JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 5, September–October 2007

Interior-Point Approach to Trajectory Optimization

Julien Laurent-Varin*
INRIA, 78153 Rocquencourt Cedex, France
J. Frédéric Bonnans†
INRIA, 78153 Rocquencourt Cedex, France
Nicolas Bérend‡
ONERA, 92322 Châtillon, France
Mounir Haddou§
Université d'Orléans, 45067 Orléans Cedex 2, France
and
Christophe Talbot¶
Centre National d'Etudes Spatiales, 91023 Evry Cedex, France

DOI: 10.2514/1.18196

This paper presents an interior-point approach for solving optimal control problems. We combine the idea of logarithmic penalization (used to solve large-scale problems with relatively few iterations) with dedicated linear algebra solvers (*QR* factorization for band matrices). The method also takes advantage of recent progress in the analysis of discretization errors. At each major iteration of the interior-point algorithm (i.e., at a solution of the penalized problem for a given value of the penalty parameter), we determine whether discretization points should be added (and how to do so at low cost), because the number of operations is proportional to one of discretization points. Numerical results are displayed for various problems, including seven variants of atmospheric reentry of a space shuttle. We can find feasible points for all of them and compute a seemingly accurate solution for five of them. It can be seen from physical considerations that the two other problems are more difficult.

I. Introduction

THIS paper discusses the numerical resolution of optimal control problems of systems governed by ordinary differential equations. We deal with trajectory-optimization methods. In other words, the object manipulated by the algorithm is a certain function of time (as opposed to the Hamilton–Jacobi–Bellman approaches that solve a certain partial differential equation). Trajectory-optimization algorithms start from an initial, generally unfeasible, trajectory and find a trajectory that approximately satisfies some necessary conditions for optimality. In this paper, we assume all data to be at least C^2 (twice continuously differentiable), which allows the use of second-order Taylor expansions. Our format includes running mixed control and state constraints (which means that some constraints involving both the control and state, or only one of them, are always active).

Although there is no full agreement on this terminology, one generally distinguishes direct and indirect methods. Direct methods approximately solve a time-discretized version of the optimal control problem by some nonlinear programming algorithm and then possibly refine the discretization mesh and loop. In indirect methods, the control variable is eliminated via Pontryagin's principle, either analytically or numerically, and the resulting two-point boundary-value problem (TPBVP), with only state and costate variables (plus the Lagrange multipliers associated with the constraints), is to be solved. What is specific in these methods is that they do not store parameterizations of functions of time. The time integration is performed via a (somewhat hidden) ODE solver. The simplest indirect method is the shooting algorithm, which [with the pair (state and costate) at initial time] associates its value at the final time. In this way, necessary optimality conditions appearing in Pontryagin's principle are reduced to the final dimensional problem of matching the endpoint conditions. Because the integration of the reduced (state and costate) differential equation over a long time may be unstable, one often prefers to take as variables the values of state and costate at sampled points, so that integration occurs over small intervals of time (see Stoer and Bulirsch [1]).

The common feeling is that indirect methods are more efficient when the intervals of time during which a set of constraints is active is easy to predict, up to a parametrization of junction times (the endpoints of these intervals). Also, these junction times have to satisfy some regularity conditions (see Maurer [2]). In that case, by using high-order integration schemes, one obtains very precise and inexpensive solvers.

We note that the pseudospectral knotting methods of Ross and Fahroo [3] are direct methods, in the sense that they apply a nonlinear programming solver to a parametrization of the control and state, but at the same time, they make hypotheses on the structure of interior and boundary arcs, similar to those used in indirect methods.

Note that (especially in the case of state constraints) determining if various regularity conditions that are needed by the indirect approach hold, and writing the optimality conditions correctly, may be a delicate task [2]. It may also happen that little is known on the structure of the solution of the problem. In that case, it may be preferable to use direct methods, which are as easy to use as any nonlinear programming solver; in other words, finding a correct initial point often requires some intuition from the user, and some scaling and tuning of algorithmic parameters may help. We refer to

Received 14 June 2005; revision received 24 March 2006; accepted for publication 26 March 2006.

*Projet Sydoco, Domaine de Voluceau, Boîte Postale 105, currently Launcher Directorate, Centre National d'Etudes Spatiales, Rond-Point de l'Espace, 91023 Evry Cedex; Julien.Laurent-Varin@cnes.fr.

†Projet Sydoco, Domaine de Voluceau, Boîte Postale 105, currently INRIA-Futurs and Centre de Mathématiques Appliquées de l'Ecole Polytechnique (CMAP), 91128 Palaiseau, France; Frederic.Bonnans@inria.fr.

‡Long-Term Design and System Integration Department, Boîte Postale 72, 29 Avenue de la Division Leclerc; Nicolas.Berend@onera.fr.

§Unité Mixte de Recherche 6628, Physique Mathématique d'Orléans, Boîte Postale 6759; Mounir.Haddou@univ-orleans.fr.

¶Launcher Directorate, Rond-Point de l'Espace; Christophe.Talbot@cnes.fr.

Betz [4] and Bulirsch and Nerz for an overview of numerical methods for optimal control problems and comparison of various approaches.

Direct methods generally solve the optimality system of the discretized problem by using Newton or quasi-Newton algorithms. An early reference on Newton's method for (unconstrained) optimal control problems is Mitter [6]. Following the discovery of sequential quadratic programming (SQP) algorithms by Han [7] and Powell [8], a first generation of direct methods in which only control variables appear in the optimization solver were proposed (see Kraft). Implementations at that time used full matrices and full quasi-Newton approximations of the Hessian of the Lagrangian [based, for instance, on the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update]. This induced a severe limitation of dimension. Next were sparse implementations of SQP algorithms and their application to optimal control problems, using sparse finite differences for derivatives (Betts and Huffman [10]) or exact values of derivatives (Bonnans and Launay [11]). In these codes, the optimization variables are both control and state constraints, and the state equation is an equality constraint of the optimization solver. In both cases, the quadratic programming (QP) algorithm consists of an active-set strategy combined with a reduction of variables similar to reduced-gradient methods (see, for example, Gill et al. [12], Sec. 6.3). For all active-set strategies for large-scale problems, the number of inner iterations (of the QP) may become very large, because the active set typically changes by one component, at most, at each iteration. It is true, however, that the cost of an active-set iteration is usually smaller than that of an interior-point algorithm.

These drawbacks pushed some authors to propose interior-point algorithms for optimal control problems. Wright [13] analyzed some path-following algorithm for solving convex quadratic problems and applied them to the resolution of QPs arising in SQP algorithms for general optimal control problems. Vicente [14] discussed interior-point methods for nonlinear programs, but (despite the title) optimal control problems are only a motivation for the paper and are not really discussed. Leibfritz and Sachs [15] analyzed SQP algorithms with inexact resolution of the QP and applied this analysis to the resolution of the QP by interior-point algorithms. Wächter and Biegler [16] developed the interior-point nonlinear programming solver IPOPT, which was applied to an optimal control problem (optimization of a chemical process) in Jockenhövel et al. [17]. For the context of optimal control of partial differential equations, refer to [18,19].

All of the preceding papers put the emphasis on the optimization solver and do not take into account the influence of discretization errors. An exception is Betts [20], in which a refinement strategy is proposed based on steps with maximum local error (we compare this strategy to ours in Remark 4).

There is also literature specifically devoted to the analysis of discretization errors, mainly the analysis of distance between the solution of the continuous problem and the discretized problem; we review some of those results at the end of Sec. II.

Our contribution is to present a direct-type algorithm based on three main features: First,

1) It is an interior-point algorithm based on the logarithmic penalty of inequality constraints. An advantage of these algorithms is their ability to find a solution in a relatively small number of iterations.

2) We use a *QR* factorization for band matrices, which takes full advantage of the dynamic structure of the data; the cost of factorization is proportional to the number of time steps.

3) We have a flexible discretization refinement procedure that is activated at each major iteration of the algorithm (i.e., when after solving the optimality system for a given value of the optimization parameter). This is possible for two reasons:

a) Because the penalized problem is unconstrained, we are able to compute (discretization) error estimates.

b) The structure of the interior-point algorithm allows us to easily add new discretization points (this is not so easy with active-set strategies). We apply the method, in particular, to the atmospheric reentry problem.

The paper is organized as follows. We review the theory of error estimates for unconstrained optimal control problems (and some extensions to constrained problems) in Sec. II. We introduce an optimal refinement procedure in Sec. III. Fast, dedicated, numerical algebra is presented in Sec. IV. The interior-point approach for constrained problems is discussed in Sec. V. Turning to numerical results, Sec. VI presents an application to Fuller's problem. Finally, reentry problems are discussed in Sec. VII.

II. Unconstrained Problems: Error Estimates

In this section, we briefly review the recent analysis of error estimates for unconstrained optimal control problems by two of the authors [21]. It combines the formulation of the optimality system of the discretized problem (Hager [22]) with the theory of error orders for one-step methods. Adding an additional state variable, if necessary, we may assume that the cost is a function of the final state:

$$\min \Phi[y(T)]; \qquad \dot{y}(t) = f[y(t), u(t)]; \qquad t \in [0, T] \qquad y(0) = y^0 \tag{1}$$

where f and Φ are C^∞ functions $\mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbb{R}^n \to \mathbb{R}$, respectively. Then use a Runge–Kutta type discretization, in which the control variable is discretized similar to the state variable (i.e., with each "inner state," y_{ki} is associated with an inner control u_{ki}):

$$\begin{cases} \Phi(y_N) \\ y_{k+1} = y_k + h_k \sum_{i=1}^s b_i f(y_{ki}, u_{ki}); & k = 0, \ldots, N-1 \\ y_{ki} = y_k + h_k \sum_{j=1}^s a_{ij} f(y_{kj}, u_{kj}), & i = 1, \ldots, s \\ y_0 = y^0 \end{cases} \tag{2}$$

The positive time steps h_k are such that

$$\sum_{k=0}^{N-1} h_k = T$$

The Runge–Kutta scheme is parameterized by its coefficients a and b, an $s \times s$ matrix, and a vector of $\mathbb{R}^s$, respectively. Assuming all coefficients b_i to be nonzero, it was shown by Hager [22] that (after some change of variables in the Lagrange multipliers) the optimality system can be written in the following form:

$$\begin{cases} y_{k+1} = y_k + h_k \sum_{i=1}^s b_i f(y_{ki}, (u_{ki}), & k = 0, \ldots, N-1 \\ y_{ki} = y_k + h_k \sum_{j=1}^s a_{ij} f(y_{kj}, u_{kj}), & i = 1, \ldots, s \\ p_{k+1} = p_k - h_k \sum_{i=1}^s \hat{b}_i f_y(y_{ki}, u_{ki})^\top p_{ki}, & k = 0, \ldots, N-1 \\ p_{ki} = p_k - h_k \sum_{j=1}^s \hat{a}_{ij} f_y(y_{kj}, u_{kj})^\top p_{kj}, & i = 1, \ldots, s \\ 0 = f_u(y_k, u_k)^\top p_k, & k = 0, \ldots, N-1 \\ 0 = f_u(y_{ki}, u_{ki})^\top p_{ki}, & i = 1, \ldots, s \\ y_0 = y^0, \qquad p_N = \Phi'(y_N), \end{cases} \tag{3}$$

where

$$\hat{b} = b \quad \text{and} \quad \hat{a}_{ij} = (b_i b_j - b_j a_{ij})/b_i \quad \forall\ i \text{ and } j \tag{4}$$

The preceding system may be interpreted as a *partitioned Runge–Kutta discretization scheme* (i.e., a Runge–Kutta discretization with coefficients that depend on the index of the differential variable) for the optimality conditions, stated next, of problem (1):

$$\begin{cases} \dot{y}(t) = f[y(t), u(t)], & t \in [0, T] \\ \dot{p}(t) = -f_y[y(t), u(t)]^\top p(t), & t \in [0, T] \\ p(T) = \Phi'[y(T)], y(0) = y^0 \\ 0 = f_u[y(t), u(t)]^\top p(t), & t \in [0, T] \end{cases} \tag{5}$$

If the Hamiltonian function $H(y, u, p) = p \cdot f(u, y)$ is, in the neighborhood of the optimal trajectory, a strongly convex function of u, then we can eliminate (analytically or numerically) the control from the preceding algebraic constraint (via the implicit function

theorem) so that Eq. (5) reduces to a two-point boundary-value problem. In that case, if the solution of the discretized problem is, as we may hope, close to the solution of the continuous problem, we can also eliminate controls from the algebraic equations in Eq. (3). We obtain so-called continuous and discrete *reduced* (i.e., without control variables) formulations, and the discrete reduced formulation appears as a partitioned Runge–Kutta discretization of Eq. (5).

The theory of error orders for partitioned Runge–Kutta discretization schemes is now well understood [23–25]. When Eq. (4) holds, it can be proved that the scheme is symplectic (see, for example, [23] for a detailed study of symplectic schemes). Conditions for an error order up to four were obtained in [4]. One may find in [21] a simplification of order conditions derived from [23–25] for symplectic schemes, which allows them to be stated up to an order of six.

The preceding results apply only for unconstrained problems with strongly convex Hamiltonians. Dontchev et al. [26] showed that for problems with control constraints, high-order Runge–Kutta schemes typically will give (because of the constraint) an error of the order of two. For mixed control and state constraints, other researchers obtained an error estimate of the order of the step size. A similar result was obtained by Dontchev and Hager [27] for first-order state constraints. Extending these results to other classes of optimal control problem (as for Fuller's problem, studied in Sec. VI) is a significant challenge.

III. Mesh Refinement

Consider an ordinary differential equation, for which the endpoint conditions will be discussed later:

$$\dot{z}(t) = F[z(t)], \qquad t \in [0, T] \tag{6}$$

where F is a C^∞ mapping $\mathbb{R}^q \to \mathbb{R}^q$. The corresponding one-step discretization is

$$z_{k+1} = z_k + h_k \Phi(z_k, h_k) \tag{7}$$

where Φ: $\mathbb{R}^q \times \mathbb{R} \to \mathbb{R}^q$. Again, $\{h_k\}$ are positive time steps, such that

$$\sum_{k=0}^{N-1} h_k = T$$

For a Cauchy problem [i.e., if the initial condition $z(0)$ is given], the number of time steps is obtained by induction: given all h_i for $i < k$ and z_k, define the local error at step k as e_k: $\hat{z}_k(h_k) - z_{k+1}$, where $\hat{z}_k(t)$ satisfies $\hat{z}_k(0) = z_k$ and $\dot{\hat{z}}_k(t) = F[\hat{z}_k(t)]$. The theory of error estimates tells us that the norm e_k of the local error on step k is (at least for the Runge–Kutta schemes that we have in mind) of the form $e_k = C_k h_k^{p+1} + o(h_k^{p+1})$, where the order p is, in principle, known (or can be identified numerically). The error estimate for a first trial of the value of h_k may therefore be estimated by comparing the local variation of the differential variable with the variable computed by a scheme of higher order. If the time step is small enough, this gives an estimate $\hat{e}_k$ of the "true" local error e_k, satisfying

$$\hat{e}_k = C_k h_k^{p+1} + o\left(h_k^{p+1}\right) \tag{8}$$

The latter can be used to set h_k to a value that ensures that the local error is below a given threshold.

For a two-point boundary-value problem, the situation is different. Typically, the discretized problem is first solved using a rough estimate of the size of the time steps, and then ones tries to refine the mesh (i.e., to add other discretization points). Consider the *optimal refinement problem*, which is defined as problem of having the sum of local errors below a given threshold E. Because $e_k = C_k h_k^{p+1} + o(h_k^{p+1})$, dividing the step h_k into q_k smaller steps of size h_k/q_k, where q_k is a positive integer, reduces the principal term of the local error to $C_k(h_k/q_k)^{p+1}$ on each smaller step (i.e., a total of e_k/q_k^p on the q_k steps). Neglecting the contribution of $o(h_k^{p+1})$ to the local error, we see that the optimal refinement problem can be formulated as follows:

$$\min_{q \in \mathbb{N}_*^N} \sum_{k=1}^N q_k; \qquad \sum_{k=1}^N \frac{\hat{e}_k}{q_k^p} \le E \tag{9}$$

where N_* denotes the set of positive integers, and $\hat{e}_k$ is again the estimate of local error obtained with a higher-order scheme. Let us prove that this nonlinear integer programming problem can be solved inexpensively by Algorithm 1.

Algorithm 1 Mesh refinement

For $k = 1, \ldots, N$ do q_k: 1ϖ. End for
While

$$\sum_{k=1}^N e_k/q_k^p > E$$

do
Compute $k_g \in \operatorname{argmax}_k\{e_k[1/q_k^p - 1/(q_k+1)^p]\}$
q_{k_g}: $q_{k_g} + 1$.
End while

Let local gain be the amount of g_k: $\hat{e}_k[1/q_k^p - 1/(q_k+1)^p]$. This is the amount by which the estimate of the sum of local errors is decreased by increasing q_k by one. The algorithm simply consists of adding points for which the local gain is maximal.

Lemma 2: Algorithm 1 stops after a finite number of steps, and its output is the solution of Eq. (9).

Proof: We first prove by contradiction that Algorithm 1 stops. Otherwise, refinement would occur an infinite number of times for at least one component k_0, which means that for the corresponding subsequence, $e_k[1/q_k^p - 1/(q_k+1)^p] \to 0$. Because this maximum is nonincreasing over the iterations, it converges to zero. But then the stopping criterion is satisfied after a finite number of iterations.

Let us now prove that the output of the algorithm is a solution of the optimal refinement problem. Consider the related problem of minimizing the sum of estimates of local errors, subject to a given number of added points:

$$\min_{q \in \mathbb{N}_*^N} \sum_{k=1}^N \frac{\hat{e}_k}{q_k^p}; \qquad \sum_{k=1}^N q_k = N + m \qquad (D_m)$$

The value function of this problem, denoted by E_m, is strictly decreasing (we may assume all $\hat{e}_k$ to be positive) and has a limit of zero when $m \uparrow +\infty$. If $E_0 < E$, then there is no need to add points and the conclusion holds. Otherwise, there exists a unique $r \in \mathbb{N}_*$, such that

$$E_r \le E < E_{r-1} \tag{10}$$

Any solution of D_r is a solution of Eq. (9), because by the definition of r, it is not possible to satisfy the constraint of Eq. (9) by adding less than E_r points. We end the proof by showing that the output of Algorithm 1, denoted by $\{\bar{q}_k\}$, is a solution of D_r.

The proof is by induction over the values of $r \in \mathbb{N}$, such that Eq. (10) holds. If $r = 0$, then the conclusion holds (no point is added). Assume that it holds for all integers until $r-1$, and take a value of E such that Eq. (9) has the value of $N + r$. Let $\bar{q}$ be the output of Algorithm 1, and let k_0 be the index for which a time step has been added at the last step of Algorithm 1. Set $\hat{q}_k = \bar{q}_k$ for all $k \ne k_0$, and $\hat{q}_{k_0} = \bar{q}_{k_0} - 1$.

Let $\{q_k\}$ satisfy the constraint of D_r. We have to prove that the amount

$$\Delta: \sum_{k=1}^N \hat{e}_k(1/\bar{q}_k^p - 1/q_k^p)$$

is nonpositive. There exists i, such that $q_i > 1$. Set $\tilde{q}_k = q_k$ for all $k \ne i$, and $\tilde{q}_i = q_i - 1$. We may write $\Delta = \Delta_1 + \Delta_2$, where

$$\begin{aligned}\Delta_1&: \sum_{k=1}^{N} \hat{e}_k\left(\frac{1}{\hat{q}_k^p}-\frac{1}{\tilde{q}_k^p}\right)\\ \Delta_2&: \left(\frac{1}{\bar{q}_{k_0}^p}-\frac{1}{(\bar{q}_{k_0}-1)}\right)-\hat{e}_k\left(\frac{1}{q_i^p}-\frac{1}{(q_i-1)^p}\right)\end{aligned} \tag{11}$$

Because by our induction, $\{\hat{q}_k\}$ is a solution of D_{r-1} and $\tilde{q}$ is feasible for D_{r-1}, we get $\Delta_1 \le 0$. Let us prove that $\Delta_2 \le 0$ for a certain i. If $\bar{q}_{k_0} \le q_{k_0}$, then we may take $i = k_0$. Otherwise, in view of the constraint of D_r, we may take $i \neq k_0$, such that $\bar{q}_i < q_i$. Because k_0 is the index of maximal local gain, we have

$$\Delta_2 \le \hat{e}_i\left(\frac{1}{\bar{q}_i^p}-\frac{1}{(\bar{q}_i-1)^p}\right)-\hat{e}_i\left(\frac{1}{q_i^p}-\frac{1}{(q_i-1)^p}\right)<0 \tag{12}$$

Remark 3: A fast implementation of Algorithm 1 is obtained by sorting the time steps along the values of the local gain, which needs, at most, $\mathcal{O}(N \log N)$ operations. Each step of the algorithm needs to update this ordering, which needs, at most, $\mathcal{O}(\log N)$ operations. We find that the algorithm needs, at most, $\mathcal{O}[(N+m)\log N]$ operations. This is negligible with respect to the number of operations needed for computing Newton steps, as we will see in the next section.

Remark 4: The refinement strategy in [20] (Sec. 4.7.5) is a greedy algorithm refining the step with maximum local error. Its aim is to reduce the maximal error below some threshold Our greedy algorithm refines the step with maximal decrease, and we prove that it is optimal for minimizing the sum of local errors. Also in [20], no error estimate on dual equations is available and, hence, the refinement strategy takes into account only the integration errors in the state equation.

Our approach takes dual equations into account via the use of logarithmic penalty. A related drawback is that we do not take into account the error estimate (on the continuous problem) due to the logarithmic penalty itself (see the discussion of this last point in [28]).

IV. Linear Algebra

Solving the discrete optimality system (3) by a Newton step requires solving the linear systems for which the Jacobian matrix is (when variables are ordered by increasing values of time) a band matrix. We recall some classical results on band matrices, referring, for example, to [29] for details. Given a square matrix A of size n_A, denote the band size by q [i.e., the smallest integer, such that $A_{ij} = 0$ if $|i-j| > q$ (for instance, a diagonal matrix has band size 0)].

Lemma 5: The computation of the QR factorization, based on Givens's rotations, of a matrix A of band size q is such that the (upper triangular) factor R has band size $q = 2$. The number of operations for factorization is of the order of $q^2 n_A$, and the number of operations for solving the linear system (after factorization) is of the order of qn_A.

We apply this result to the Jacobian of Eq. (3). We may assume that $m = \mathcal{O}(n)$. Then the band size satisfies $q = \mathcal{O}(sn)$, where s is the number of inner steps in the Runge–Kutta scheme, and $n_A = \mathcal{O}(snN)$. We obtain the following.

Lemma 6: The computation of the QR factorization, based on Givens's rotations, of the Jacobian of Eq. (3) needs $\mathcal{O}(s^3 n^3 N)$ operations for factorization and $\mathcal{O}(s^2 n^2 N)$ operations for solving the linear system (after factorization).

Remark 7: The following hold

1) The cost of factorization of the Jacobian is of the same order that the integration of the state equation (the control being given) by a fully implicit scheme, and the ratio is of the order of s^2 for a semi-implicit scheme [with $\mathcal{O}(s)$ implicit steps]. This means that the computation of the Newton step is not much more expensive that the simulation of the system by an implicit scheme.

2) The ratio of the number of operations between factorization and solution is of the order of sn. This means that, generally, factorization will be, by far, the most expensive part of resolution.

3) For constrained problems, we must add rows (constraints) and variables (Lagrange multipliers). However, provided the number of constraints is of the order of n, the conclusion of Lemma 6 also holds in that case.

4) For multiple phase problems, we may first eliminate the variables indexed by time along each arc and then solve the resulting reduced system (see more on this subject in [29,30]).

V. Interior-Point Algorithms for Constrained Problems

A. Interior Point

Consider now a constrained optimal control problem of the following form:

$$\min \int_0^T \ell[y(t),u(t)]t + \Phi[y(T)]; \qquad \begin{cases}\dot{y}(t) = f[y(t),u(t)], & t\in[0,T]\\ 0 \le g[y(t),u(t)], & t\in[0,T]\\ y(0) = y^0\end{cases} \tag{13}$$

where $g\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^r$ is a mapping of class C^∞. The idea of logarithmic penalty is, for a (small) parameter ε, to introduce the penalized running cost

$$\ell_\varepsilon(y,u) := \begin{cases}\ell(y,u) - \varepsilon\sum_{i=1}^q \log[g_i(y,u)] & \text{if } g_i(y,u)>0, i=1,\dots,r\\ +\infty, & \text{otherwise}\end{cases} \tag{14}$$

and the related penalized optimal control problem

$$\min \int_0^T \ell_\varepsilon[y(t),u(t)]t + \Phi[y(T)]; \qquad \begin{cases}\dot{y}(t) = f[y(t),u(t)], & t\in[0,T]\\ y(0) = y^0\end{cases} \tag{15}$$

A feasible point of this problem is such that the state equation holds, $g_i[y(t),u(t)] > 0$ almost everywhere, and $\log\{g_i[y(t),u(t)]\}$ is a Lebesgue integrable $\forall\ i = 1$ to r.

Little is known on the theoretical properties of logarithmic penalty for optimal control problems. Bonnans and Guilbaud [31], assuming that running constraints are bound constraints on controls, proved convergence of optimal control and states (and costates) for either (strongly) convex linear quadratic problems or for a larger class in which the Hamiltonian function is convex with respect to the control. In addition, they proved that controls remain at a distance of at least $c\varepsilon$ from its bounds, where $c > 0$ is uniform for all sufficiently small ε. Asymptotic expansions of the solution for small ε might help for designing efficient algorithms; a rather specific case of control constraints is discussed in Alvarez et al. [28]. Weiser [32] (Theorem 3.10] obtains $\mathcal{O}\sqrt{\varepsilon}$ stability estimates for control-constrained problems. Little is known for problems with state constraints (see, however, [32]).

In practice, an initial feasible point is often unknown. In that case, it is advisable to rewrite the running constraint as

$$g[y(t),u(t)] - e(t) = 0; \qquad e(t) \ge 0, \qquad t\in[0,T] \tag{16}$$

and the penalized problem as

$$\begin{aligned}&\min \int_0^T \left\{\ell[y(t),u(t)] - \varepsilon\sum_{i=1}^q \log[e_i(t)]\right\}\mathrm{d}t + \Phi[y(T)]\\ &\dot{y}(t) = f[y(t),u(t)]; \qquad g[y(t),u(t)] - e(t) = 0; \qquad e(t) \ge 0\\ &t\in[0,T]; \qquad y(0) = y^0\end{aligned} \tag{17}$$

Clearly, problems (15) and (17) are equivalent. The advantage of the formulation in Eq. (17) is that, given a starting point for which the slack variable is positive, we may still consider Newton steps on the optimality system (even if the equality constraints are not satisfied). Therefore, our practical implementations are based on Eq. (17).

The penalized problem (15) is unconstrained with an integral and final cost. Adding an additional state variable y_{n+1}, such that $y_{n+1}(0) = 0$ and $\dot{y}_{n+1}(t) = \ell_\varepsilon[y(t),u(t)]$, allows the reduction to a

problem with final cost only, which can be discretized by the Runge–Kutta schemes discussed in Sec. II. For a given value of ε, the error analysis of that section applies when the steps vanish. The constants, however, will depend on ε.

An interesting open problem is to obtain error estimates for a given parameter ε and discretization steps. This is obviously difficult, because without logarithmic penalty, one may encounter a reduction order when state constraints are active (it is known that, in general, a Runge–Kutta scheme applied to an algebraic differential system suffers from order reduction). In addition, junction points for constraints (i.e., times when the constraint begins to be or stops being active) need a specific analysis.

Related to this is the question of the choice of a path of convergence (i.e., at which relative speed should the parameter ε and the discretization steps converge to zero). Again, this is essentially an open problem. In our present implementation, we simply require that the error estimate is not more than a constant times ε.

The idea of logarithmic penalty could also have been used on a discretized version of problem (13). In view of the discussion of Sec. II, a natural extension of discretization (2) would be

$$\begin{cases} \min \Phi(y_N) \\ y_{k+1} = y_k + h_k \sum_{i=1}^{s} b_i f(y_{ki}, u_{ki}), & k = 0, \ldots, N-1 \\ y_{ki} = y_k + h_k \sum_{j=1}^{s} a_{ij} f(y_{kj}, u_{kj}), & i = 1, \ldots, s \\ 0 \le g(y_{ki}, u_{ki}), & i = 1, \ldots, s \\ y_0 = y^0 \end{cases} \tag{18}$$

where we use the extended formulation: y has $n+1$ components and $f_{n+1} = \ell$. It appears that the logarithmic penalization of Eq. (18) is nothing but the Runge–Kutta discretization of Eq. (17). This remark is useful, because if we estimate the distance of the solution u_h^ε of the discretized problem to the solution $\bar{u}$ of the original distance by the following inequalities (for conveniently chosen norms), then each norm in the right-hand side corresponds either to a discretization estimate or to the estimate of variation due to logarithmic penalty:

$$\begin{cases} ||-u_h^\varepsilon|| \le ||\bar{u} - u^\varepsilon|| + ||u^\varepsilon - u_h^\varepsilon|| \\ ||\bar{u} - u_h^\varepsilon|| \le ||\bar{u} - u_h|| + ||u_h - u_h^\varepsilon|| \end{cases} \tag{19}$$

where u^ε and u_h denote the solutions of problems (15) and (18), respectively.

B. Dogleg Procedure

The first-order optimality conditions of the optimal control problem with logarithmic penalty are in a set of nonlinear equations of the form $F(X) = 0$, where X is the set of optimization parameters and Lagrange multipliers, and F is a smooth mapping with a square Jacobian matrix. We solve the nonlinear equation by applying a dogleg method to the least-squares formulation $||F(X)||^2$, as in Moré and Sorensen [33].

C. Threshold on the Sum of Local Errors

Two positive parameters, the penalty parameter ε and the threshold E on the sum of local error, have to reach zero, but at which relative speed? We do not know any theoretical analysis that would answer this question, and so in our tests, we arbitrarily take $E = \varepsilon/10$. Thus, the average local error is $\varepsilon/(10N)$, where N is the number of time steps.

VI. Fuller's Problem

Let us show how our method behaves when applied to Fuller's problem [34,35]. This is a simple academic problem with, as we will see, a nonregular junction point. The problem is as follows:

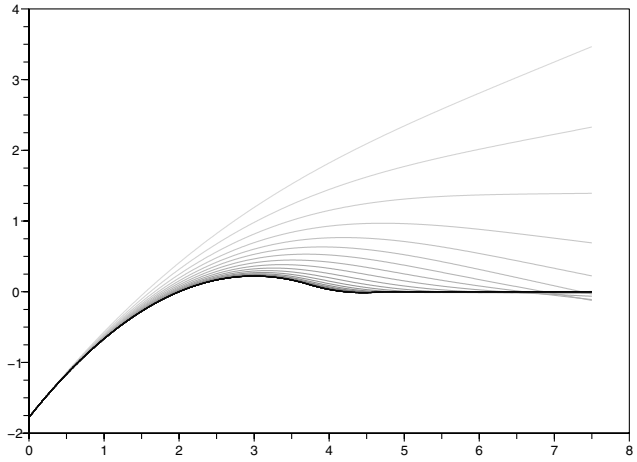

Fig. 1 Plot of x_1 vs time.

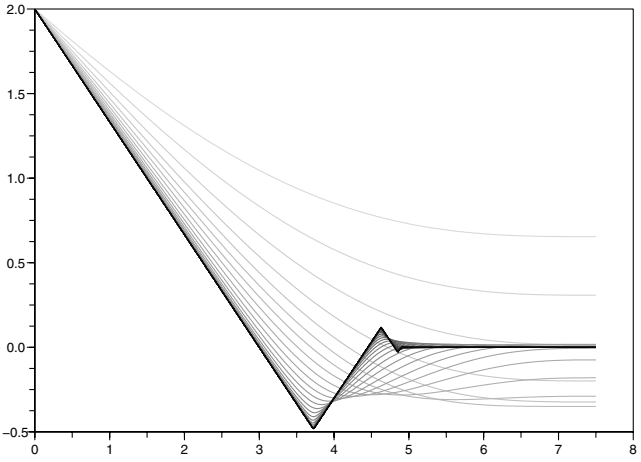

Fig. 2 Plot of x_2 vs time.

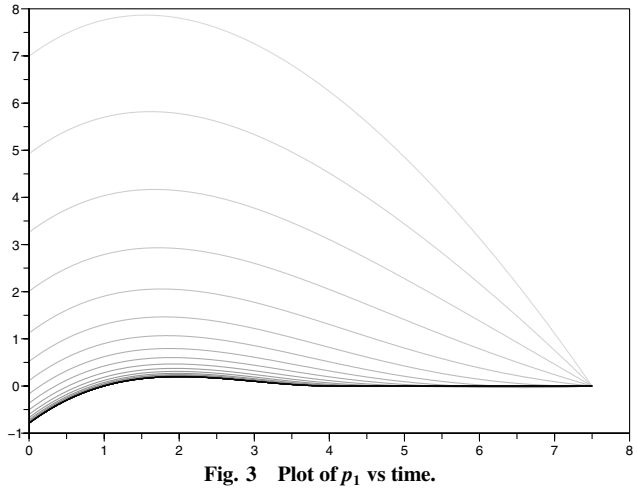

Fig. 3 Plot of p_1 vs time.

$$\min \frac{1}{2} \int_0^T x_1^2(t) \mathrm{d}t; \qquad \dot{x_1}(t) = x_2(t), \qquad \dot{x_2}(t) = u(t) \in [-1, 1]$$
$$t \in [0, T]; \qquad x(0) = a \tag{21}$$

where $a \in \mathbb{R}^2$ is given. Note that the Hamiltonian function $H(x, u, p) = \frac{1}{2}x_1^2 + p_1 x_2 + p_2 u$ is, as in Goddard's problem, linear with regard to the control variable. It was shown in [34,35] that an infinite number of switches may occur before reaching the singular

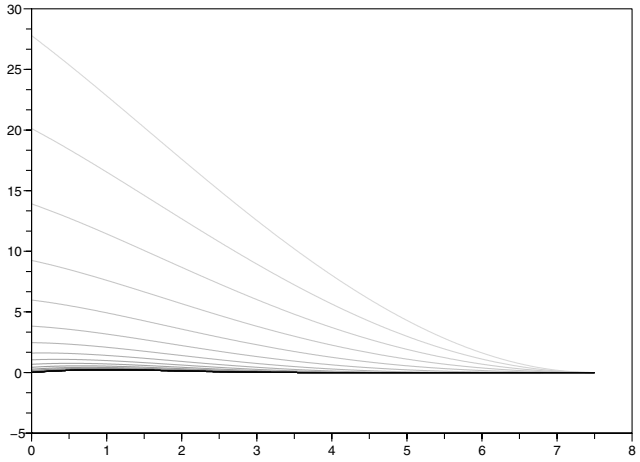

Fig. 4 Plot of p_2 vs time.

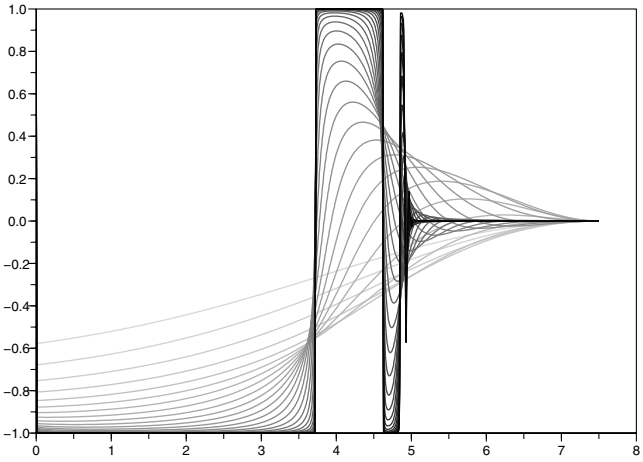

Fig. 5 Plot of u vs time (Fuller).

arc in which the state has a null value. Thus, numerical difficulties may be expected for this difficult problem, which does not satisfy the hypotheses stated before (strongly convex Hamiltonian).

In this example, the renement procedure did not behave well, probably in relation with the previously mentioned behavior, and so we display the results with a fixed grid of 400 points. We set a to the value of $(-1.778, 2.0)$ to obtain an infinite number of switches.

Figures 1–5 show the states, costates, and control variables. The algorithm does not converge if ε decreases too rapidly. Therefore, we choose a linear evolution of the form: $\varepsilon_{k+1} = \frac{1}{2}\varepsilon_k$. We start with $\varepsilon_0 = 16$ and stop with $\varepsilon_{35} = 4.657 \times 10^{-10}$.

VII. Atmospheric Reentry

A. Model

In this section, we study a problem of atmospheric reentry for a space shuttle, already considered by Betts [20] (Chap. 5, p. 133), to which we refer for a precise statement of state equation and constraints. It is enough to say that there are six state variables: altitude, longitude, latitude, velocity, flight path angle, and azimuth. The two control variables are the angle of attack and bank angle. There are bound constraints on controls and optional running constraints on the heating flux, all of which are identical to those in [20]. In addition, we have an optional running constraint of nonpositive flight path angle, which prevents skipping.

We maximize either the cross range or the range. Because we start at latitude 0 moving eastward, cross-range maximization results in maximizing the final altitude, whereas range maximization results in maximizing the final longitude. We use the Gauss–Legendre discretization schemes.

We number problems as shown in Table 1 (the third column refers to constraints other than bounds on control variables).

Table 1 Reentry problems

Problem number	Cost function	Constraints
1	Max cross range	None
2	Max cross range	Heating
3	Max cross range	Path angle
4	Max cross range	Heating and path angle
5	Max range	None
6	Max range	Heating
7	Max range	Path angle

B. Initialization Procedure

As already mentioned, the initialization procedure is an important component for the success of the method, especially for reentry trajectories, because it is known that the trajectory is very sensitive to the control.

1. *Maximum Cross Range with Only Control Constraints*

We initialize variables by the following heuristics: we integrate the optimality system starting from $t = 0$ and with zero initial costate. Of course, the final conditions are not satisfied.

2. *Maximum Cross Range with Heating Flux or Path Angle Constraints*

For these problems, we initialize the control, state, and costate with the value obtained by solving the problem without heating flux or path angle constraints, with a given value of the penalty parameter $\varepsilon \geq 1$. Then we initialize the slack variables and Lagrange multipliers in the following way:

$$e_i = \min\{\max[\tilde{g}_i, 1 - \sqrt{(}\varepsilon)], -1 + \sqrt{\varepsilon}\} \tag{20}$$

$$\lambda = \varepsilon(e-1)^{-1}; \qquad \mu = \varepsilon(e+1)^{-1} \tag{24}$$

where $\tilde{g}$ is the scaled path constraint function that (if the constraint holds) belongs to $[-1, 1]$. Thus, the inequality conditions on the slack variables are satisfied.

3. *Maximum Cross Range Without Any Constraints*

We initialize the problem with the solution obtained for the maximum cross-range cost function and with $\varepsilon = 16$.

4. *Maximum Cross Range with Each Constraint*

We initialize the problem with the solution obtained for the maximum range cost function, and with $\varepsilon = 1$.

Table 2 Problem 1: cross range

ε	Iterations	CPU time	Final latitude
16	37	1 min 12 s	24.31
8	10	23 s	24.82
4	8	20 s	25.79
2	8	21 s	27.48
1	7	20 s	29.67
0.5	7	18 s	31.59
0.25	6	18 s	32.94
0.125	7	22 s	33.66
4.41942×10^{-02}	6	19 s	34.01
9.29068×10^{-03}	6	20 s	34.11
8.95512×10^{-04}	5	16 s	34.1283
2.67983×10^{-05}	5	17 s	34.1288

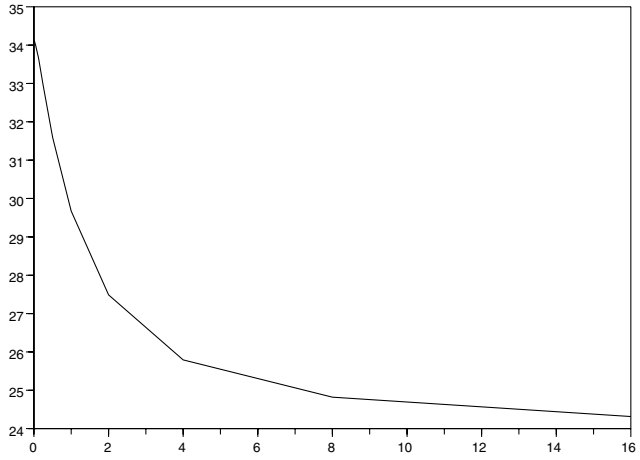

Fig. 6 Final latitude, deg vs ε.

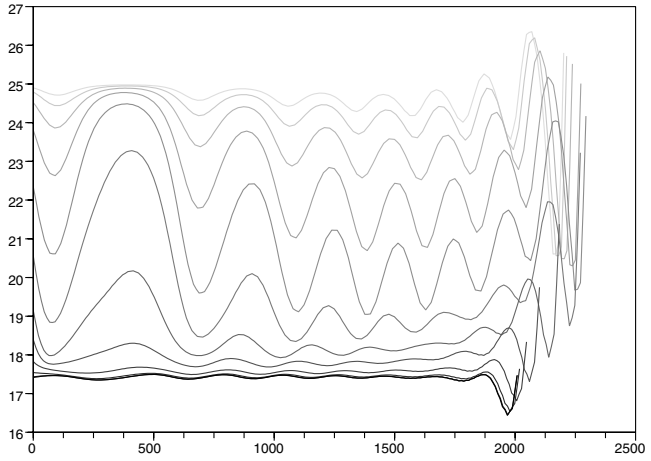

Fig. 7 Bank angle of attack, deg vs time, s.

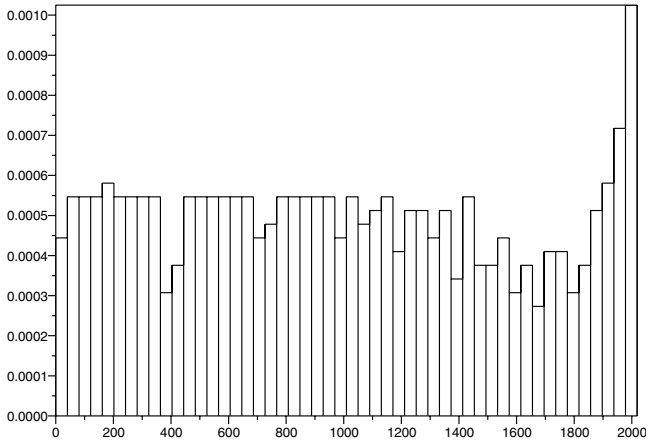

Fig. 8 Final density mesh in problem 1.

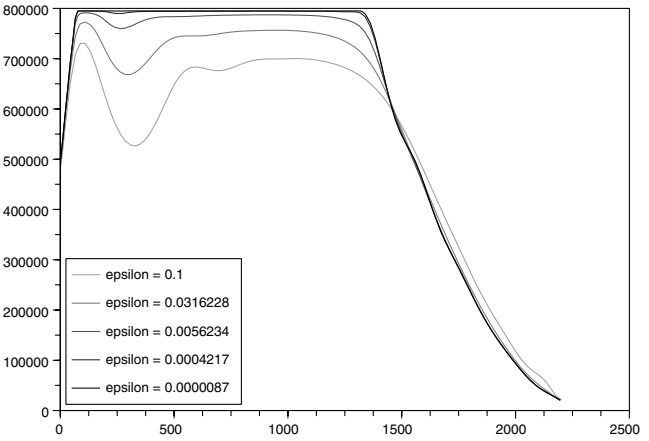

Fig. 9 Heating, W/m^2, vs time, s.

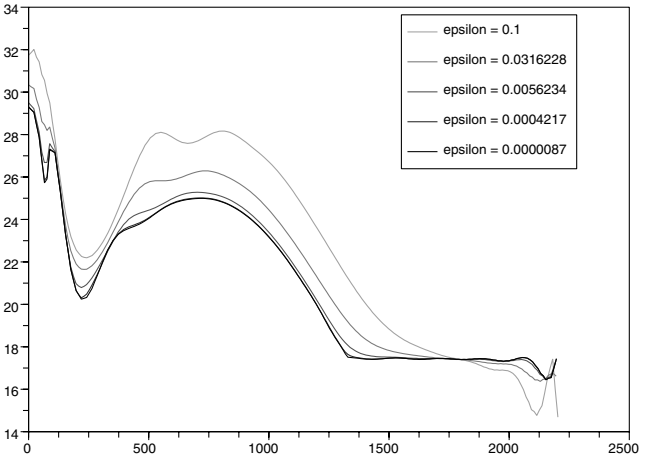

Fig. 10 Angle of attack, deg vs time, s.

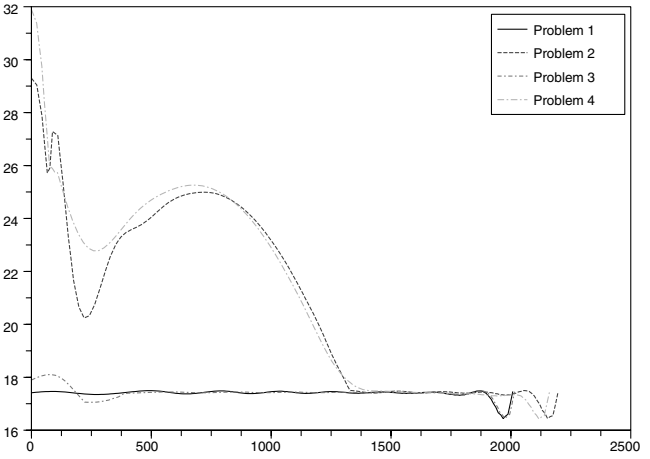

Fig. 11 Angle of attack, deg vs time, s.

C. Update of the Penalty Parameter

We update the penalty parameter ε in the following way:

$$\varepsilon_{k+1} = \min(\tfrac{1}{2}\varepsilon_k, \varepsilon^{3/2})$$

D. Numerical Results for Cross-Range Maximization

All figures display the evolution of some variable as a function of time. We first display the results for cross-range maximization.

1. Cross-Range Maximization with Only Control Constraints (Problem 1)

We first display the results for a grid of 100 equidistant points without refinement. Table 2 shows the evolution of the interior-point parameter, the number of dogleg iterations for each ε, the CPU time for each ε, and the performance index (here, the final latitude), to be compared with the value of 34.1412 deg given in [20].

We observe that although ε decreases rapidly in the last iterations, the number of Newton steps at each iteration remains small and, in fact, tends to decrease. The time needed for one Newton step is

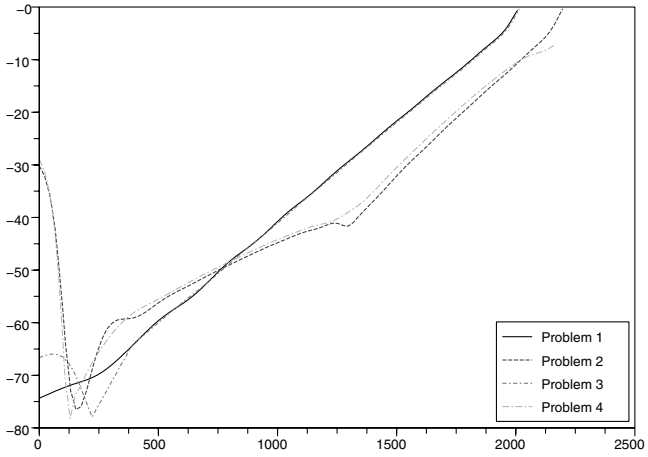

Fig. 12 Bank angle of attack, deg vs time, s.

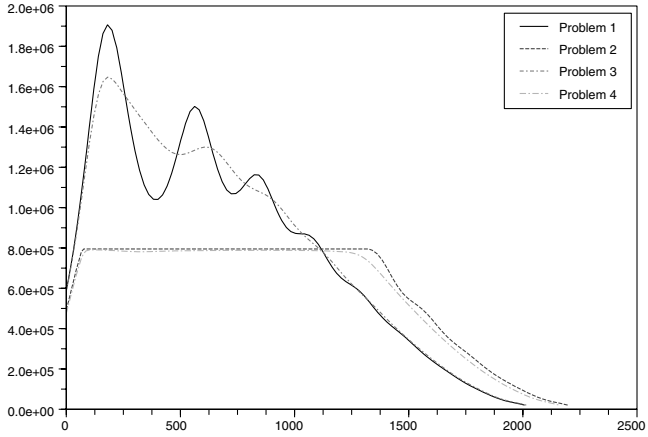

Fig. 13 Heating, W/m^2 vs time, s.

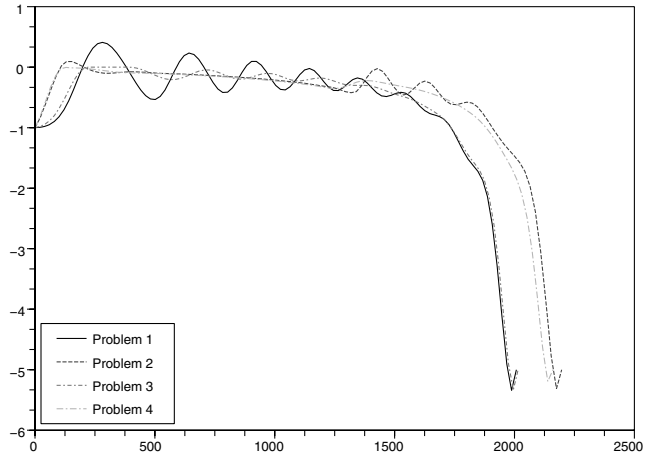

Fig. 14 Flight path angle, deg vs time, s.

approximately 3 s. Figures 6 and 7 display the performance index and the angle of attack as a function of ε.

In Table 3, we can notice the use of the refinement policy. We can notice that numerically, there is no difference in the performance index for the resolution without refinement. The density of discretization points for obtaining this precision is displayed in the histogram in Fig. 8.

There is a fast pull-up maneuver close to the final time, to comply with the prescribed final value of the flight path angle. This can be seen in the display of the angle of attack; accordingly, the density of discretization points is higher at the end of the trajectory.

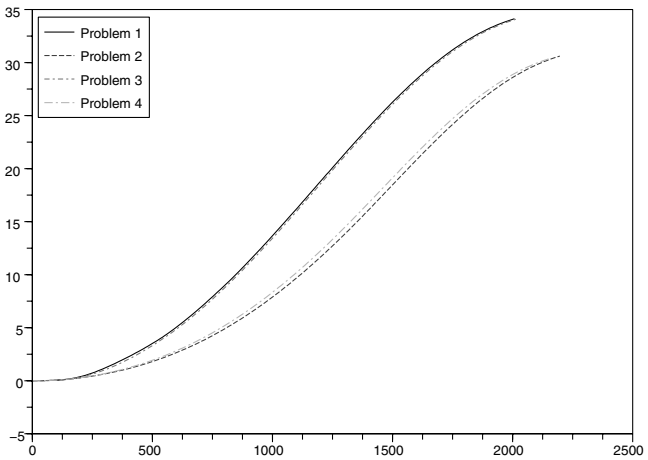

Fig. 15 Latitude, deg vs time, s.

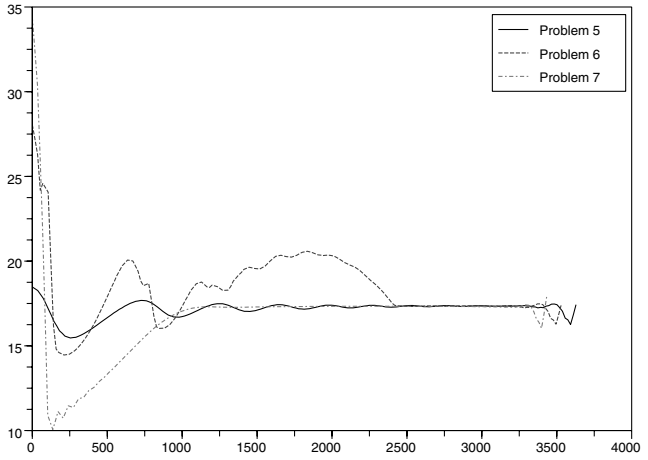

Fig. 16 Angle of attack, deg vs time, s.

Figures 11–15 compare the outputs of problems 1–4.

2. Cross-Range Maximization with Heating Flux Constraint (Problem 2)

We display our results for the problem with heating flux (problem 2), without refinement of the initial grid of 100 equidistant points. The execution report is given in Table 4, and the values of heating flux and angle of attack as a function of time for several values of ε are given in Figs. 9 and 10. We can observe that the angle of attack is now far from being almost constant, as it was in problem 1. With this additional constraint, we obtain a worse (as was to be expected) index performance, by about 10%.

3. Cross-Range Maximization with Path Angle Constraint (Problem 3)

We add the constraint of nonpositive path flight angle. As for the heating constraint, we initialize with the unconstrained problem

Table 3 Problem 1: cross range with refinement

ε	Iterations	Final latitude	Time steps	CPU time
16	1	24.31	100	3 s
8	10	24.82	100	23 s
4	8	25.79	100	23 s
2	8 + 3	27.48	100 + 2	29 s
1	7 + 3	29.68	102 + 42	37 s
0.5	6 + 3 + 2	31.59	144 + 49 + 9	1 min 45 s
0.25	7 + 3	32.94	202 + 31	2 min 40 s
0.125	7 + 2	33.66	233 + 27	2 min 56 s
0.0441942	6 + 2	34.01	260 + 93	3 min 56 s
0.0092907	6 + 2 + 2	34.11	353 + 366 + 5	17 min 9 s

Table 4 Problem 2: cross range, heating flux constraint

ε	Iterations	CPU time	Final latitude
0.1	13	1 min 2 s	27.85 deg
0.0316228	5	26 s	29.62 deg
0.0056234	4	19 s	30.42 deg
0.0004217	5	24 s	30.59 deg
0.0000087	6	29 s	30.61 deg

Table 5 Problem 3: cross range, path angle constraint

ε	Iterations	CPU time	Final latitude
0.1	14	39 s	33.46
0.032	11	38 s	33.95
0.0056	9	35 s	34.06
0.00042	7	30 s	34.087
0.0000087	9	41 s	34.090

Table 6 Problem 4: cross range, heating flux and path angle constraints

ε	Iterations	CPU time	Final latitude
0.1	15	1 min 19 s	26.98
0.032	11	1 min 12 s	29.45
0.0056	11	1 min 7 s	30.38

trajectory and obtain the results shown in Table 5. It appears now that the degradation of the criterion is of only about 0.1%, although the constraint was not satisfied in problem 1.

4. Cross-Range Maximization with Heating Flux and Flight Path Angle Constraint (Problem 4)

Problem 4 involves both constraints of flight path angle and heating. The algorithm converges until $\varepsilon = 0.0056$, but not for smaller values (perhaps a more conservative policy for the decrease of ε should be considered; see Table 6). It is likely that a numerical instability occurs because the two running state constraints induce a singularity. Indeed, observe that the trajectory obtained with the heating flux constraint only has a quite small flight path angle during a large portion of the flight. Similarly, on the view of the figures, we can see that the prediction of active constraints is not obvious (in particular, the flight path angle constraint might be active before the heating flux constraint becomes active and when it becomes inactive).

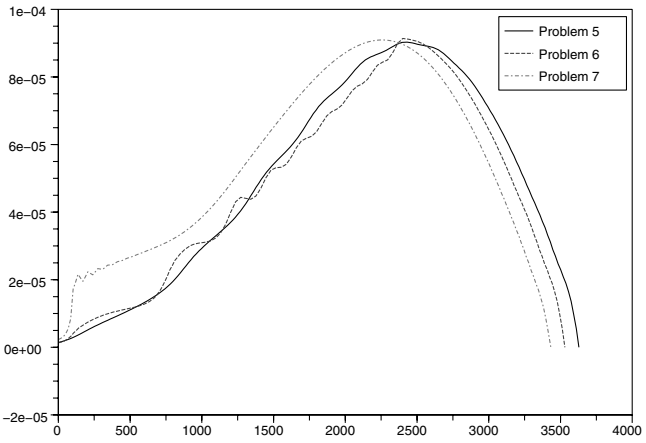

Fig. 17 Bank angle of attack, deg vs time, s.

Fig. 18 Heating, W/m^2 vs time, s.

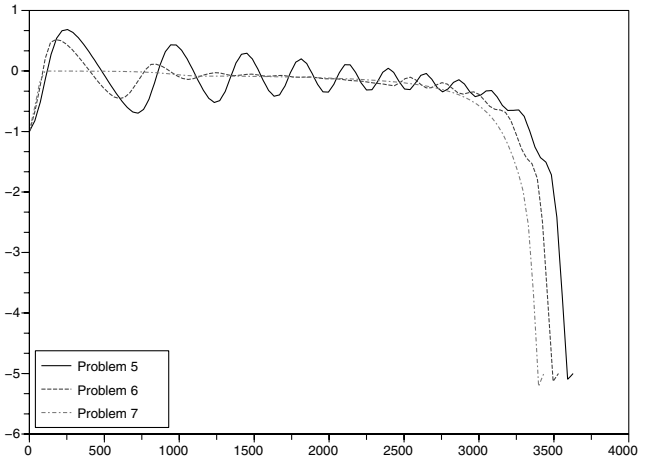

Fig. 19 Flight path angle, deg vs time, s.

E. Numerical Results for Range Maximization

In the model chosen, we would like to try another performance index: to maximize the forward range of the space shuttle. Figures 16–20 compare the outputs of the various range maximization problems.

1. Range Maximization with Only Control Constraints (Problem 5)

Table 7 shows the behavior of the algorithm. The number of dogleg iterations decreases and in the last major iterations, is no more than three. Note that the cost function is almost constant for $\varepsilon \leq 0.125$.

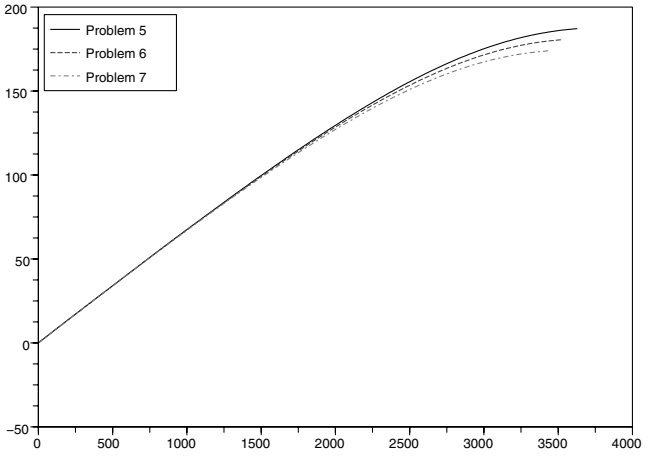

Fig. 20 Longitude, deg vs time, s.

Table 7 Problem 5: maximum range

ε	Iterations	Longitude	CPU time
16	28	171.53	1 min 16 s
8	11	176.47	43 s
4	9	181.13	36 s
2	8	184.55	32 s
1	7	186.27	28 s
0.5	5	186.89	20 s
0.25	3	187.08	14 s
0.125	3	187.16	13 s
0.0441942	3	187.16	13 s
0.00929068	3	187.16	14 s
0.000895512	2	187.16	10 s
$2.67983e-05$	2	187.16	13 s

Table 8 Problem 6: maximum range, heating flux constraint

ε	Iterations	Longitude	CPU time
1	386	167.95	24 min 6 s
0.5	21	170.79	1 min 44 s
0.25	20	172.36	1 min 36 s
0.125	17	173.22	1 min 21 s
0.0441942	18	173.86	1 min 26 s

Table 9 Problem 7: maximum range, path angle constraint

ε	Iterations	Longitude	CPU time
0.125	11	178.09	1 min 9 s
0.0441942	5	179.66	0 min 47 s
0.00929068	6	180.42	0 min 53 s
0.000895512	4	180.61	1 min 11 s
$2.67983e-05$	6	180.63	1 min 23 s

2. Range Maximization with Heating Flux Constraints (Problem 6)

For this problem, for which results are displayed in Table 8, the number of dogleg iterations is much larger. In the last iterations, the cost function does not change much; hence, we may hope that the last solution computed with $\varepsilon = 0.044$ is not far from the optimum. Interestingly, the problem seems to have three boundary arcs (and a total of seven arcs) for the state constraints, and this is probably related to the limited accuracy of our results in this case.

3. Range Maximization with Path Angle Constraints (Problem 7)

The results are displayed in Table 9. Here, the smallest value of ε for which convergence occurs is $\varepsilon = 8.95\text{e}-4$; again, the cost function does not change much in the last iterations, and so we may hope to have computed a reasonably good approximate solution.

VIII. Conclusions

We presented an approach for solving optimal control problems based on the interior-point methodology combined with a refinement procedure. Our present implementation allows the efficient solving of various applied problems. The cost of a Newton step is of the same order as a simulation with a fully implicit scheme. The refinement procedure leads to higher computing times but, because the number of operations is proportional to the number of time steps and the number of added steps seems to be minimal, that is the price to pay for precision (perhaps, in some example, we could test if some of the time steps could be merged). The software, however, encountered difficulties for the reentry problem when both constraints on flight path angle and heating flux are active for small values of the parameter ε.

Theoretical advances might help to give precise error estimates (to the solution of the original problem) and to determine the optimal path in the (ε, E) plane (E is the estimate of integration errors). Also, our dogleg procedure is quite simple and would deserve to be improved by taking into account the nature of optimality conditions. The way of decreasing the parameter ε is important for the efficiency of the method. We also observed that factorization is much more expensive that solving (when measured by the number of operations; the need for memory is similar). Therefore, a simplified Newton strategy might help. It is clear that there is a lot of room for improvement with this method.

Acknowledgment

We thank J. T. Betts from The Boeing Company for his constructive report, suggestions, and information about results obtained by his software SOCS on the problems dealt with in this paper.

References

[1] Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis*, Springer–Verlag, New York, 1993.

[2] Maurer, H., "On The Minimum Principle for Optimal Control Problems with State Constraints," *Schriftenreihe des Rechenzentrum* Univ. Münster, Münster, North Rhine-Westphalia, Germany, 1979.

[3] Ross, I., and Fahroo, F., "Pseudospectral Knotting Methods for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 397–508.

[4] Betts, J., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[5] Bulirsch, R., Nerz, E., Pesch, H., and von Stryk, O., *Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider*, Birkhäuser, Basel, Switzerland, 1993, pp. 273–288.

[6] Mitter, S., "Successive Approximation Methods for the Solution of Optimal Control Problems," *Automatica*, Vol. 3, Nos. 3–4, 1966, pp. 135–149.

[7] Han, S., "A Globally Convergent Method for Nonlinear Programming," *Journal of Optimization Theory and Applications*, Vol. 22, No. 3, 1977, pp. 297–309.

[8] Powell, M., "Algorithms for Nonlinear Constraints that Use Lagrangian Functions," *Mathematical Programming*, Vol. 14, No. 1, 1978, pp. 224–248.

[9] Kraft, D., "Finite-Difference Gradients Versus Error-Quadrature Gradients in the Solution of Parameterized Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 2, No. 2, 1981, pp. 191–199.

[10] Betts, J., and Huffman, W., "Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1992, pp. 198–206.

[11] Bonnans, J., and Launay, G., "Large Scale Direct Optimal Control Applied to a Re-Entry Problem," *Journal of Guidance, Control, and Dynamics*, , Vol. 21, No. 6, 1998, pp. 996–1000.

[12] Gill, P., Murray, W., and Wright, M., *Practical Optimization*, Academic Press, London, 1981.

[13] Wright, S., "Interior Point Methods for Optimal Control of Discrete Time Systems," *Journal of Optimization Theory and Applications*, Vol. 77, No. 1, 1993, pp. 161–187.

[14] Vicente, L., "On Interior-Point Newton Algorithms for Discretized Optimal Control Problems with State Constraints," *Optimization Methods and Software*, Vol. 8, Nos. 3–4, 1998, pp. 249–275.

[15] Leibfritz, F., and Sachs, E., "Inexact SQP Interior Point Methods and Large Scale Optimal Control Problems," *SIAM Journal on Control and Optimization*, Vol. 38, No. 1, 1999, pp. 272–293.

[16] Wächter, A., and Biegler, L., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Series A, Vol. 106, No. 1, 2006, pp. 25–57.

[17] Jockenhövel, T., Biegler, L. T., and Wächter, A., "Dynamic Optimization of the Tennessee Eastman Process Using the OptControlCentre," *Computers and Chemical Engineering*, Vol. 27, No. 11, 2003, pp. 1513–1531.

[18] Bergounioux, M., Haddou, M., Hintermüller, M., and Kunisch, K., "A Comparison of a Moreau-Yosida-Based Active Set Strategy and Interior Point Methods for Constrained Optimal Control Problems," *SIAM Journal on Optimization*, Vol. 11, No. 2, 2000, pp. 495–521.

[19] Betts, J., Eldersveld, S., Frank, P., and Lewis, J., "An Interior-Point Algorithm for Large Scale Optimization," *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Vol. 30, Springer, Berlin, 2003, pp. 184–198.

[20] Betts, J., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[21] Bonnans, J., and Laurent-Varin, J., "Computation of Order Conditions for Symplectic Partitioned Runge–Kutta Schemes with Application to Optimal Control," *Numerische Mathematik*, Vol. 103, No. 1, 2006, pp. 1–10.INRIA Rept. RR-5398, Rocquencourt, France, 2004.

[22] Hager, W., "Runge–Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, Vol. 87, No. 2, 2000, pp. 247–282.

[23] Hairer, E., Lubich, C., and Wanner, G., *Geometric Numerical Integration*, Springer Series in Computational Mathematics, Vol. 31, Springer–Verlag, Berlin, 2002.

[24] Hairer, E., Nørsett, S. P., and Wanner, G., *Solving Ordinary Differential Equations, 1*, 2nd ed., Springer Series in Computational Mathematics, Vol. 8, Springer–Verlag, Berlin, 1993.

[25] Hairer, E., and Wanner, G., *Solving Ordinary Differential Equations, 2*, 2nd ed., Springer–Verlag, Berlin, 1996.

[26] Dontchev, A., Hager, W., and Veliov, V., "Second-Order Runge–Kutta Approximations in Control Constrained Optimal Control," *SIAM Journal on Numerical Analysis*, Vol. 38, No. 1, 2000, pp. 202–226.

[27] Dontchev, A., and Hager, W., "The Euler Approximation in State Constrained Optimal Control," *Mathematics of Computation*, Vol. 70, No. 233, 2001, pp. 173–203.

[28] Alvarez, F., Bonnans, J., and Laurent-Varin, J., "Asymptotic Expansion of the Optimal Control Under Logarithmic Penalty: Worked Example and Open Problems," *HAL–INRIA* [online archive], INRIA, Research Rept. RR-6170, Rocquencourt, France, https://hal.inria.fr/inria-00143515 [retrieved Apr. 2007].

[29] Bérend, N., Bonnans, J., Laurent-Varin, J., Haddou, M., and Talbot, C., "Fast Linear Algebra for Multiarc Trajectory Optimization," *Large-Scale Nonlinear Optimization*, Vol. 83, edited by G. D. Pillo and M. Roma, Nonconvex Optimization and Its Applications Springer, New York, 2006, pp. 1–14.

[30] Laurent-Varin, J., Bérend, N., Bonnans, J., and Talbot, C., "An Efficient Optimization Method Dealing with Global RLV (Ascent and Descent) Trajectories," 56th International Astronautical Congress, International Astronautical Federation Paper IAC-05-C1.P.18, 2005.

[31] Bonnans, J. F., and Guilbaud, T., "Using Logarithmic Penalties in the Shooting Algorithm for Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 24, No. 5, 2003, pp. 257–278.

[32] Weiser, M., "Interior Point Methods in Function Space," *SIAM Journal on Control and Optimization*, Vol. 44, No. 5, 2005, pp. 1766–1786.

[33] Moré, J., and Sorensen, D., "Computing a Trust Region Step," *SIAM Journal on Scientific and Statistical Computing*, Vol. 4, No. 3, 1983, pp. 553–572.

[34] Fuller, A., "Relay Control Systems Optimized for Various Performance Criteria," *Proceedings of the IFAC Congress*, Butterworths, London, 1961, pp. 510–519.

[35] Fuller, A., "Study of an Optimum Non-Linear Control System," *Journal of Electronics and Control*, Vol. 15, 1963, pp. 63–71.