using temperatures derived from interpolated heat RATEs. This requires surface emissivities in place of the upper bounds on the heat rates in 'traj_opt.surface_bounds'.

SURF2 Distributed heating constraint type 2 (heat flux): the total violation of heat rate limits over all surface points is constrained to BU = 0.

TRIM The time integral of the square of the instantaneous pitching moment (or possibly flap deflection) deviation from TARGET_TRIM. Choose BL = 0., BU > 0. according to the estimated value of the integral at convergence for the indicated TARGET_TRIM. Use of a RHO_TRIM contribution to the objective is probably better than using this constraint.

BL Lower bound on nonlinear constraint value; -999. = -BIGBND

BU Upper bound on nonlinear constraint value; 999. = BIGBND

N.B.: these bounds should be in the same units as the quantities being bounded, which is sometimes an "area" or "distance", not the intended constraint quantity, for which (as in the case of ACCEL) XNLCON is used to enter the upper limit; the bounds and XNLCON (if used) are entered in real space units; SNLCON will be applied by Traj_opt to the given BL and BU; this is in contrast to the bounds on the optimization variables themselves, which (as for their finite differencing intervals) refer to the scaled variables

XNLCON Real quantity needed (or not) by the nonlinear constraint; used for entering the simple bound imposed on a quantity such as acceleration for which all violations are integrated w.r.t. time and forced to the lower bound of zero

INLCON First index needed (or not) by the nonlinear constraint; [Unused so far.]

JNLCON *Second index needed (or not) by the nonlinear constraint; [Unused so far.]*

SNLCON *Scale factor used to provide the optimizer with nonlinear constraint derivatives comparable to those of the objective and of other constraints - preferably O(1)*

OPTIONAL INPUTS FOR NPOPT/SNOPT:

Traj_opt generates a run-time "specs" file for NPOPT. It also provides a namelist, $NPOPTIONS, for entering further options via traj_opt.inp. Remember to start each line in column 2. Sample:

```
 $NPOPTIONS
 LEVELVER=-1, MAJORPL=1, MINORPL=0, MINORIL=2000, NPRFREQ=100,
 TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2,
 PENPARAM=0., STEPLIM=0.01, TIGHTEN=0.1,
 $END
```

Note: If NNDIFF ‡ 3 and 0. < TIGHTEN < 1., Traj_opt will reset NNDIFF to 3, scale TOLNLIN, TOLOPT, and STEPLIM by TIGHTEN upon return from NPOPT, and call it again for a warm start. Central differencing offers some hope of satisfying the tighter tolerances where forward differencing normally does not. Starting with looser tolerances and 2-point differencing allows switching to more accurate gradients once the problem has been largely solved, and may reduce excessive numbers of major iterations. However, there is no easy way of terminating when the solution is clearly "good enough" for practical purposes. TIGHTEN = 0. (the default) suppresses the warm restart option.

See the NPSOL or SNOPT User Guides for descriptions of the other namelist parameters.

Also: File npopt.specs may be used to override further defaults not addressed by the namelist. This file is optional (need not be present).

4. Other Input Files

Other *Traj_opt* input files are largely self-explanatory. Beginnings and endings suffice to clarify the formats in the order of the summary in section 2.

First, the ***control schedule starting guesses*** are normally entered as spline knots fixed in time. (For aero-capture mode, see ***traj_opt.steps***.) Files ***traj.alp*** and ***traj.bnk*** are read by *Traj*, not *Traj_opt*. File ***traj.in*** should contain the following lines in order to specify use of monotonic splines for interpolating Alpha and bank at every time step:

```
Schedul Fnct_alpha_mono
Schedul Fnct_bank_mono
```

traj.alp

```
Alpha knots
41    ! secs    degrees
0.000000E+00  4.734705E+01
1.750000E+01  4.867465E+01
3.500000E+01  4.752564E+01
5.250000E+01  1.642498E+01
7.000000E+01  4.787833E+01
 :             :
 :             :
6.825000E+02  2.065252E+01
7.000000E+02  4.044869E+01
```

traj.bnk

```
Bank knots
41
0.000000E+00 -1.688803E+02
1.750000E+01 -1.745495E+02
3.500000E+01 -1.667354E+02
5.250000E+01  2.613672E+00
7.000000E+01  1.717165E+02
 :             :
 :             :
6.825000E+02  5.198638E+00
7.000000E+02 -2.478324E+00
```

***.aer*:** The aerodynamic database as specified in 'traj.in' is a rectangular table of lift and drag coefficients as functions of Mach, Alpha, and $\log_{10}$(Reynolds number). Typically, these quantities are generated as functions of Mach, Alpha, and dynamic pressure and must be converted to the form handled by *Traj*. Program *Calc_Re* serves for *HAVOC*-type data; *Combine_tables* and *ConvertQ* serve for *CBAERO*-type data.

Some aerodynamic databases contain the pitching moment in column 6, but trimmed databases can use that column for the body flap deflection (in degrees) needed to achieve trim. Values of ±100. mean that the vehicle could not be trimmed at that condition. Column 7 is not used, but needs to be present. (*Calc_Re* generates it as an altitude, in unspecified units (ft?).)

Traj scans the table till EOF to determine its contents and perform sanity checks. Regrettably, it eschews integer counts at the top of the file. The dimensions of the aerodynamic database are also needed at the *Traj_opt* level, and are thus inputs. *Traj_opt* checks that these dimensions match the line count in the file.

```
[ Mach   Alpha logRe       CL        CD   Deflectn.    Altitude ]
0.100   0.000  5.00  0.020800  0.075915   1.845500  -150000.00
0.500   0.000  5.00  0.020800  0.137485   1.845500   100925.96
0.900   0.000  5.00  0.024400  0.147752   5.758800   181022.02
1.000   0.000  5.00  0.009500  0.160013   5.138000   183980.55
1.200   0.000  5.00 -0.020300  0.119528   3.896300   192929.16
1.900   0.000  5.00  0.065300  0.115235  -5.545700   209598.39
2.000   0.000  5.00  0.054800  0.085596  -6.188100   212613.30
2.850   0.000  5.00 -0.037191  0.046449 -13.465154   230888.22
4.000   0.000  5.00 -0.037283  0.058550 -15.552859   249182.42
6.000   0.000  5.00 -0.038679  0.054969 -17.668387   265407.90
 :       :      :     :         :          :              :
 :       :      :     :         :          :              :
 :       :      :     :         :          :              :
16.000  50.000  9.00  1.310180  1.619549 -52.696587    62967.14
18.000  50.000  9.00  1.297638  1.592064 -42.231969    74025.28
20.000  50.000  9.00  1.274397  1.548599 -50.350940    73332.54
25.200  50.000  9.00  1.298082  1.598240 -47.851683    84550.25
```

thrm_*.bin*:* The Fay-Riddell-type stagnation point heating calculations performed by *Traj* model the equilibrium thermodynamic and transport properties for a gas of arbitrary chemical composition. Computation time is reduced if a Mollier diagram generated ahead of time is interpolated during a trajectory simulation. During initialization, *Traj* looks for such planet-specific data as a disk file in various standard locations starting with the local directory, or generates the file in /tmp if it is not found. Each such file (e.g., 'thrm_earth.bin' for planet Earth) is binary for efficient I/O. The format is otherwise irrelevant. The *Traj_opt* user need only know not to delete this file, to save several minutes of regeneration time at the start of a next run. For Unix[-like] systems, the /usr/local/lib directory is supported by *Traj* for permanent storage. Alternatively, a local symbolic link can point to the file in (say) the *Traj* directory.

apc.dat: The "aerothermal performance constraint" definition is by way of a discrete curve in (velocity, altitude) space. Descending below it means the vehicle is too low/too fast/too hot. The defining file is needed only if an APC constraint is specified. Its format is shown below. The velocities may be descending or ascending—*Traj_opt* will reverse the order to descending if necessary. The units must be kilometers.

```
Single-use UHTC APC (R = 0.01m, 10 deg cone half angle, 0.1 m, 3088 K)
16 ! km/sec      km
10.2766143576948 75
8.61447728364106 70
7.54176520480093 65
6.76513398969863 60
6.17209960639086 55
 :                :
 :                :
3.50569598173243 20
3.21217517970764 15
```

```
3.06845138670501 10
2.97449482561426  5
2.90517697929474  0
```

upper_corridor.dat: This input is analogous to the APC, and used if an UPPERC constraint is specified. It is also of interest on (velocity, altitude) plots even if it is omitted during optimization, as it commonly can be. (Long down-range trajectories need it most, to help prevent skipping.) The upper corridor boundary is commonly defined as the curve corresponding to 10 psf where control surface effectiveness is minimal. Program *CONSTANT_Q* is available from the author to generate such curves.

```
Constant Q:  10 psf = 478.8 pascal
124 ! km/sec      km
9.9734049E+00   8.4067421E+01
9.8898296E+00   8.3966393E+01
9.8062544E+00   8.3864151E+01
9.7226791E+00   8.3760696E+01
:               :
 :               :
1.1153420E-01   2.0916861E+01
9.0640381E-02   1.8267950E+01
6.9746584E-02   1.4932690E+01
4.8852790E-02   1.0232590E+01
2.7959000E-02   0.0000000E+00
```

aerothermal.database: This distributed heating database file is needed if a SURF* constraint is specified. Any number of surface points can be handled as one file of concatenated databases, one database for each point. Any number of surface quantities may be specified for each point (temperature and heat flux being the only two quantities so far). The corresponding upper bounds on temperature, heat flux[, …] are specified via ***traj_opt.surface_bounds***, *q.v.*

```
CTV aerothermal database
18  ! # Mach numbers
10  ! # angles of attack
 5  ! # dyn. Pressures (Pa)
 2  ! # surf. Types (degrees K, J/cm^2,   )
 9  ! # surf. Points
Surface Point  1  :  xyz =    86.147    -1.146   -14.648  (C-C)
Mach   Alpha   Qbar     Temp      Qdot
 0.1   0.0    478.8   285.98   0.03098
 0.5   0.0    478.8   244.76   0.01163
 0.9   0.0    478.8   262.93   0.02439
 1.0   0.0    478.8   268.09   0.02917
 1.2   0.0    478.8   297.93   0.02906
 1.9   0.0    478.8   366.87   0.07186
 2.0   0.0    478.8   307.48   0.04172
 2.8   0.0    478.8   350.87   0.06690
```

```
 4.0   0.0    478.8   367.21   0.08061
 6.0   0.0    478.8   409.04   0.12448
  :     :        :       :      :
  :     :        :       :      :
16.0  50.0  95760.5  2650.21 223.33485
18.0  50.0  95760.5  2807.26 281.12405
20.0  50.0  95760.5  2991.54 362.50589
25.2  50.0  95760.5  3373.04 586.00360
Surface Point  2  :  xyz =   105.290    -1.435   -17.725 (CMC-1)
Mach   Alpha   Qbar     Temp      Qdot
 0.1   0.0    478.8   285.82   0.03095
 0.5   0.0    478.8   241.65   0.01593
 0.9   0.0    478.8   261.37   0.02477
 1.0   0.0    478.8   265.76   0.03013
  :     :        :       :      :
  :     :        :       :      :
```

traj_opt.surface_bounds: Different files are needed for single- and multi-use trajectories. The first column is intended for an area-based weight, in anticipation of performing some type of TPS optimization, but this is a loose end. ***Note:*** The SURF1R constraint expects emissivities, not heat rate limits, in column 3 here (a kludge for this heat-**R**ate-based variant of SURF1).

```
CTV single-use upper bounds for temperature (deg.K) and Qdot (W/cm^2)
 2   9  ! # quantities and # points
  28.4  2755.4 261.03
 391.6  2088.7  86.20
 794.8   977.6   4.54
 981.9  2088.7  86.20
 999.4  1644.3  33.10
1362.6  2755.4 261.03
1362.6  1088.7   6.37
1362.6  2088.7  86.20
1362.6  2755.4 261.03
```

traj_opt.ascent: Ascent abort trajectories require launch trajectory data as a time history in ***inertial*** coordinates, as shown below. These data are interpolated via monotonic splines when TABORT is an optimization variable. Showing the ascent on the latitude/longitude plot of results can use the same data file (at least with *Gnuplot*, which ignores lines such as headers that it cannot read properly).

```
TSTO ascent trajectory to ISS
528 ! T   V km/sec  Gamma       Heading       Alt. km     Geodetic lat    Long
   0.00 0.40905103  0.00000000  90.00000000   0.00609600  28.50000000  -80.60000610
   1.00 0.40906137  0.41800001  89.99900055   0.00758312  28.50000000  -80.60000610
   2.00 0.40909341  0.84299999  89.99800110   0.01208105  28.50000000  -80.60000610
   3.00 0.40914798  1.27699995  89.99600220   0.01964528  28.50000000  -80.60000610
   4.00 0.40922663  1.71800005  89.99500275   0.03033217  28.50000000  -80.60000610
   5.00 0.40933055  2.16700006  89.99400330   0.04419874  28.50000000  -80.60000610
```

```
  6.00 0.40946129  2.62400007  89.99299622   0.06130290  28.50000000  -80.60000610
  7.00 0.40961978  3.08800006  89.99099731   0.08170438  28.50000000  -80.60000610
  8.00 0.40980816  3.56100011  89.98999786   0.10546263  28.50000000  -80.60000610
   :    :           :           :            :           :             :
   :    :           :           :            :           :             :
515.00 7.75100231 -0.14800000  52.37300110 100.89311981  38.50299835  -68.08099365
516.00 7.78039837 -0.11300000  52.40900040 100.89067078  38.54499817  -68.01599121
517.00 7.80999279 -0.07900000  52.44400024 100.89291382  38.58700180  -67.95098877
518.00 7.83978748 -0.04300000  52.47999954 100.89993286  38.62900162  -67.88500977
519.00 7.86978579 -0.00800000  52.51699829 100.91182709  38.67200089  -67.81799316
519.22 7.87630033  0.00000000  52.52500153 100.91504669  38.68099976  -67.80398560
520.00 7.87629175  0.00000000  52.55899811 100.92717743  38.71400070  -67.75201416
```

traj_opt.steps: Aero-capture trajectories may best be simulated with small numbers of Alpha and bank control points that may or may not be fixed in time. The input format is shown below. *Traj_opt* reads a step-function file if MODE_VAR = 3 or 4, and transcribes it to the equivalent ***traj.alp*** and ***traj.bnk*** expected by *Traj* initialization, using PITCH_RATE and ROLL_RATE to simulate finite rates.

```
Alpha & bank steps
  4  !  Alpha          bank   duration (s)
 28.000000E+00 -5.286404E+01  4.292902E+01
 28.000000E+00 -4.645406E+01  3.505116E+01
 28.000000E+00  1.314490E+02  7.209257E+01
 28.000000E+00  8.619661E+01  5.000000E+01
```

5. Output Files

Interpretation of *Traj_opt* output files is outlined below, in the order of the file summary appearing in section 2. An additional group of ***derived*** files is then discussed. As indicated elsewhere, the two files 'traj_opt.out' and 'traj_opt.qplot' are opened as "new" to guard against unintentional overwriting, so they must be removed or renamed prior to another run. The other outputs are opened as "unknown" and hence *will* overwrite the outputs of a previous run in the current directory if they have not been renamed.

traj_opt.out: This printable log starts with an echo of 'traj_opt.inp' and summarizes all optimization steps by tabulating the current (scaled) optimization variables, the first derivatives prior to each line search, and the constraints and trajectory characteristics at the end of each line search (major iteration).

For the first two optimization iterations, *Traj_opt* also tabulates the perturbations of the variables and the corresponding objective function values and trajectory durations calculated during the finite difference gradient estimation. If the durations are not the same or very similar to 4 digits during these perturbations, the finite differencing intervals may be too large. However, the only proper way to gauge gradient accuracy is to plot the derivatives from single-iteration runs using different inputs for the intervals and/or *Traj*'s two time step controls.

Using something like "tail –60 traj_opt.out" during a run normally displays the summary at the end of the current iteration. Capturing it to a summary file at the end of a run is also recommended as a printed record. (The entire 'traj_opt.out' is generally too large to print.) An example is shown below. Asterisks flag (slightly) violated constraints. For APC, the current value > 0 is the minimum altitude (in km) above the constraint curve. Likewise, the negative value shown for DYN_PR indicates how far below the limit the peak dynamic pressure is (in pascals), while the SURF1 value means that the surface point nearest to its temperature limit is approximately 114.5°K below the limit—evidently the 5th surface point. The slight violation shown for ACCEL is a measure of the time history *area* above the acceleration limit specified (3 G here, but such limits for constraints involving quadrature appear as XNLCON in 'traj_opt.inp'). In other words: non-violations show the "nearest *distance* clear of the bound"; violations show the sum of *area* elements above or below the relevant bound.

```
# NLCTYPE  INLCON  CURRENT VALUE    LOWER BOUND    UPPER BOUND  TIME or VELOCITY
    1  APC       1         0.421747       0.000000      50.000000          6.224
    2  ENDLAT    1        48.933000      48.933000      48.933000        749.002  ***
    3  ENDLON    1       -54.567000     -54.567000     -54.567000        749.002  ***
    4  ENDFPA    1       -25.041979     -30.000000      -1.000000        749.002
    5  ACCEL     1         0.000016      -3.000000       0.000000        253.183  ***
    6  DYN_PR    1        -0.859589  -38000.000000       0.000000        443.683
    7  SURF1     1      -114.539032   -1000.000000       0.000000        999.000

 Surface heating summary:

 PT #  HT LOAD, J/CM^2      WEIGHT     PEAK TEMP, K       BOUND  PK QDOT, W/CM^2       BOUND
    1   9.57306654E+03  2.8400E+01     1749.4882987    2755.400       58.7471884     261.030
    2   9.17702322E+03  3.9160E+02     1742.6477493    2088.700       59.5764533      86.200
    3   2.00944785E+02  7.9480E+02      681.1297072     977.600        1.3786922       4.540
```

```
   4   6.64326598E+03  9.8190E+02     1576.3515861    2088.700       45.4288117      86.200
   5   6.31917623E+03  9.9940E+02     1529.7609683    1644.300       41.8578658      33.100
   6   2.10575753E+04  1.3626E+03     2268.0967515    2755.400      137.2745576     261.030
   7   4.79098246E+02  1.3626E+03      770.9078890    1088.700        1.8428515       6.370
   8   1.18816072E+04  1.3626E+03     1812.4101731    2088.700       86.6423837      86.200
   9   2.85067750E+04  1.3626E+03     2447.8282525    2755.400      186.0759279     261.030


 PERFORMANCE PARAMETERS AT OPTIMIZATION ITERATION 132

                                          Initial            Current     Corresponding Times
 Objective function                -4.9017156770E+00  -4.9015711841E+00
 Total time               (sec)     7.4895616195E+02   7.4900186704E+02
 Cross-range (degrees latitude)     4.8933002554E+01   4.8933000056E+01
 Down-range   (surface arc, km)     2.2635102411E+03   2.2636744136E+03
 Terminal altitude         (km)     2.0000000000E+01   2.0000000000E+01
     "    velocity     (km/sec)     2.4076223862E-01   2.4084933303E-01
     "  flight path angle (deg)    -2.5048621021E+01  -2.5041978520E+01
     "  heading angle     (deg)    -1.5122608216E+02  -1.5120302719E+02
 Fay-Riddell heat load (J/cm^2)     1.4306713916E+04   1.4310907406E+04
 Convective heat flux    (peak)     9.9666106865E+01   9.9649691696E+01     169.308     169.183
 Radiative  heat flux  (W/cm^2)     0.0000000000E+00   0.0000000000E+00       0.000       0.000
 Total      heat flux (stag.pt)     9.9666106865E+01   9.9649691696E+01     169.308     169.183
 Convective/ablative       "        7.5240843639E+01   7.5230260882E+01     170.808     170.808
 Radiative /ablative       "        0.0000000000E+00   0.0000000000E+00       0.000       0.000
 Total     /ablative       "        7.5240843639E+01   7.5230260882E+01     170.808     170.808
 Peak temperature   (degrees K)     2.1650228517E+03   2.1649337005E+03     169.308     169.183
   "  acceleration          (G)     3.0051203004E+00   3.0000611387E+00     181.683     252.933
   "  dynamic pressure (pascal)     3.8308282673E+04   3.8303290535E+04     443.933     443.933
   "  stagnation pr.       "        7.2700040367E+04   7.2679422242E+04     488.145     488.145
 Shortfall from target   (n mi)           0.0001650          0.0000042
 Ascent abort time        (sec)         490.171568         490.157118
 Integrated heat load  (J/cm^2)       101232610.06       101242796.88

 Terminal (latitude, longitude)         48.93300006       -54.56699996

 SOLVE: CPU secs. for this run so far:      5054.38

 Total # trajectory calculations: 16009

 Normal termination.

 Traj_opt: CPU secs. for this run:      5054.46
```

The ***Fay-Riddell*** heating figures refer to the nose radius indicated in ‘traj.in’ (commonly the HL-20 reference of 0.76 m), and negligible radiative heating is typical for Earth entry. The *“**Integrated heat load**”* figures are not meaningful yet, for lack of good area-based weights associated with the distributed heating at the key surface points.

npopt.*: This is the standard output from *NPOPT* as redirected to a user-specified file. Run number is recommended as the extension. With MAJORPL = 1 and MINORPL = 0, one line is printed per iteration following display of the optimizer set-up parameters. The following is a typical outcome with forward differencing:

```
.....................................................................................
Itns Major Minors    Step  nCon Feasible  Optimal  MeritFunction  L+U BSwap nS  condHz Penalty
 369    80      2 7.2E-02    83  4.7E-04  5.7E-02 -4.9006933E+00   73       72 6.5E+00 6.1E+01 _
 372    81      3 7.5E-02    84  4.4E-04  5.2E-02 -4.9006618E+00   72       72 4.5E+00 6.4E+01 _
```

```
 373    82      1 6.5E-02    85  4.1E-04  4.4E-02 -4.9006294E+00   73     1 72 9.4E+00 6.9E+01 _
 379    83      6 8.3E-01    86  1.1E-04  2.4E-02 -4.9009646E+00   70       75 6.2E+00 6.9E+01 _
 380    84      1 4.9E-03    88  1.0E-04  2.4E-02 -4.9009592E+00   66       75 6.2E+00 9.9E+01 _n
Search exit  6 -- no minimizer.         Itn =     84  Dual Inf =  2.403E-02

EXIT -- the current point cannot be improved
```

Here, the line search has been unsuccessful. The nonlinear constraints have essentially been satisfied to the 1.E-4 tolerance, but as is usually the case, the test for optimality has not been met. Use of the TIGHTEN = 0.1 option (with an automatic switch to central differencing) produced the following final outcome in another 44 iterations:

```
.....................................................................................
Itns Major Minors    Step  nCon Feasible  Optimal  MeritFunction  L+U BSwap nS  condHz Penalty
 174    40      3 7.1E-01    41 (7.0E-08) 5.8E-03 -4.9015662E+00   65       79 1.3E+02 1.7E+03 _
 178    41      4 1.3E-01    42 (6.1E-08) 1.8E-02 -4.9015658E+00   66     2 77 2.0E+02 9.1E+05 _
 184    42      6 7.3E-02    43 (5.6E-08) 1.1E-02 -4.9015664E+00   68       77 9.0E+01 3.6E+04 _
 185    43      1 1.0E+00    44 (9.3E-10) 4.2E-03 -4.9015666E+00   66       77 5.0E+01 4.4E+06 _
 187    44      2 1.0E+00    46 (4.8E-09) 8.4E-03 -4.9015688E+00   66       76 8.4E+01 1.6E+05 _m

EXIT -- optimal, but the requested accuracy could not be achieved
```

traj_opt.qplot: This plottable form of the current Alpha and bank control points is updated at the end of every optimization iteration. It allows display of the control schedules during a run, but depends on program *QPLOT*, an ARC-developed utility, the portability of which is unfortunately hindered by the need for a *CA-DISSPLA* graphics library license. A simple *GETANGLES* program easily extracts ‘alpha.opt’ and ‘bank.opt’ from ‘traj_opt.qplot’ in the same format as ‘traj.alp’ and ‘traj.bnk’.

traj.out: The *Traj* package generates this printable time history when it is invoked at the end of a *Traj_opt* run with outputs activated. Generally, ‘traj.plt’ is more useful. Both represent the stored time history. Note the half-second resolution, which is not necessarily at exact 0.5 intervals for reasons discussed in section 1—see ***Terminating the Optimization***.

```
.....................................................................
  Time    Range   Velocity Altitude  Lat.    Long. Mach Accel  Gamma Orbit 42
  sec.     km      km/sec     km     deg.     deg. Num.   g     deg. Info.
 -----  ------- -------- -------- ------ ------ ---- -----  ----- -----
 739.0  2261.52   0.232    20.95   48.95 -54.55  0.8   0.7 -21.54  rESR
 739.5  2261.62   0.232    20.91   48.95 -54.55  0.8   0.7 -21.83  rESR
 740.0  2261.73   0.233    20.86   48.95 -54.55  0.8   0.7 -22.11  rESR
 740.5  2261.84   0.233    20.82   48.95 -54.55  0.8   0.8 -22.38  rESR
 741.0  2261.95   0.234    20.78   48.95 -54.56  0.8   0.8 -22.64  rESR
 741.5  2262.05   0.234    20.73   48.95 -54.56  0.8   0.8 -22.89  rESR
 742.0  2262.16   0.235    20.68   48.94 -54.56  0.8   0.8 -23.12  rESR
 742.5  2262.27   0.235    20.64   48.94 -54.56  0.8   0.8 -23.34  rESR
 743.0  2262.38   0.236    20.59   48.94 -54.56  0.8   0.8 -23.54  rESR
 743.5  2262.48   0.236    20.54   48.94 -54.56  0.8   0.8 -23.74  rESR
 744.0  2262.59   0.237    20.50   48.94 -54.56  0.8   0.8 -23.92  rESR
 744.5  2262.70   0.237    20.45   48.94 -54.56  0.8   0.8 -24.09  rESR
 745.0  2262.81   0.237    20.40   48.94 -54.56  0.8   0.8 -24.24  rESR
```

```
745.5  2262.92   0.238    20.35   48.94 -54.56  0.8   0.9 -24.39  rESR
746.0  2263.02   0.238    20.30   48.94 -54.56  0.8   0.9 -24.52  rESR
746.5  2263.13   0.239    20.25   48.94 -54.56  0.8   0.9 -24.64  rESR
747.0  2263.24   0.239    20.20   48.94 -54.56  0.8   0.9 -24.74  rESR
747.5  2263.35   0.240    20.15   48.94 -54.56  0.8   0.9 -24.84  rESR
748.0  2263.46   0.240    20.10   48.93 -54.57  0.8   0.9 -24.92  rESR
748.5  2263.57   0.240    20.05   48.93 -54.57  0.8   0.9 -24.99  rESR
749.0  2263.67   0.241    20.00   48.93 -54.57  0.8   0.9 -25.04  rESR
```

traj.plt: With TPL command "PltType Show_Cm" in 'traj.in', a 28-column file is written at the end of each *Traj_opt* run. Stagnation point heating calculations need to have been invoked for certain columns to be meaningful. *Gnuplot* is well suited to script-driven generation of *.ps, *.eps or *.gif files containing multiple plot pages. Some of these plots benefit from a subset of the traj.plt data (say every 100 seconds) shown as symbols. See the derived files below.

The obscure order of the columns in 'traj.plt' is fixed by *Traj* documentation, although non-standard additions have been made at the end of the list—hence the "Show_Cm" specification, which was originally used to plot pitching moment but now tends to be used for the body flap deflection needed to trim the vehicle during generation of the aerodynamic coefficients. Here, ±100° means trimming was not possible.

The columns of 'traj.plt' for the "Show_Cm" case are defined as follows:

1 time, seconds (~uniform steps for time history quadratures and plotting purposes)
2 ballistic coefficient, kg/m^2
3 L/D
4 C_D
5 C_L
6 dynamic pressure, pascals
7 |acceleration|, G
8 Reynolds number based on L_{ref}
9 Mach number (free stream)
10 altitude, km
11 |velocity|, km/sec
12 range, km (arc along surface)
13 longitude, degrees
14 boundary layer edge temperature, K
15 wall temperature, K (Fay-Riddell stagnation point calculations)
16 heat flux (convective), $watts/cm^2$ (stagnation point)
17 heat flux (radiative)
18 heat flux (total)
19 heat load, $joules/cm^2$ (stagnation point)

20 wall pressure, pascals
21 shock layer ratio of specific heats
22 free stream density, kg/m^3
23 heading, degrees (0 = due N, 90 = due E)
24 geocentric latitude, degrees
25 angle of attack, degrees
26 bank angle, degrees (> 0 = right wing down)
27 flap deflection, degrees (> 0 = down), or possibly C_M
28 flight path angle, degrees (> 0 = up)

traj_opt.temperatures: If a SURF* constraint is specified, this file is written by *Traj_opt* (not *Traj*) at the end of a run as a plottable time history of surface point temperatures in °K. Column 1 contains the stored time steps as for 'traj.plt'; remaining columns number as many as the number of surface points in 'aerothermal.database' and 'traj_opt.surface_bounds':

```
0.7  1194.36  1166.83   272.12  1002.36   959.15  1702.53   515.50  1187.38  1848.65
1.2  1194.50  1166.97   272.12  1002.60   959.39  1703.25   515.54  1187.99  1848.96
1.7  1194.65  1167.11   272.13  1002.84   959.63  1703.95   515.58  1188.60  1849.27
2.2  1194.79  1167.26   272.13  1003.08   959.87  1704.65   515.62  1189.19  1849.59
2.7  1194.94  1167.41   272.13  1003.32   960.11  1705.35   515.66  1189.78  1849.91
3.2  1195.10  1167.56   272.14  1003.56   960.34  1706.03   515.71  1190.36  1850.23
3.7  1195.25  1167.71   272.14  1003.79   960.58  1706.71   515.75  1190.93  1850.55
4.2  1195.41  1167.87   272.15  1004.03   960.81  1707.37   515.79  1191.49  1850.88
4.7  1195.57  1168.03   272.15  1004.26   961.04  1708.03   515.83  1192.04  1851.21
.....................................................................................
```

traj_opt.fluxes: If a SURF* constraint is specified and there are at least 2 quantities per surface point in the distributed heating database, this file is written by *Traj_opt* at the end of a run as a plottable history of surface point heat fluxes, analogous to 'traj_opt.temperatures'. The units are joules/cm^2:

```
0.7   7.502   6.832   0.015   3.793   3.185  33.005   0.248   7.731  45.045
1.2   7.504   6.834   0.015   3.796   3.188  33.058   0.248   7.746  45.061
1.7   7.505   6.836   0.015   3.799   3.190  33.111   0.248   7.761  45.077
2.2   7.508   6.838   0.015   3.802   3.193  33.163   0.247   7.776  45.094
2.7   7.510   6.840   0.015   3.805   3.195  33.214   0.247   7.791  45.112
3.2   7.512   6.842   0.015   3.807   3.198  33.265   0.247   7.805  45.131
3.7   7.514   6.844   0.015   3.810   3.200  33.316   0.247   7.819  45.150
4.2   7.517   6.847   0.015   3.813   3.203  33.366   0.247   7.833  45.169
4.7   7.520   6.849   0.015   3.816   3.205  33.415   0.247   7.847  45.189
5.2   7.522   6.852   0.015   3.819   3.208  33.463   0.247   7.860  45.210
.....................................................................
```

traj_opt.steps_opt: This file is updated after each optimization iteration, in the same format as 'traj_opt.steps', *q.v.*, if MODE_VAR = 3 or 4 (step-function mode for aero-capture).

Derived Results

alpha.opt & **bank.opt:** *Traj_opt* writes the current control points in *QPLOT*able form at the end of each optimization iteration, as ‘traj_opt.qplot’. To derive the solution in ‘traj.alp’ and ‘traj.bnk’ format, program *GETANGLES* is available from the author. It generates ‘alpha.opt’ and ‘bank.opt’ suitable for starting a new run or for imposing on the time history plot of Alpha and bank via *Gnuplot* or equivalent. (*QPLOT* is an ARC-developed plotting program employing the *CA-DISSPLA* graphics library, for which a license is required from Computer Associates.)

subset.plt: Interpretation of plots where time is *not* the abscissa is facilitated by superimposing symbols at (say) 100-second intervals. Spline interpolation of all columns of a time history such as ‘traj.plt’, at any specified interval, can be performed with program *REGULARIZE*, available from the author.

traj.qalpha: The quantity Q * Alpha (dynamic pressure in psf times angle of attack in degrees) is meaningful to the missile community, as a measure of bending loads. *Traj* does not calculate it, and dynamic pressure is not always included in the portion of the time history transferred from *Traj* to *Traj_opt*. Therefore, for plotting purposes, this quantity can be derived from ‘traj.plt’ as ‘traj.qalpha’ using program *QALPHA*, available from the author.

6. Displaying Results

At ARC, rapid display of results is performed with the aid of two scripts. The first generates derived results by tying together some ancillary programs (with appropriate path names or aliases); the second is passed to *Gnuplot* following update of the title to be shown on each plot page.

(1) **plt** script:

```
/codes/aero/traj_opt/get_angles
/codes/numerics/single/regularize/regularize << HERE
traj.plt
s
100
HERE
mv s subset.plt
/codes/aero/traj_opt/qalpha << THERE
traj.plt
t
THERE
mv t traj.qalpha
```

(2) **traj.gnu** script:

```
set title 'CTV-XX:  Ascent Abort to Gander; Tabort Maximized to 490 seconds    08/16/02'
set term postscript color
#
#set term post eps color solid "Helvetica" 28
#
set out 'ctv.ps'
set data style lines
set grid
#
set xlabel 'Time (seconds)'
set ylabel 'Surface temperature (degrees K)'
set key right
plot 'traj.plt'             using 1:15 title 'Fay-Riddell  (0.76 m radius)' with lines lw 5,\
     'traj_opt.temperatures' using 1:2  title '1: Center,  X ~  86",    C-C' with lines lw 3,\
     'traj_opt.temperatures' using 1:3  title '2: Center,  X ~ 105",  CMC-1' with lines lw 3,\
     'traj_opt.temperatures' using 1:4  title '3: Canopy,  X ~ 182", Quartz' with lines lw 3,\
     'traj_opt.temperatures' using 1:5  title '4: Center,  X ~ 392",  CMC-1' with lines lw 3,\
     'traj_opt.temperatures' using 1:6  title '5: Hip,     X ~ 564",  CMC-2' with lines lt -1 lw 3,\
     'traj_opt.temperatures' using 1:7  title '6: Wing LE, X ~ 434",    C-C' with lines lw 3,\
     'traj_opt.temperatures' using 1:8  title '7: U. wing, X ~ 667",     Ti' with lines lw 3,\
     'traj_opt.temperatures' using 1:9  title '8: L. wing, X ~ 667",  CMC-1' with lines lw 3,\
     'traj_opt.temperatures' using 1:10 title '9: Wing LE, X ~ 608",    C-C' with lines lw 3
#
```

```
set ylabel 'Surface heat flux (watts/cm^2)'
set key right
plot 'traj.plt'              using 1:18 title 'Fay-Riddell  (0.76 m radius)' with lines lw 5,\
     'traj_opt.fluxes' using 1:2  title '1: Center,  X ~  86",    C-C' with lines lw 3,\
     'traj_opt.fluxes' using 1:3  title '2: Center,  X ~ 105",  CMC-1' with lines lw 3,\
.....
.....
     'traj_opt.fluxes' using 1:9  title '8: L. wing, X ~ 667",  CMC-1' with lines lw 3,\
     'traj_opt.fluxes' using 1:10 title '9: Wing LE, X ~ 608",    C-C' with lines lw 3
#
set ylabel 'Angle (degrees)'
set key right
plot 'traj.plt' using 1:25 title 'Angle of attack' with lines lw 2,\
     'alpha.opt' using 1:2 title 'Alpha control points' with points lw 2,\
     'traj.plt' using 1:26 title 'Bank angle' with lines lw 2,\
     'bank.opt' using 1:2 title 'Bank control points' with points lw 2
#
set xlabel 'Velocity (km/sec)'
set ylabel 'Altitude (km)'
set key left
plot 'traj.plt' using 11:10 title 'Altitude vs. velocity' with lines lw 3,\
     'subset.plt' using 11:10 title '100-second intervals' with points lw 2,\
     'apc.dat' using 1:2 title 'Single-use APC (UHTC, cone, 10 mm)' with lines lw 2,\
     'upper_corridor.dat' using 1:2 title 'Dynamic pressure = 10 psf' with lines lw 2
#
set xlabel 'Time (seconds)'
set ylabel 'Altitude (km)'
set key right
plot 'traj.plt' using 1:10 title 'Altitude' with lines lw 3
#
set ylabel 'Velocity (km/sec)'
set key right
plot 'traj.plt' using 1:11 title 'Velocity' with lines lw 3
#
set ylabel 'L/D'
set key left
plot 'traj.plt' using 1:3 title 'L/D' with lines lw 3
#
set ylabel 'Dynamic pressure (pascals)'
set key right
plot 'traj.plt' using 1:6 title 'Dynamic pressure' with lines lw 3
#
set xlabel 'Time (seconds)'
set ylabel 'Dynamic Pressure * Alpha (psf-degrees)'
plot 'traj.qalpha' using 1:2 title 'Q * Alpha' with lines lw 3
#
set ylabel '|Acceleration| (G)'
set key right
plot 'traj.plt' using 1:7 title '|Acceleration|' with lines lw 3
#
set ylabel 'Mach number'
set key right
plot 'traj.plt' using 1:9 title 'Mach number' with lines lw 3
#
set ylabel 'Heading (degrees)'
set key right
plot 'traj.plt' using 1:23 title 'Heading' with lines lw 3
#
set ylabel 'Flight path angle (degrees)'
set key right
plot 'traj.plt' using 1:28 title 'Flight path angle' with lines lw 3
#
set ylabel 'Heat load (joules/cm^2)'
```

```
set key left
plot 'traj.plt' using 1:19 title 'Fay-Riddell heat load (0.76 m nose radius)' with lines lw 3
#
set ylabel 'Body flap deflection (degrees)'
set key left
plot 'traj.plt' using 1:27 title 'Body flap deflection' with lines lw 3
#
set xrange [-84:-51]
set xtics -84,2,-51
set yrange [28:50]
set ytics 28,2,50
#
set xlabel 'Longitude (degrees)'
set ylabel 'Latitude (degrees)'
set key left
plot 'traj.plt' using 13:24 title 'Latitude vs. longitude' with lines lw 3,\
     'subset.plt' using 13:24 title '100-second intervals' with points lw 2,\
     'traj_opt.ascent' using 7:6 title 'ISS launch trajectory' with lines lw 3,\
     'launch.point.cape-canaveral' using 2:1 title 'Cape Canaveral launch site' with points lw 3,\
     'target.point.gander' using 2:1 title 'Gander landing site' with points lw 3
```

7. References

1. Allen, Jr., G. A.; Wright, M. J.; and Gage, P. J.: The Trajectory Program (Traj): Reference Manual and User Guide. NASA TM-2004-212847, March 2005.

2. Gill, P. E.; Murray, W.; Saunders, M. A.; and Wright, M. H.: User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming. Technical Report SOL 86-2, Department of Operations Research, Stanford University, Stanford, Calif., 1986.

3. Saunders, D. A.; Allen, Jr., G. A.; Gage, P. J.; and Reuther, J. J.: Crew Transfer Vehicle Trajectory Optimization. AIAA Paper 2001-2885, 35th AIAA Thermophysics Conference, Anaheim, Calif., 2001.

4. Hargraves, C. R.; and Paris, S. W.: Direct Trajectory Optimization Using Nonlinear Programming and Collocation. J. Guidance, Control and Dynamics, vol. 10, no. 4, 1987.

5. Gill, P. E.; Murray, W.; and Saunders, M. A.: User's Guide for SNOPT 5.3: A Fortran Package for Large-scale Nonlinear Programming. Technical Report SOL 98-1, Department of Engineering Economic Systems & Operations Research, Stanford University, Stanford, Calif., 1998.

6. Kinney, D.; Bowles, J.; Yang, L.; and Roberts, C.: Conceptual Design of a SHARP Crew Transfer Vehicle. AIAA Paper 2001-2887, 35th AIAA Thermophysics Conference, Anaheim, Calif., 2001.

7. Kinney, D.: CBAERO v1.3 User's Guide, August 2002 (unpublished).

8. Prabhu, Dinesh: A Preliminary Assessment of CBAERO v1.3. ELORET Report ASA-01.ST-B.2.1, September 2002.

9. Gill, P. E.; Murray, W.; and Wright, M. H.: Practical Optimization. Academic Press, 1986.

Appendix A: Useful Coordinates

	Latitude (deg)	Longitude (deg)
Cape Canaveral	*28.50*	*-80.55*
Antigua	*17.133*	*-61.783*
Cape Verde	*16.73*	*-22.93*
Gander	*48.933*	*-54.567*
Goose Bay	*53.33*	*-60.42*
Oceana NAS	*36.8419*	*-76.0133*
San Juan	*18.408*	*-66.064*
Shannon	*52.7*	*-8.917*
St. Johns	*47.08*	*-52.71*

Appendix B: Maximum Cross-Range Controls

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV  NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
152     76      76      1         CTV maximum cross-range
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
0       6       400     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       2       0       100000  9000
-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
0.0     0.1     0.0001  1.E+30  1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.95    10.           1.E-5   0.      180.    10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
10.         0.          0.        0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
 0.         0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE             TARGET_TRIM
0.               0.               0.                           0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES ----------------------------------------------
#  VTYPE        V      VSCALE       H       BL      BU
------- LINEAR CONSTRAINTS ----------------------------------------------
# LCTYPE       BL        BU     TLCON    ILCON
------- NONLINEAR CONSTRAINTS -------------------------------------------
# NLCTYPE      BL        BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC         0.      50.0         0.       1       1     0.1
2  UPPERC    -50.       0.0         0.       1       1     0.1
3  ENDFPA    -30.      -2.0         0.       1       1     1.0
4  ENDVEL   0.15        0.4         0.       1       1     1.0
5  DYN_PR -20000.       0.0     23940.       1       1  0.0001
6  SURF1R  -1000.       0.0         0.       1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.01,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
```

Appendix C: Large Down-Range Controls

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV  NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
122     61      61      1         CTV large down-range
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
2       8       400     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       2       0       100000  12000
-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
0.0     0.1     0.0001  1.E+30  1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.95    10.           1.E-5   -180.   180.    10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
0.          0.          1.E-5     0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
 0.         0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE             TARGET_TRIM
0.               0.               0.                           0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES ----------------------------------------------
#  VTYPE        V      VSCALE       H       BL      BU
------- LINEAR CONSTRAINTS ----------------------------------------------
# LCTYPE       BL        BU     TLCON    ILCON
1  ROLL      -10.       10.       1.        1
------- NONLINEAR CONSTRAINTS ------------------------------------------
# NLCTYPE      BL        BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC         0.      50.0        0.        1       1     0.1
2  UPPERC    -50.       0.0       0.        1       1     0.1
3  DRANGE  30000.    30000.       0.        1       1  0.0001
4  ENDLAT      0.       0.0       0.        1       1     1.0
5  ENDFPA    -30.      -2.0       0.        1       1     1.0
6  ENDVEL   0.15       0.4       0.        1       1     1.0
7  DYN_PR -20000.       0.0    23940.       1       1  0.0001
8  SURF1R  -1000.       0.0       0.        1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.1,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
```

Appendix D: Minimum Heat Load Controls

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV  NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
102     51      51      1         CTV minimum heat load
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
2       7       400     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       2       0       100000  4000
-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
0.0     0.1     0.0001  1.E+30  1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.99    10.           1.E-5   -180.   180.    10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
0.          0.          0.0001    0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
 0.         0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE             TARGET_TRIM
0.               0.               0.                           0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES ----------------------------------------------
#  VTYPE        V      VSCALE       H       BL      BU
------- LINEAR CONSTRAINTS ----------------------------------------------
# LCTYPE       BL        BU     TLCON    ILCON
1  ROLL      -10.       10.       1.        1
------- NONLINEAR CONSTRAINTS ------------------------------------------
# NLCTYPE      BL        BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC         0.      50.0        0.        1       1     0.1
2  ENDLAT      0.       0.0        0.        1       1     1.0
3  ENDFPA    -30.      -2.0        0.        1       1     1.0
4  ENDVEL   0.15       0.4        0.        1       1     1.0
5  ACCEL      -3.      0.0       3.         1       1     0.1
6  DYN_PR -20000.      0.0    23940.        1       1  0.0001
7  SURF1R  -1000.      0.0        0.        1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.01,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
```

Appendix E: Latest Abort to Gander Controls

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV   NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
111     55      55      2         CTV latest ascent abort to Gander
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
2       7       400     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       3       0       100000  2500
-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
2.0     0.1     0.0001  1.E+30  1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.99    10.           1.E-5   -180.   180.    10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
0.          0.          0.        0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
-0.01       0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE            TARGET_TRIM
0.               48.933           -54.567                     0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES ----------------------------------------------
#  VTYPE    V           VSCALE    H          BL       BU
1  TABORT  4.83          100.     0.0000001 4.60     4.95
------- LINEAR CONSTRAINTS ----------------------------------------------
# LCTYPE       BL        BU     TLCON    ILCON
1  ROLL     -10.0      10.0      999.      999
------- NONLINEAR CONSTRAINTS -------------------------------------------
# NLCTYPE      BL        BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC          0.     50.0         0.       1       1     0.1
2  ENDLAT  48.933    48.933         0.       1       1     1.0
3  ENDLON -54.567   -54.567         0.       1       1     1.0
4  ENDFPA    -30.      -2.0         0.       1       1     0.1
5  ACCEL      -3.       0.0         3.       1       1     0.1
6  DYN_PR  -38000.      0.0     38304.       1       1  0.0001
7  SURF1R   -1000.      0.0         0.       1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.005,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
```

Appendix F: Earliest Abort to Shannon Controls

```
-------- MAJOR TRAJ_OPT OPTIONS ----------------------------------------
NDV  NDV_ALPHA NDV_BANK MODE_VAR  VEHICLE (Text for plottable outputs)
83      41      41      2         CTV earliest ascent abort to Shannon
N_MACH  N_ALPHA N_RE    PERCENT_L/D
18      10      9       0.
LINCON  NCNLN   NITMAX  KNOT1_A KNOT1_B
0       7       400     1       1
NNOPT  NNSPLINE NNGRAD  NNDIFF  NNTIME  NALLOW  MAX_STEP
2       2       2       2       0       100000  6000
-------- OPTIMIZER INPUTS -----------------------------------------------
OBJBND  ETA     OPTOL   STEPMX  EPSOBJ  ZETA    ENTRY_DV
2.0     0.1     0.0001   1.E+30 1.E-13  10.     0.
H_ALPHA BL_ALPHA BU_ALPHA PITCH_RATE    H_BANK  BL_BANK BU_BANK ROLL_RATE
1.E-6   5.       49.95    10.           1.E-5   -90.    90.     10.
UNITL   QNMPR
T       T
RHO_C_RANGE RHO_D_RANGE HEAT_LOAD MAX_ACCEL MAX_TEMP MAX_DYN_PR MAX_H_FLUX
0.          0.          0.        0.        0.       0.         0.
RHO_TABORT  RHO_END_ALTITUDE  RHO_ECCENTRICITY  RHO_HEATLD      RHO_TRIM
0.01        0.                0.                0.              0.
RHO_TARGET_POINT TARGET_LATITUDE  TARGET_LONGITUDE             TARGET_TRIM
0.               52.7             -8.917                       0.
RHO_ALPHA_TVD    RHO_BANK_TVD     RHO_DURATION
0.01             0.01             0.
-------- DESIGN VARIABLES ----------------------------------------------
#  VTYPE    V           VSCALE    H         BL       BU
1  TABORT  4.20         100.      1.E-7     3.80     4.60
------- LINEAR CONSTRAINTS ----------------------------------------------
# LCTYPE       BL         BU     TLCON    ILCON
------- NONLINEAR CONSTRAINTS -------------------------------------------
# NLCTYPE      BL         BU    XNLCON   INLCON  JNLCON  SNLCON
1  APC          0.     50.0         0.       1       1     0.1
2  ENDLAT    52.7      52.7         0.       1       1     1.0
3  ENDLON  -8.917    -8.917         0.       1       1     1.0
4  ENDFPA    -30.      -1.          0.       1       1     1.0
5  ACCEL      -3.       0.          3.       1       1    0.01
6  DYN_PR  -38000.      0.0     38304.       1       1  0.0001
7  SURF1R   -1000.      0.0         0.       1       1   0.001
 $NPOPTIONS TOLLIN=1.E-6, TOLNLIN=1.E-4, TOLOPT=1.E-2, STEPLIM=0.01,
 TIGHTEN=0., MAJORPL=1, MINORPL=0, $END
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
03/2005	Technical Memorandum	

4. TITLE AND SUBTITLE
Traj_opt User's Guide

5a. CONTRACT NUMBER
NAS2-99092

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
David A. Saunders

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ELORET
Sunnyvale, California and
Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER
A-0513375

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA 94035-1000

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2005-213449

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited Distribution: Nonstandard
Subject Category: 18, 66
Availability: NASA CASI (301) 621-0390

13. SUPPLEMENTARY NOTES
Point of Contact: David A. Saunders, ELORET, Ames Research Center, MS 230-2, Moffett Field, CA 94035-1000 (650) 604-1480. This report was prepared for NASA Ames Research Center under contract NAS2-99092.

14. ABSTRACT
Trajectory optimization program *Traj_opt* was developed at Ames Research Center to help assess the potential benefits of ultra-high temperature ceramic materials applied to reusable space vehicles with sharp noses and wing leading edges. *Traj_opt* loosely couples the Ames three-degrees-of-freedom trajectory package *Traj* (see NASA-TM-2004-212847) with the *SNOPT* optimization package (Stanford University Technical Report SOL 98-1). *Traj_opt* version January 22, 2003 is covered by this user guide. The program has been applied extensively to entry and ascent abort trajectory calculations for sharp and blunt crew transfer vehicles. The main optimization variables are control points for the angle of attack and bank angle time histories. No propulsion options are provided, but numerous objective functions may be specified and the nonlinear constraints implemented include a distributed surface heating constraint capability. Aero-capture calculations are also treated with an option to minimize orbital eccentricity at apoapsis. *Traj_opt* runs efficiently on a single processor, using forward or central differences for the gradient calculations. Results may be displayed conveniently with *Gnuplot* scripts. Control files recommended for five standard reentry and ascent abort trajectories are included along with detailed descriptions of the inputs and outputs.

15. SUBJECT TERMS
Trajectory optimization, Atmospheric reentry, Aero-capture, Ascent abort, 3DOF, CTV

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			David A. Saunders
Unclassified	Unclassified	Unclassified	Unclassified	69	19b. TELEPHONE *(Include area code)* (650) 604-1480