Computing gradients in parametrization–discretization schemes for constrained optimal control problems *

Marta L. Baguer
Universidad La Habana, Facultad de Matematica, La Habana, Cuba
Werner Römisch
Humboldt-Universität Berlin, Institut für Mathematik
D–10099 Berlin, Germany

Abstract

General parametrization–discretization schemes for transforming constrained optimal control problems into finite-dimensional nonlinear programs are considered. Formulae for computing gradients of the objective and constraints are derived via adjoint systems and discussed for linear multistep and Runge-Kutta discretization schemes. These formulae require one integration of the state equation only.

Keywords: optimal control, parametrization, discretization, computation of gradients

1991 *Mathematics Subject Classification*: 49 M 25, 65 K 10

1 Introduction

In this paper we consider the following optimal control problem of a system described by an ordinary differential equation (ode) for the state variables and by constraints for controls and states. Find a control function $u \in L_\infty([0,1];\mathbb{R}^r)$ and a design parameter $v \in \mathbb{R}^s$ such that the objective

$$(1.1) \qquad F(v,u) := g_0(x(1),v)$$

*This research has partially been supported by the Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung" of the Deutsche Forschungsgemeinschaft.

is minimized subject to the constraints

$$(1.2) \qquad \dot{x}(t) = f(x(t),u(t),v), \text{ a.e. } t \in [0,1],\ x(0) = x_0(v),$$
$$(1.3) \qquad g_j(x(1),v) \le 0 \quad (j = 1,\dots,d_1),$$
$$(1.4) \qquad g_j(x(t),u(t),v) \le 0 \text{ a.e. } t \in [0,1] \quad (j = d_1+1,\dots,d),$$
$$(1.5) \qquad u(t) \in U \quad \text{a.e. } t \in [0,1],\ v \in V.$$

For the data in (1.1) – (1.5) we assume that the functions g_j $(j = 0,\dots,d_1)$ and f, g_j $(j = d_1+1,\dots,d)$ are continuously differentiable on their domains $\mathbb{R}^m \times \mathbb{R}^s$ and $\mathbb{R}^m \times \mathbb{R}^r \times \mathbb{R}^s$, respectively. Furthermore, we assume that the sets $U \subseteq \mathbb{R}^r$ and $V \subseteq \mathbb{R}^s$ are closed and convex, and that x_0 is a continuously differentiable function from $\mathbb{R}^s$ into $\mathbb{R}^m$.
Optimal control models involving an integral functional and (or) integral constraints can be transformed into problem (1.1) – (1.5) by introducing additional state variables and equations. Models with free initial and (or) final time can be converted into (1.1) – (1.5) by including them into the design vector after a transformation to a fixed time interval. Nonautonomous ode's (1.2) or constraints (1.4) are also converted into the autonomous case by introducing a new state variable.

Computational methods for optimal control problems are usually subdivided into *direct* and *indirect* methods. Here "direct" means that the optimal control model is attacked as minimization problem in functional spaces, while "indirect" refers to treating the nonlinear boundary value problems arising as necessary optimality conditions for (1.1) – (1.5).
The *indirect* approach turned out to be very successful by applying the multiple shooting technique for solving the nonlinear boundary value problem (see e.g. [7], [10], [27], [32] and [12] for an impressive recent application to astronautics). In [34] the necessary optimality conditions are formulated as boundary value problems for differential-algebraic equations (dae) and solved by a version of the multiple shooting method for dae's.
In this paper, we are concerned with the *direct* approach. Direct methods usually consist of a combination of

(i) a finite-dimensional approximation, and

(ii) a minimization algorithm.

Papers contributing to (i) can be subdivided into papers devoted to finite-dimensional approximations of controls (control parametrization, Ritz method e.g. [20], [30], [38], [41], [42], [45], [46]) and somtimes of states (e.g. [47]), and to discrete approximations (e.g. [1], [9], [13], [14], [15], [16], [17], [18], [31], [33], [43]). Discrete approximations also include a discretization

of the state equation by using an integration scheme or by collocation. The finite-dimensional approximations are then solved by standard or adapted nonlinear programming methods or by multiple shooting techniques (cf. [7], [25]). A second stream of papers concerning the direct approach deals with minimization algorithms in functional spaces (like gradient-type and Newton-type methods in Banach spaces, see [37] for a modern survey and e.g. [2] for a very promising recent contribution), which are followed by a discretization step.
For solving complex real-world problems with high accuracy, a consensus is apparently tending toward a combination of direct and indirect methods. The result of a direct method together with estimates for the adjoint variables is then used to improve it by an accurate indirect method. This combination and the transition from direct to indirect methods is discussed in [25], [44].

The present paper contributes to the interplay of finite-dimensional approximations to (1.1) – (1.5) and their solution by nonlinear programming methods. An important question in this interplay consists in the appropriate and efficient calculation of first order derivatives of the objective and constraints. Several proposals (e.g. divided differences, backward integration of the adjoint equation, direct computation of derivatives from the discretized model) were made during the last fifteen years (see e.g. [3], [7], [17], [18], [20], [29], [36], [38], [43]) and used in several implementations (e.g. [4], [7], [17], [21], [22], [28], [30], [35], [38], [43]). We discuss these approaches in more detail in Section 3 and then extend the "direct approach" or "internal numerical differentiation". For a general approximation scheme including control parametrization and discretization of the state equation, we derive first order derivatives via discrete adjoint systems (Theorem 3.1). The structure of the discrete adjoint systems is studied for linear multistep and for Runge-Kutta discretizations of the state equation. Several parametrization approaches are considered, including those containing grid (or switching) points as parameters. We compare our results with more specific results in this direction obtained earlier ([5], [7], [18], [22], [36]) and discuss the approach and its complexity from the viewpoint of automatic differentiation of algorithms (see [24] for a modern survey). The implementation of the formulae for computing gradients together with the development of an optimal control package including numerical results for a set of illustrative test examples is described in the companion paper [4].

2 The parametrization–discretization scheme

In this section, we describe a general approach for approximating the infinite-dimensional optimization problem (1.1) – (1.5) by finite-dimensional nonlinear programs. The approach combines a control parametrization technique with a general discretization scheme for the initial value problem (1.2).
The idea of control parametrization consists in replacing the set of admissible controls

$$U_{\mathrm{ad}} := \{u \in L_\infty([0,1]; \mathbb{R}^r) : u(t) \in U \text{ a.e. } t \in [0,1]\}$$

by a set $U_{\mathrm{pa}} \subset L_\infty([0,1]; \mathbb{R}^r)$ of functions determined by a finite-dimensional parameter w belonging to some subset W of $\mathbb{R}^L$, i.e.

$$U_{\mathrm{pa}} := \{u(\cdot, w) : u(t,w) \in U \text{ a.e. } t \in [0,1],\ w \in W\}. \tag{2.1}$$

Typical examples of parametrizations are

(i) expansions of the control functions in terms of a finite number of (given) basis functions (here w corresponds to the coefficients in the expansion), and

(ii) a representation of control functions on a finite number of subintervals by given basis functions (e.g. polynomials) depending on a finite set of parameters (here w consists of parameters and (possibly) gridpoints).

To illustrate these general examples we mention Spline [42] and Chebyshev series [47] approximations (for (i)), "smooth" piecewise interpolating polynomials (Spline or Hermite) [29], [30], [35] and piecewise smooth control approximations [3], [20], [36], [38], [41], [45] (for (ii)).
If the parametrization is complicated, the constraints on the controls in (2.1) are of the same type as (1.4). However, they take a particularly simple form if a piecewise constant/linear parametrization or a representation of splines via the B-spline basis is used. These parametrizations are now described in some more detail.

Example 2.1
a) Piecewise constant parametrization:

Let $M \in \mathbb{N}$ and $\{0 =: \tau_0 < \tau_1 < \ldots < \tau_{M-1} < \tau_M := 1\}$ be a discretization of the interval $[0,1]$. Let χ_j $(j = 1, \ldots, M)$ denote the characteristic functions of the subintervals $[\tau_0, \tau_1]$, $(\tau_{j-1}, \tau_j]$, $j = 2, \ldots, M$. The first variant of the parametrization is to put $W := \underset{j=1}{\overset{M}{\times}} U = U^M$, $L := Mr$, $w := (u_1, \ldots, u_M)$ and

$$u(\cdot,w) := \sum_{j=1}^{M} u_j \chi_j. \tag{2.2}$$

In the second variant the switching points $\tau_1,\dots,\tau_{M-1}$ are considered as parameters in addition to the function values u_j in the subintervals, i.e., $L := Mr + M - 1$, $W := U^M \times \mathcal{T}$,
$\mathcal{T} := \{(\tau_1,\dots,\tau_{M-1}) : 0 \le \tau_{j-1} \le \tau_j \le 1, j = 2,\dots;M-1\}$,
$w := (u_1,\dots,u_M;\tau_1,\dots,\tau_{M-1})$ and $u(\cdot,w)$ has the form (2.2).
If $U \subseteq \mathbb{R}^r$ is convex polyhedral, this also holds for W as a subset of $\mathbb{R}^{Mr}$ and $\mathbb{R}^{M(r+1)-1}$, respectively.

b) Piecewise linear parametrization with jumps:

Let $M \in \mathbb{N}$ and put $W := U^{2Mr} \times \mathcal{T}$, where $\mathcal{T}$ is given as in a), $w := (u_1^+, u_1^-, u_2^+, \dots, u_M^+, u_M^-; \tau_1, \dots, \tau_{M-1})$ and

$$u(t,w) := \sum_{j=1}^{M} \frac{1}{\tau_j - \tau_{j-1}} (u_j^-(t-\tau_{j-1}) + u_j^+(\tau_j - t))\chi_j(t),\ t \in [0,1], \tag{2.3}$$

where χ_j are defined as in a).
Since U is convex, $u(t,w)$ belongs to U for any $w \in W$ and $t \in [0,1]$. u_j^+ and u_j^- are the right limit of $u(\cdot,w)$ at τ_{j-1} and the left limit of $u(\cdot,w)$ at τ_j, respectively ($j = 1,\dots,M$), and it holds that $u(\tau_0,w) = u_1^+$, $u(\tau_j,w) = u_j^-$, $j = 1,\dots,M$. W is a polyhedron in $\mathbb{R}^{M(2r+1)-1}$ if U has this property in $\mathbb{R}^r$. In the simplified variant of this parametrization the switching times $\tau_1,\dots,\tau_{M-1}$ are considered to be fixed.

Example 2.2 Let $M, k \in \mathbb{N}$ and $\{0 =: \tau_0 < \tau_1 < \dots < \tau_M := 1\}$ be a grid in $[0,1]$. Assume that $2k$ additional grid points $\tau_{-k} < \tau_{-k+1} < \dots < \tau_0$ and $\tau_M < \tau_{M+1} < \dots < \tau_{M+k}$ are given and let B_{ki} denote the i-th B-spline of k-th order, i.e.,

$$B_{ki}(t) := (\tau_{i+k+1} - \tau_i)[\tau_i, \tau_{i+1}, \dots, \tau_{i+k+1}][\cdot - t]_+^k,\ t \in \mathbb{R},$$
$$(i = -k, \dots, M-1),$$

where $[\tau_i,\dots,\tau_{i+k+1}]$ denotes the $(k+1)$-th divided difference and $[x]_+ := \max\{x,0\}$ (cf. [8]). For the parametrization we put $W := \overset{M-1}{\underset{j=-k}{\times}} U = U^{M+k}$, $L := (M+k)r$, $w := (u_{-k},\dots,u_{M-1})$ and

$$u(\cdot,w) = \sum_{j=-k}^{M-1} u_j B_{kj}(\cdot). \tag{2.4}$$

(Note that $w \in W$ implies $u(t,w) \in U$ for all $t \in [0,1]$, since B-splines are nonnegative and $\sum_{j=-k}^{M-1} B_{kj}(t) = 1$ for all $t \in [0,1]$ (see [8]).)
Including the grid points $\tau_1,\dots,\tau_{M-1}$ into the set W of parameters we obtain $W := U^{M+k} \times \mathcal{T}$, $L := (M+k)r + M - 1$, $w := (u_{-k},\dots,u_{M-1};\tau_1,\dots,\tau_{M-1})$ and $u(\cdot,w)$ has the form (2.4). Values of $u(\cdot,w)$ can be computed efficiently by using the recursion formulae for B-splines (cf. [8]).

Next we describe the discretization of the initial value problem (1.2). Suppose that an arbitrary grid $\tau_0 := 0 < \tau_1 < \dots < \tau_M := 1$ of the interval $[0,1]$ is given. This grid is used for defining the control parametrization and is either fixed or considered as a decision variable. For the integration of (1.2), where the control function $u(\cdot)$ is replaced by the parametrization $u(\cdot,w)$, we use a general variable multistep method of the form

$$\sum_{j=0}^{k_\ell} a_{\ell j} x_{\ell-j} = h_\ell \varphi_\ell(x_\ell,\dots,x_{\ell-k_\ell};v,w),\quad \ell = 1,\dots,N, \tag{2.5}$$

where $x_0 = x_0(v)$, $k_\ell \le \ell$, $a_{\ell 0} := 1$, for all $\ell = 1,\dots,N$, $h_\ell := t_\ell - t_{\ell-1}$, $\ell = 1,\dots,N$, denote the stepsizes and $\{t_0 := 0 < t_1 < t_2 < \dots < t_N := 1\}$ forms a grid of $[0,1]$ such that $\tau_i \in \{t_1, t_2, \dots, t_{N-1}\}$, $i = 1,\dots,M-1$, i.e., the grid for the integration of (1.2) is selected as a refinement of that for the control parametrization. The variable orders $k_\ell \in \mathbb{N}$, (real) coefficients $a_{\ell j}$ ($j = 0,\dots,k_\ell$) and functions φ_ℓ ($\ell = 1,\dots,N$) define the integration method (2.5). (2.5) includes a variety of numerical methods for solving (1.2), e.g., linear multistep (multiderivative) methods and nonlinear one-step methods (cf. e.g. [26]). For linear multistep methods the functions φ_ℓ are of the form

$$\varphi_\ell(x_\ell,\dots,x_{\ell-k_\ell};v,w) := \sum_{j=0}^{k_\ell} b_{\ell j} f(x_{\ell-j}, u(t_{\ell-j},w), v), \tag{2.6}$$

where $b_{\ell j}$ ($j = 0,\dots,k_\ell$) are real coefficients.
The variable orders k_ℓ are selected such that at each possible jump τ_i of the parametrized control the order is set to be equal to 1, i.e., $k_{\ell(i)+1} := 1$ if $\tau_i = t_{\ell(i)}$ ($i = 1,\dots,M-1$). This means that at each τ_i a restart of the (self-starting) multistep method is organized to perform the integration of (1.2) in the interval $(\tau_i, \tau_{i+1}]$ where $u(\cdot,w)$ is "smooth", and to avoid losses of accuracy at the jumps. Of course, the value of $u(\cdot,w)$ at $t = \tau_i$ in (2.5), (2.6) has to be taken as its right limit $u(\tau_i+,w)$.
For a q-stage Runge-Kutta method, (2.5) takes the form

$$
\begin{aligned}
x_\ell &= x_{\ell-1} + h_\ell \sum_{i=1}^{q} b_i K_{\ell i}, \quad \ell = 1, \ldots, N, \\
K_{\ell i} &= f(x_{\ell-1} + h_\ell \sum_{j=1}^{q} \alpha_{ij} K_{\ell j},\ u(t_{\ell-1} + c_i h_\ell, w), v),\ i = 1, \ldots, q,
\end{aligned} \tag{2.7}
$$

where $A = (\alpha_{ij})$, $b = (b_1, \ldots, b_q)$, $c = (c_1, \ldots, c_q)$ are the parameters of the Runge-Kutta scheme. The method is called explicit if $\alpha_{ij} = 0$ whenever $i < j$ (implicit otherwise).
Of course, combinations of linear multistep methods with nonlinear one-step methods (e.g. Runge-Kutta starters for multistep methods) also belong to the general scheme (2.5).

Discretizing the control and state constraints (1.4) and (1.5) at each grid point t_ℓ ($\ell = 1, \ldots, N$) of the state grid, we arrive altogether at the following parametrization–discretization scheme for the approximate solution of (1.1) – (1.5): Minimize

$$\hat{F}(v, w) := g_0(x_N, v) \tag{2.8}$$

subject to the constraints

$$\sum_{j=0}^{k_\ell} a_{\ell j} x_{\ell-j} = h_\ell \varphi_\ell(x_\ell, \ldots, x_{\ell-k_\ell}; v, w),\ \ell = 1, \ldots, N,\ x_0 = x_0(v), \tag{2.9}$$

$$g_j(x_N, v) \le 0 \quad (j = 1, \ldots, d_1), \tag{2.10}$$

$$g_j(x_\ell, u(t_\ell, w), v) \le 0 \quad (j = d_1 + 1, \ldots, d;\ \ell = 1, \ldots, N), \tag{2.11}$$

$$v \in V, \quad w \in W. \tag{2.12}$$

(2.8) - (2.12) represents a nonlinear program with many constraints and $s + L$ variables.

3 Computing gradients

Solving (2.8) – (2.12) by standard nonlinear programming methods requires the (approximate) calculation of derivatives for both the objective and constraints. In this section, we discuss several knwon principles for the computation of gradients and extend one of these approaches. To this end, we consider the following function

$$G(p) := g(x_{\ell_*}, p), \quad p := (v, w) \in \mathbb{R}^{s+L}, \text{ where} \tag{3.1}$$

$$\sum_{j=0}^{k_\ell} a_{\ell j} x_{\ell-j} = h_\ell \varphi_\ell(x_\ell, \ldots, x_{\ell-k_\ell}; p),\ \ell = 1, \ldots, \ell_*,\ x_0 = x_0(v) \tag{3.2}$$

and $\ell_* \in \{1, \ldots, N\}$ is fixed. The objective (2.8) and the functions in (2.10), (2.11) are special cases of (3.1) – (3.2).
For the calculation of the gradient (partial derivatives) of G the following general techniques have been used for the last two decades:

(i) Numerical calculation by finite differences (e.g. [7], [29]);

(ii) numerical approximation of the "continuous" gradient (in (1.1) – (1.4)) by integration of the dual equation (e.g. [20], [21], [38], [46]);

(iii) numerical integration of the enlarged ode (1.2) for the state $x(\cdot)$ and the derivatives $\frac{\partial x(\cdot)}{\partial p_i}$ ($i = 1, \ldots, s + L$) simultaneously;

(iv) computation of the gradient of the discretized problem or internal numerical differentiation (e.g. [5], [6], [17], [18], [22], [36]).

Method (i) is easy to implement, but has the drawback that $s + L + 1$ high accuracy integrations of (1.2) have to be performed. (ii) requires the numerical integration of two ode's, namely the state equation (1.2) and the dual (adjoint, costate) equation

$$\lambda'(t) = -\Big[\frac{\partial f}{\partial x}(x(t), u(t, w), v)\Big]^T \lambda(t),\ t \in [0, 1],\ \lambda(1) = \frac{\partial g}{\partial x}(x(1), p), \tag{3.3}$$

successively. The partial derivatives of G can then be computed according to the formulae in [20], [37], [38]. The essential drawback of this approach consists in numerical difficulties during the automatic backward integration of (3.3), since the state variables are only available at the nodes selected during forward integration of (1.2). Relations of this approach to internal numerical integration (iv) using adjoint systems are discussed in Remark 3.7.
Method (iii) is again easy to implement, but suffers from the high dimension ($(s + L + 1)m$) of the ode. In contrast to the "external" differentiation methods (i) – (iii) the basic idea of (iv) is to compute the derivatives of the internally selected discretization in two modes. The first mode is the direct computation of the partial derivatives $\frac{\partial x_{\ell_*}}{\partial p_i}$ ($i = 1, \ldots, s + L$) during the (automatic) integration of (1.2) with variable order and step (leading to the scheme (3.2)). The partial derivatives are calculated by formulae that follow from (3.2) by applying the chain rule. The second (indirect) mode consists in the computation of $\frac{\partial G}{\partial p_i}$ by using adjoint systems. The underlying idea is similar to the use of the dual equation in (ii) and leads, in fact, to a certain "induced" discretization of the dual (3.3). The approach (iv), although developed independently during the last 15 years, resembles

the idea of automatic differentiation of algorithms (cf. [24] for a recent state-of-the-art). Both modes (direct and indirect) are variants of the so-called forward and reverse modes of automatic differentiation ([19], [24], Section 4). Compared to the techniques (i) - (iii), the internal numerical differentiation (iv) leads to a drastic reduction of computing time (60 - 80 % are reported in [5], [6] compared to finite difference approximations), since it requires the numerical integration of (1.2) only once.
In what follows, we extend earlier results on the indirect mode for specific integration schemes ([7], [18], [36]) to general variable multistep methods (3.2) and discuss particular cases. We begin with stating a general result in this direction.

Theorem 3.1 *Let g, φ_ℓ and h_ℓ, $a_{\ell j}$ (viewed as a function of p) be continuously differentiable for each $j = 1, \dots k_\ell$, $\ell = 1, \dots, \ell_*$.*
If the partial derivatives $\frac{\partial x_\ell}{\partial p_i}$, $\ell = 1, \dots, \ell_*$, exist for some $i \in \{1, \dots, s+L\}$, then it holds

$$\frac{\partial G}{\partial p_i} = \sum_{\ell=1}^{\ell_*} \lambda_\ell^T \Big(h_\ell \frac{\partial \varphi_\ell}{\partial p_i} + \frac{\partial h_\ell}{\partial p_i} \varphi_\ell - \sum_{j=0}^{k_\ell} \frac{\partial a_{\ell j}}{\partial p_i} x_{\ell-j} \Big) + \lambda_0^T \frac{\partial x_0}{\partial p_i} + \frac{\partial g}{\partial p_i}, \tag{3.4}$$

whenever

$$\lambda_{\ell_*} = \Big(\frac{\partial g}{\partial x}\Big)^T + h_{\ell_*} \Big(\frac{\partial \varphi_{\ell_*}}{\partial x_{\ell_*}}\Big)^T \lambda_{\ell_*}, \quad \textit{and} \tag{3.5}$$

$$\sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{\ell_*-\ell} a_{\ell+j,j} \lambda_{\ell+j} = \sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{\ell_*-\ell} h_{\ell+j} \Big(\frac{\partial \varphi_{\ell+j}}{\partial x_\ell}\Big)^T \lambda_{\ell+j}, \tag{3.6}$$

$$\ell = \ell_* - 1, \dots, 0, \; h_0 := 0.$$

(Here we use the abbreviations $\frac{\partial \varphi_\ell}{\partial p_i} = \frac{\partial \varphi_\ell}{\partial p_i}(x_\ell, \dots, x_{\ell-k_\ell}; p)$, $\frac{\partial \varphi_\ell}{\partial x_j} = \frac{\partial \varphi_\ell}{\partial x_j}(x_\ell, \dots, x_{\ell-k_\ell}; p)$, $\frac{\partial x_0}{\partial p_i} = \frac{\partial x_0}{\partial p_i}(v)$, $\frac{\partial g}{\partial p_i} = \frac{\partial g}{\partial p_i}(x_{\ell_*}, v)$ etc.)

Proof: Let the partial derivatives $\frac{\partial x_\ell}{\partial p_i}$, $\ell = 1, \dots, \ell_*$, exist. Then we obtain from (3.2) by the chain rule,

$$\sum_{j=0}^{k_\ell} a_{\ell j} \frac{\partial x_{\ell-j}}{\partial p_i} = \Phi_\ell + h_\ell \sum_{j=0}^{k_\ell} \frac{\partial \varphi_\ell}{\partial x_{\ell-j}} \frac{\partial x_{\ell-j}}{\partial p_i}, \; \ell = 1, \dots, \ell_*, \tag{3.7}$$

where $\Phi_\ell := \frac{\partial h_\ell}{\partial p_i} \varphi_\ell + h_\ell \frac{\partial \varphi_\ell}{\partial p_i} - \sum_{j=0}^{k_\ell} \frac{\partial a_{\ell j}}{\partial p_i} x_{\ell-j}$, $\ell = 1, \dots, \ell_*$.

Now, let λ_ℓ, $\ell = 0, \dots, \ell_*$, be given by (3.5), (3.6). Then we have by (3.7),

$$\begin{aligned}
\sum_{\ell=1}^{\ell_*} \lambda_\ell^T \Phi_\ell &= \sum_{\ell=1}^{\ell_*} \lambda_\ell^T \Big\{ \sum_{j=0}^{k_\ell} \Big(a_{\ell j} - h_\ell \frac{\partial \varphi_\ell}{\partial x_{\ell-j}} \Big) \frac{\partial x_{\ell-j}}{\partial p_i} \Big\} \\
&= \sum_{r=0}^{\ell_*} \Big[\sum_{\substack{\ell=1 \\ \ell-j=r}}^{\ell_*} \sum_{j=0}^{k_\ell} \Big\{ a_{\ell j} \lambda_\ell - h_\ell \Big(\frac{\partial \varphi_\ell}{\partial x_{\ell-j}} \Big)^T \lambda_\ell \Big\}^T \Big] \frac{\partial x_r}{\partial p_i} \\
&= \sum_{r=1}^{\ell_*} \sum_{\substack{j=0 \\ j \le k_{r+j}}}^{\ell_*-r} \Big\{ a_{r+j,j} \lambda_{r+j} - h_{r+j} \Big(\frac{\partial \varphi_{r+j}}{\partial x_r} \Big)^T \lambda_{r+j} \Big\}^T \frac{\partial x_r}{\partial p_i} \\
&\quad + \sum_{\substack{j=1 \\ j \le k_j}}^{\ell_*} \Big\{ a_{jj} \lambda_j - h_j \Big(\frac{\partial \varphi_j}{\partial x_0} \Big)^T \lambda_j \Big\}^T \frac{\partial x_0}{\partial p_i} \\
&= \Big\{ \lambda_{\ell_*} - h_{\ell_*} \Big(\frac{\partial \varphi_{\ell_*}}{\partial x_{\ell_*}} \Big)^T \lambda_{\ell_*} \Big\}^T \frac{\partial x_{\ell_*}}{\partial p_i} - \lambda_0^T \frac{\partial x_0}{\partial p_i} = \frac{\partial g}{\partial x} \frac{\partial x_{\ell_*}}{\partial p_i} - \lambda_0^T \frac{\partial x_0}{\partial p_i}.
\end{aligned}$$

Hence, we obtain finally

$$\frac{\partial G}{\partial p_i} = \frac{\partial g}{\partial x} \frac{\partial x_{\ell_*}}{\partial p_i} + \frac{\partial g}{\partial p_i} = \sum_{\ell=1}^{\ell_*} \lambda_\ell^T \Phi_\ell + \lambda_0^T \frac{\partial x_0}{\partial p_i} + \frac{\partial g}{\partial p_i}. \qquad \square$$

Remark 3.2 The partial derivatives $\frac{\partial x_\ell}{\partial p_i}$ ($\ell = 1, \dots, \ell_*$, $i = 1, \dots, s+L$) exist and are continuous by the implicit function theorem if the matrices $I - h_\ell \frac{\partial \varphi_\ell}{\partial x_\ell}$ ($\ell = 1, \dots, \ell_*$) are nonsingular. This condition also implies that the discrete adjoints λ_ℓ ($\ell = 0, \dots, \ell_*$) are well-defined by (3.5), (3.6), and it is implied if the maximal stepsize of the state grid is sufficiently small. (3.5), (3.6) may be viewed as an *"induced" discretization* of the dual equation (3.3). Hence, we call the maximal number of the summation index j in (3.6), i.e.,

$$k_\ell^* := \max\{j \in \{1, \dots, \ell_* - \ell\} : j \le k_{\ell+j}\}, \tag{3.8}$$

the (variable) *dual* order in step ℓ ($\ell = \ell_* - 1, \dots, 1$) of the backward scheme (3.5), (3.6). It turns out that for each index ℓ such that t_ℓ belongs to the control grid $\{\tau_1, \dots, \tau_{M-1}\}$, we have $k_{\ell-1}^* = 1$ since $k_{\ell+1} = 1$. Hence, the induced backward scheme has the same behaviour as the forward scheme (2.4) at those grid points.

Remark 3.3 Since the stepsizes h_ℓ and the coefficients $a_{\ell j}$ depend on control grid points only, the formula (3.4) for the partial derivatives $\frac{\partial G}{\partial p_i}$ take a particular form for a special choice of p_i. We obtain from (3.4) for

$$(3.9) \qquad p_i = v_i : \frac{\partial G}{\partial v_i} = \sum_{\ell=1}^{\ell_*} h_\ell \lambda_\ell^T \frac{\partial \varphi_\ell}{\partial v_i} + \lambda_0^T \frac{\partial x_0}{\partial v_i} + \frac{\partial g}{\partial v_i},$$

$$(3.10) \qquad p_i = w_i \neq \{\tau_1, \ldots, \tau_{M-1}\} : \frac{\partial G}{\partial w_i} = \sum_{\ell=1}^{\ell_*} h_\ell \lambda_\ell^T \frac{\partial \varphi_\ell}{\partial w_i} + \frac{\partial g}{\partial w_i},$$

$$(3.11) \qquad p_i = \tau_i : \frac{\partial G}{\partial \tau_i} = \sum_{\ell=1}^{\ell_*} h_\ell \lambda_\ell^T \frac{\partial \varphi_\ell}{\partial \tau_i} + \lambda_{\ell(i)}^T \varphi_{\ell(i)} - \lambda_{\ell(i)+1}^T \varphi_{\ell(i)+1} -$$
$$- \sum_{\ell=1}^{\ell_*} \sum_{j=0}^{k_\ell} \frac{\partial a_{\ell j}}{\partial \tau_i} x_{\ell-j}, \quad \text{where } \tau_i = t_{\ell(i)} < t_{\ell_*}.$$

For the parametrizations considered in Example 2.1, the summands in (3.10) and (3.11) vanish for $\ell \notin \{\ell(i), \ldots, \ell(i+1)\}$ if $\tau_i = t_{\ell(i)}$ and $w_i = u_{i+1}$ (or $w_i \in \{u_{i+1}^+, u_{i+1}^-\}$).

Next we turn to the particular case that (3.2) is a (self-starting) linear multistep method, i.e., that φ_ℓ takes the form (2.6).

Corollary 3.4 *Under the assumptions of Theorem* 3.1, *we have for* (3.5), (3.6) *in case of linear multistep methods*:

$$(3.12) \qquad \lambda_{\ell_*} = \Big(\frac{\partial g}{\partial x}\Big)^T + h_{\ell_*} b_{\ell_* 0} \Big(\frac{\partial f}{\partial x}\Big|_{\ell_*}\Big)^T \lambda_{\ell_*}$$

$$(3.13) \qquad \sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{k_\ell^*} a_{\ell+j,j} \lambda_{\ell+j} = \Big(\frac{\partial f}{\partial x}\Big|_\ell\Big)^T \sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{k_\ell^*} h_{\ell+j} b_{\ell+j,j} \lambda_{\ell+j},$$
$$j = \ell_* - 1, \ldots, 0, \; h_0 := 0,$$

where k_ℓ^* *is the dual order* (3.8).
Furthermore, the following formulae for the partial derivatives of φ_ℓ *in* (3.9) – (3.11) *hold*:

$$\frac{\partial \varphi_\ell}{\partial v_i} = \sum_{j=0}^{k_\ell} b_{\ell j} \frac{\partial f}{\partial v_i}\Big|_{\ell-j}, \quad \frac{\partial \varphi_\ell}{\partial w_i} = \sum_{j=0}^{k_\ell} b_{\ell j} \frac{\partial f}{\partial u}\Big|_{\ell-j} \frac{\partial u(t_{\ell-j}, w)}{\partial w_i},$$

$$\frac{\partial \varphi_\ell}{\partial \tau_i} = \sum_{j=0}^{k_\ell} \left(\frac{\partial b_{\ell j}}{\partial \tau_i} f(x_{\ell-j}, u(t_{\ell-j}, w), v) + b_{\ell j} \frac{\partial f}{\partial u}\Big|_{\ell-j} \frac{\partial u(t_{\ell-j}, w)}{\partial \tau_i} \right)$$
$$(\ell = 1, \ldots, \ell_*).$$

(Here we denote $\frac{\partial f}{\partial x}\Big|_\ell = \frac{\partial f}{\partial x}(x_\ell, u(t_\ell, w), v)$, $\frac{\partial f}{\partial u}\Big|_{\ell-j} = \frac{\partial f}{\partial u}(x_{\ell-j}, u(t_{\ell-j}, w), v)$ etc.)

Proof: Since $\varphi_{\ell+j}(x_{\ell+j}, \ldots, x_{\ell+j-k_{\ell+j}}; p) = \sum_{i=0}^{k_{\ell+j}} b_{\ell+j,i} f(x_{\ell+j-i}, u(t_{\ell+j-i}, w), v)$, we obtain $\frac{\partial \varphi_{\ell+j}}{\partial x_\ell} = b_{\ell+j,j} \frac{\partial f}{\partial x}(x_\ell, u(t_\ell, w), v)$ whenever $0 \le j \le k_{\ell+j}$. The formulae for the partial derivatives of φ_ℓ are immediate. □

Remark 3.5 If the self-starting linear multistep method has the property that the order of the method at each step is increased at most by one (i.e. $k_\ell \le k_{\ell-1} + 1$, $\ell = 2, \ldots, N$), the constraint $j \le k_{\ell+j}$ on the sums in (3.13) is redundant.

Example 3.6 Let us consider the *piecewise constant parametrization* in Example 2.1 (i.e. $w := (u_1, \ldots, u_M; \tau_1, \ldots, \tau_{M-1})$) and an *Adams-type variable linear multistep scheme* (i.e. $a_{\ell 0} := 1$, $a_{\ell 1} := -1$ and $a_{\ell j} := 0$ for all $j = 2, \ldots, k_\ell$, $\ell = 1, \ldots, \ell_*$). Then the partial derivatives of G have the following form:

$$\frac{\partial G}{\partial v_i} = \sum_{\ell=1}^{\ell_*} h_\ell \lambda_\ell^T \sum_{j=0}^{k_\ell} b_{\ell j} \frac{\partial f}{\partial v_i}(x_{\ell-j}, u(t_{\ell-j}, w), v) + \lambda_0^T \frac{\partial x_0}{\partial v_i} + \frac{\partial g}{\partial v_i} \quad (i = 1, \ldots, s),$$

$$\frac{\partial G}{\partial u_i} = \sum_{\ell=\ell(i)+1}^{\ell(i+1)} h_\ell \lambda_\ell^T \sum_{j=0}^{k_\ell} b_{\ell j} \frac{\partial f}{\partial u}(x_{\ell-j}, u_i, v), \text{ if } u(\cdot, w) = u_i \text{ in } (t_{\ell(i)}, t_{\ell(i+1)}],$$
$$(i = 1, \ldots, M),$$

$$\frac{\partial G}{\partial \tau_i} = \sum_{\ell=\ell(i)}^{\ell(i+1)} h_\ell \lambda_\ell^T \sum_{j=0}^{k_\ell} \frac{\partial b_{\ell j}}{\partial \tau_i} f(x_{\ell-j}, u(t_{\ell-j}, w), v) + \lambda_{\ell(i)}^T \varphi_{\ell(i)} - \lambda_{\ell(i)+1}^T \varphi_{\ell(i)+1}$$

where $\tau_i = t_{\ell(i)}$, $\varphi_\ell = \sum_{j=0}^{k_\ell} b_{\ell j} f(x_{\ell-j}, u(t_{\ell-j}, w), v)$, $\ell \in \{\ell(i), \ell(i)+1\}$,

$$\lambda_{\ell_*} = \Big(\frac{\partial g}{\partial x}\Big)^T + h_{\ell_*} b_{\ell_* 0} \Big(\frac{\partial f}{\partial x}(x_{\ell_*}, u(t_{\ell_*}, w), v)\Big)^T \lambda_{\ell_*}$$

$$\lambda_\ell = \lambda_{\ell+1} + \Big(\frac{\partial f}{\partial x}(x_\ell, u(t_\ell, w), v)\Big)^T \sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{k_{\ell_*}} h_{\ell+j} b_{\ell+j,j} \lambda_{\ell+j}, \quad h_0 := 0 \tag{3.14}$$

$$(\ell = \ell_* - 1, \ldots, 0)$$

When a particular implementation of variable stepsize multistep methods is used (e.g. in [11] based on the Nordsieck representation, cf. also [26], p. 417-419), the computation of derivatives has to be adapted, of course, to this implementation.

Remark 3.7 The formulae for the partial derivatives of G in Example 3.6 are discrete analogues of the formulae derived e.g. in [38]. The discrete adjoint equations (3.12), (3.13) or (3.14) show that they are different from discretized dual variables obtained by applying the same linear multistep method to integrate the dual equation (3.3). Even for linear one-step methods

$$x_\ell = x_{\ell-1} + h_\ell \{ b f(x_\ell, u(t_\ell, w), v) + (1-b) f(x_{\ell-1}, u(t_{\ell-1}, w), v) \},$$

we obtain, by specifying (3.14), the scheme

$$\lambda_\ell = \lambda_{\ell+1} + \Big(\frac{\partial f}{\partial x}(x_\ell, u(t_\ell, w), v)\Big)^T (h_\ell b \lambda_\ell + h_{\ell+1}(1-b)\lambda_{\ell+1}) \tag{3.15}$$

and observe the mentioned difference. For the explicit Euler method (i.e. $b := 0$), (3.15) appears in several papers (e.g. [16], [18]). For the implicit Euler scheme ($b := 1$), (3.15) was used in [28].

For general one-step methods ($k_\ell := 1$ in (3.2)) Theorem 3.1 immediately yields the following specification, which was stated, in parts, already in [36].

Corollary 3.8 *Under the assumptions of Theorem* 3.1, *we have for* (3.5), (3.6) *in case of one-step methods*:

$$\lambda_{\ell_*} = \Big(\frac{\partial g}{\partial x}\Big)^T + h_{\ell_*} \Big(\frac{\partial \varphi_{\ell_*}}{\partial x_{\ell_*}}\Big)^T \lambda_{\ell_*} \tag{3.16}$$

$$\lambda_\ell - \lambda_{\ell+1} = h_\ell \Big(\frac{\partial \varphi_\ell}{\partial x_\ell}\Big)^T \lambda_\ell + h_{\ell+1} \Big(\frac{\partial \varphi_{\ell+1}}{\partial x_\ell}\Big)^T \lambda_{\ell+1}, \tag{3.17}$$

$$\ell = \ell_* - 1, \ldots, 0, \; h_0 := 0.$$

In addition, the formulae (3.9), (3.10) *are valid and* (3.11) *specifies to* $\frac{\partial G}{\partial \tau_i} = \sum_{\ell=\ell(i)}^{\ell(i)+1} h_\ell \lambda_\ell^T \frac{\partial \varphi_\ell}{\partial \tau_i} + \lambda_{\ell(i)}^T \varphi_{\ell(i)} - \lambda_{\ell(i)+1}^T \varphi_{\ell(i)+1}$ (*where* $\tau_i = t_{\ell(i)}$).

Example 3.9 For *explicit Runge-Kutta methods* (2.7) the function φ_ℓ has the form

$$\varphi_\ell(x_{\ell-1}) := \sum_{i=1}^q b_i K_{\ell i}, \; K_{\ell i} = f(x_{\ell-1} + h_\ell \sum_{j=1}^{i-1} \alpha_{ij} K_{\ell j}, u(t_{\ell-1} + c_i h_\ell, w), v),$$

$i = 1, \ldots, q$, and $\sum_{i=1}^q b_i = 1$, $\sum_{j=1}^{i-1} \alpha_{ij} = c_i$, $i = 1, \ldots, q$, q is the number of stages. From Corollary 3.8 we obtain the following discrete adjoint system:

$$\lambda_{\ell_*} = \Big(\frac{\partial g}{\partial x}\Big)^T$$

$$\lambda_\ell = \lambda_{\ell+1} + h_{\ell+1} \Big(\sum_{i=1}^q b_i Y_{\ell+1,i}^T \Big) \lambda_{\ell+1}, \quad \ell = \ell_* - 1, \ldots, 0, \tag{3.18}$$

where

$$Y_{\ell+1,i} = \frac{\partial f}{\partial x}(x_\ell + h_{\ell+1} \sum_{j=1}^{i-1} \alpha_{ij} K_{\ell+1,j}, u(t_\ell + c_i h_{\ell+1}, w), v)(I + h_{\ell+1} \sum_{j=1}^{i-1} \alpha_{ij} Y_{\ell+1,j})$$

$i = 1, \ldots, q$. Similar expressions arise for the partial derivatives $\frac{\partial \varphi_\ell}{\partial v_i}$, $\frac{\partial \varphi_\ell}{\partial w_i}$, $\frac{\partial \varphi_\ell}{\partial \tau_i}$. For specific Runge-Kutta schemes (with $q = 2$, $q = 4$ stages) discrete adjoint schemes of the form (3.18) are derived already in [7], [17], [18], [22]. For the Heun scheme

$$x_\ell = x_{\ell-1} + \frac{h_\ell}{2} \Big\{ f(x_{\ell-1}, u(t_{\ell-1}, w), v) + f\Big(x_{\ell-1} + \tfrac{1}{2} h_\ell f(x_{\ell-1}, u(t_{\ell-1}, w), v), u(t_{\ell-1} + \tfrac{1}{2} h_\ell, w), v\Big) \Big\}$$

we obtain for instance the following discrete adjoints:

$$\lambda_\ell = \lambda_{\ell+1} + \frac{h_{\ell+1}}{2} \{ Y_{\ell+1,1} + Y_{\ell+1,2} \}^T \lambda_{\ell+1}, \quad \ell = \ell_* - 1, \ldots, 0, \text{ where}$$

$$Y_{\ell+1,1} := \frac{\partial f}{\partial x}(x_\ell, u(t_\ell, w), v),$$

$$Y_{\ell+1,2} := \frac{\partial f}{\partial x}\Big(x_\ell + \tfrac{1}{2} h_{\ell+1} f(x_\ell, u(t_\ell, w), v), u(t_\ell + \tfrac{1}{2} h_{\ell+1}, w), v\Big) \cdot (I + \tfrac{1}{2} h_{\ell+1} Y_{\ell+1,1})$$

Remark 3.10 Formulae for the gradients and the discrete adjoints can be derived from Corollary 3.8 for implicit Runge-Kutta and extrapolation methods, too. For example, we obtain for the implicit midpoint rule

$$x_\ell = x_{\ell-1} + h_\ell f\Big(\tfrac{1}{2}(x_{\ell-1} + x_\ell), u(\tfrac{1}{2}(t_{\ell-1} + t_\ell), w), v\Big), \quad \ell = 1, \ldots, \ell_*,$$

the following discrete adjoint system:

$$\lambda_\ell = \lambda_{\ell+1} + \frac{h_\ell}{2}\Big(\frac{\partial f}{\partial x}\Big(\tfrac{1}{2}(x_{\ell-1}+x_\ell), u(\tfrac{1}{2}(t_{\ell-1}+t_\ell), w), v\Big)\Big)^T \lambda_\ell + \frac{h_{\ell+1}}{2}\Big(\frac{\partial f}{\partial x}\Big(\tfrac{1}{2}(x_\ell+x_{\ell+1}), u(\tfrac{1}{2}(t_\ell+t_{\ell+1}), w), v\Big)\Big)^T \lambda_{\ell+1},$$
$$\ell = \ell_* - 1, \ldots, 0.$$

Since extrapolation methods represent explicit Runge-Kutta schemes ([26], Chapt. II.9), the formulae in Example 3.9 are also relevant to these methods. However, the adjoint system for extrapolation methods (e.g. for the Gragg-Bulirsch-Stoer algorithm) can also be derived in a direct way, of course.

Remark 3.11 A comparison of the discrete adjoint systems (3.14) and (3.18) (for linear multistep and Runge-Kutta methods, respectively) shows the somewhat surprising effect that one backward step in (3.14) requires only one evaluation of the Jacobian $\frac{\partial f}{\partial x}$, whereas the number of evaluations of $\frac{\partial f}{\partial x}$ in (3.18) corresponds to the number of stages. This effect might indicate that linear multistep schemes are more advantageous compared to nonlinear methods in this context.

Remark 3.12 The methodology of this section applies to *parameteridentification* in ode's, too. In order to illustrate this, let us consider the following problem:

Minimize $F(p) := \frac{1}{2}\sum_{k=1}^{M} \|x(\tau_k) - y_k\|^2$ subject to $p \in P \subseteq \mathbb{R}^s$,

where $x'(t) = f(x(t), p)$, $t \in [0,1]$, $x(0) = x_0(p)$, and

y_k are observations at the data points $\tau_k \in [0,1]$, $k = 1, \ldots, M$.

If the ode is solved by the general variable multistep (3.2) and the data points τ_k are included into the underlying grid (i.e. $\tau_k = t_{\ell(k)}$, $k = 1, \ldots, M$), we obtain from Theorem 3.1 the following formulae for partial derivatives of F:

$$\frac{\partial F}{\partial p_i} = \sum_{k=1}^{M}\Big[\sum_{\ell=1}^{\ell(k)} h_\ell \lambda_{k,\ell}^T \frac{\partial \varphi_\ell}{\partial p_i} + \lambda_{k,0}^T \frac{\partial x_0}{\partial p_i}\Big] \quad (i = 1, \ldots, s)$$

where

$$\lambda_{k,\ell(k)} = (x(\tau_k) - y_k) + h_{\ell(k)}\Big(\frac{\partial \varphi_{\ell(k)}}{\partial x_{\ell(k)}}\Big)^T \lambda_{k,\ell(k)}$$

$$\sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{k_\ell^*} a_{\ell+j,j}\lambda_{k,\ell+j} = \sum_{\substack{j=0 \\ j \le k_{\ell+j}}}^{k_\ell^*} h_{\ell+j}\Big(\frac{\partial \varphi_{\ell+j}}{\partial x_\ell}\Big)^T \lambda_{k,\ell+j},$$
$$\ell = \ell(k) - 1, \ldots, 0,\ h_0 := 0 \quad (k = 1, \ldots, M).$$

4 Conclusions

A general approximation scheme for nonlinear constrained optimal control problems is considered, where the controls are suitably parametrized ("simple") functions and the state equation is replaced by a discretization scheme. These schemes lead to finite-dimensional nonlinear programs (NLP), which can be solved numerically by standard methods. An important problem for applying NLP-solvers consists in the efficient computation of first (and sometimes also higher) order derivatives of the objective and constraints. The present paper contributes to the direct development of the discrete adjoint system from the discretized model, which is called "internal numerical differentiation via adjoint systems" in [7]. This approach represents a variant of the "reverse mode of automatic differentiation" (cf. [24]). Its underlying idea consists in saving information (on the discrete states and the discretization scheme) during the (forward) integration of the state equation. This information is then used to compute discrete adjoint variables and, after that, first oder derivatives. The results in Section 3 provide some insight into the way in which discrete adjoints can be computed from the information obtained during the forward integration. Although the reverse mode of automatic differentiation enjoys an excellent temporal complexity result (cf. [23], [24]), its practical applicability seemed to be limited by an increase of the memory requirement, which is proportial to the run time T of the function evaluation program (i.e. of the forward ode solver). However, a recently developed recursive checkpointing scheme ([23]) achieves a logarithmic growth of temporal and spatial complexity relative to T in reverse automatic differentiation.

Finally, we mention that a system for solving constrained optimal control problems has been developed and described in [3], [4]. This system is based on an implementation of the piecewise constant and piecewise linear parametrization in Example 2.1, the integration of the state equation by Adams methods or backward differentiation formulae (in the nonstiff or stiff case) in the implementation from [11] and on solving the nonlinear programs by SQP-methods ([39], [40]). The computation of gradients is carried out via discrete adjoint systems according to the formulae in Corollary 3.4,

Example 3.6. First test runs of the system have been performed on a set of small but illustrative examples (control of a point mass moving [21], mixed catalyst problem [38], rocket start [18], moon landing of a spacecraft [4]) with or without including grid points into the optimization process. The results in [3], [4] show the reliability and efficiency of the technique for computing gradients described in this paper.

Acknowledgement:

The second author wishes to thank H.G. Bock (Heidelberg), A. Griewank (Dresden) and O. von Stryk (München) for fruitful discussions.

References

[1] W. Alt, On the approximation of infinite optimization problems with an application to optimal control problems, Applied Mathematics and Optimization 12 (1984), 15-27.

[2] W. Alt and K. Malanowski, The Lagrange-Newton method for nonlinear optimal control problems, Computational Optimization and Applications 2 (1993), 77-100.

[3] M.L. Baguer, Entwicklung eines Programmsystems zur numerischen Lösung von Steuerungsproblemen bei gewöhnlichen Differentialgleichungen, Dissertation, Humboldt-Universität Berlin, Fachbereich Mathematik, 1990.

[4] M.L. Baguer, A system for the numerical solution of constrained optimal control problems, manuscript, Universidad La Habana, Facultad de Matematica, 1994.

[5] H.G. Bock, Numerical treatment of inverse problems in chemical reaction kinetics, in: Modelling of Chemical Reaction Systems (K.H. Ebert, P. Deuflhard and W. Jäger Eds.), Springer-Verlag, Berlin 1981, 102-125.

[6] H.G. Bock, Recent advances in parameteridentification techniques for o.d.e., in: Numerical Treatment of Inverse Problems in Differential and Integral Equations (P. Deuflhard, E. Hairer Eds.), Birkhäuser, Boston 1983, 95-121.

[7] H.G. Bock, Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen, Bonner Mathematische Schriften 183, Bonn 1987.

[8] C. de Boor, Splinefunktionen, Birkhäuser, Basel 1990.

[9] B.M. Budak and F.P. Vasilev, Some computational aspects of optimal control problems (in Russian), Moscow University Press, Moscow, 1975.

[10] R. Bulirsch, Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung, Report of the Carl-Cranz Gesellschaft, Oberpfaffenhofen 1971.

[11] G.D. Byrne and A.C. Hindmarsh, A polyalgorithm for the numerical solution of ordinary differential equations, ACM Transactions on Mathematical Software 1 (1975), 71-96.

[12] R. Callies, Optimal design of a mission to Neptune, in: Optimal Control (R. Bulirsch, A. Miele, J. Stoer and K.H. Well Eds.), Birkhäuser, Basel 1993, 341-349.

[13] L. E. Cuesta and M. Gelbrich, Convergence rates of discretizations for constrained nonlinear optimal control problems with C^1 data, submitted to the Proceedings of the Second International Conference "Approximation and Optimization" in the Caribbean, Havana 1993.

[14] J. Cullum, Finite-dimensional approximations of state-constrained continuous optimal control problems, SIAM Journal Control 10 (1972), 649-670.

[15] A.L. Dontchev, Discrete approximations in optimal control, in: Nonsmooth Analysis and Geometric Methods in Optimal Control (B.S. Mordukhovich, R.T. Rockafellar and H. Sussmann Eds.), Proceedings IMA, Springer (to appear).

[16] A.L. Dontchev and W.W. Hager, Lipschitz stability in nonlinear control and optimization, SIAM Journal Control and Optimization 31 (1993), 569-603.

[17] P.J. Enright and B.A. Conway, Discrete approximation to optimal trajectories using direct transcription and nonlinear programming, AIAA Paper 90-2963-CP (1990).

[18] Yu.G. Evtushenko, Numerical Optimization Techniques, Optimization Software Inc., New York 1985.

[19] Yu.G. Evtushenko, Automatic differentiation viewed from optimal control theory, in [22], 25-30.

[20] C.J. Goh and K.L. Teo, Control parametrization: a unified approach to optimal control problems with linear constraints, Automatica 24 (1988), 3-18.

[21] C.J. Goh and K.L. Teo, MISER: a FORTRAN program for solving optimal control problems, Adv. Engineering Software 10 (1988), 90-99.

[22] N.I. Grachev and Yu.G. Evtushenko, A library of programs for the solution of optimal control problems (in Russian), Ž. Vyčisl. Mat. i. Mat. Fiz. 19 (1979), 367-387.

[23] A. Griewank, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, Optimization Methods and Software 1 (1992), 35-54.

[24] A Griewank and G.F. Corliss (Eds.), Automatic Differentiation of Algorithms: Theory, Implementation and Application, SIAM, Philadelphia 1991.

[25] W. Grimm, Convergence relations between optimal control and optimal parametric control, DFG-Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung". Report No. 420, 1993.

[26] E. Hairer, S.P. Nørsett and G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Second Edition, Springer-Verlag, Berlin 1993.

[27] P. Hiltmann, Numerische Lösung von Mehrpunkt-Randwertproblemen und Aufgaben der optimalen Steuerung mit Steuerfunktionen über endlichdimensionalen Räumen, DFG-Schwerpunktprogramm "Anwendungsbezogene Optimierung und Steuerung", Report No. 448, 1993.

[28] A. Isa and W. Römisch, Zur numerischen Behandlung der Optimierung eines Polymerisationsprozesses, in: 4. Fachtagung "Numerische Realisierung mathematischer Modelle", ZfR-Informationen 81.16, ZfR der Akad. Wiss. DDR, 1981, 71-84.

[29] D. Kraft, Finite-difference gradients versus error-quadrature gradients in the solution of parametrized optimal control problems, Optimal Control Applications & Methods 2 (1981), 191-199.

[30] D. Kraft, On converting optimal control problems into nonlinear programming problems, in: Computational Mathematical Programming (K. Schittkowski Ed.), NATO ASI Series 15, Springer-Verlag, Berlin 1985, 261-280.

[31] K. Malanowski, Finite difference approximations to constrained optimal control problems, in: Optimization and Optimal Control (A. Auslender, W. Oettli and J. Stoer Eds.), Lecture Notes in Control and Information Sciences Vol. 30, Springer-Verlag, Berlin 1981, 243-254.

[32] H. Maurer, Numerical solution of singular control problems using multiple shooting techniques, Journal Optimization Theory and Applications 18 (1976), 235-257.

[33] B.S. Mordukhovich, On difference approximations of optimal control systems, Journal Applied Mathematics and Mechanics 42 (1979), 452-461.

[34] M.C. Perez, Indirect methods for singular control problems: a differential algebraic equation approach, submitted to the Proceedings of the Second International Conference "Approximation and Optimization" in the Caribbean, Havana 1993.

[35] C. Richter and B. Renner, Efficient methods for solving optimal control problems, in: Numerical Methods of Nonlinear Programming and their Implementations (C. Richter, H. Hollatz and D. Pallaschke Eds.), Akademie-Verlag, Berlin 1991, 99-109.

[36] W. Römisch, Remarks on the numerical treatment of optimal control problems, Berichte des IX. Internationalen Kongresses für Anwendungen der Mathematik, Heft 2, Weimar 1981, 82-86.

[37] E.W. Sachs, Topics in modern computational methods for optimal control problems, In: Modern Methods of Optimization (W. Krabs and J. Zowe Eds.), Lecture Notes in Economics and Mathematical Systems Vol. 378, Springer-Verlag, Berlin 1992, 313-348.

[38] R.W.H. Sargent and G.R. Sullivan, The development of an efficient optimal control package, in: Optimization Techniques (J. Stoer Ed.),

Lecture Notes in Control and Information Sciences Vol. 7, Springer-Verlag, Berlin 1978, 158-168.

[39] K. Schittkowski, NLPQL: A FORTRAN-subroutine solving constrained nonlinear programming problems, Annals of Operations Research 5 (1985/86), 485-500.

[40] K. Schittkowski, Solving nonlinear programming problems with very many constraints, Optimization 25 (1992), 179-196.

[41] H.R. Sirisena, Computation of optimal controls using a piecewise polynomial parametrization, IEEE Transactions on Automatic Control 18 (1973), 409-411.

[42] H.R. Sirisena and F.S. Chou, Convergence of the control parametrization Ritz method for nonlinear optimal control problems, Journal Optimization Theory and Applications 29 (1979), 369-382.

[43] O. von Stryk, Numerical solution of optimal control problems by direct collocation, in: Optimal Control (R. Bulirsch, A. Miele, J. Stoer and K.H. Well Eds.), Birkhäuser, Basel 1993, 129-143.

[44] O. von Stryk and R. Bulirsch, Direct and indirect methods for trajectory optimization, Annals of Operations Research 37 (1992), 357-373.

[45] K.L. Teo, Convergence analysis for a computational method for optimal control, Problems of Control and Information Theory 16 (1987), 315-328.

[46] K.L. Teo, C.J. Goh and K.H. Wang, A unified computational approach to optimal control problems, Longman, Harlow 1991.

[47] J. Vlassenbroeck and R. van Doren, A Chebyshev technique for solving nonlinear optimal control problems, IEEE Transactions on Automatic Control 33 (1988), 333-340.

The moment problem associated with the Cowen tridiagonal

D. Barrios, V. Tomeo* and E. Torrano
Facultad de Informática, U. Politécnica de Madrid
Escuela de Estadística, U. Complutense de Madrid*

Abstract

In this paper we solve the hermitian moment problem associated to a non-symmetric tridiagonal matrix depending on two parameters. The problem has a solution when the operator on ℓ^2 defined by this matrix is subnormal. We analyze the spectrum of the operator and also the spectrum of its normal extension for several values of the parameters. We get the support of the measure, and the distribution in some case.

1 Introduction

Let us consider an infinite positive definite hermitian matrix $M = (c_{ij})_{i,j=0}^{\infty}$ generated by $c_{ij} =< D^i e_0, D^j e_0 >$ with $e_0^t = (1, 0, 0, \ldots)$, D is the tridiagonal operator $D = T + \lambda T^*$, where

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & \sqrt{1+s} & 0 & 0 & 0 & \cdots \\ 0 & 0 & \sqrt{1+s+s^2} & 0 & 0 & \cdots \\ 0 & 0 & 0 & \sqrt{1+s+s^2+s^3} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

with $\lambda \in \mathbb{C}$ and $0 \leq s \leq I$. Cowen proved [6] that D is a subnormal operator if and only if $\lambda = 0$ or $|\lambda| = s^{k/2}$ with $k \in \{0, 1, 2, \ldots\}$.

We know [2] that if the right-shift operator associated to a bounded hermitian moment problem is subnormal, the problem to find μ and Ω such that