A MULTIPLE SHOOTING ALGORITHM FOR DIRECT SOLUTION OF OPTIMAL CONTROL PROBLEMS*

H. G. Bock and K. J. Plitt

Institut für Angewandte Mathematik, SFB 72, Universität Bonn, 5300 Bonn, Federal Republic of Germany

Abstract. An algorithm for the numerical solution of parameterized optimal control problems is presented, which is based on *multiple shooting* in connection with a *recursive quadratic programming* technique. A *condensing algorithm* for the solution of the approximating linearly constrained quadratic subproblems, and *high rank update* procedures are introduced, which are especially suited for optimal control problems and lead to significant improvements of the convergence behaviour and reductions of computing time and storage requirements. The algorithm is *completely derivative-free* due to internal numerical differentiation schemes, it can be conveniently combined with indirect multiple shooting. Numerical results are given in the field of aerospace engineering.

Keywords. Optimal control, numerical methods, nonlinear programming, quadratic programming, aerospace trajectories, multiple shooting.

INTRODUCTION

There are at least two basically different ways of solving optimal control problems. In the *indirect* approach, the controls are expressed by the maximum principle in terms of state and adjoint variables, which can be computed by solving a possibly very intricate multipoint boundary value problem with jumps and switching conditions.

In recent years reliable, stable and efficient numerical algorithms have been developed for the solution of this general class of problems, based on the multiple shooting technique (e.g.[2 , 6 , 7 , 8 ,13]), which actually made accessible the wide applicability of the indirect approach, which includes control- or state-constrained and Chebyshev-problems as well as feed-back control (e.g.[4 , 5]).

The present paper introduces a numerical algorithm for the *direct* approach, which solves the optimal control problem directly in terms of control and state variables. In the new method, a multiple shooting parameterization of the state differential equations is coupled with a simultaneous control parameterization. This leads to a large constrained finite optimization problem, for which a specially suited recursive quadratic programming algorithm with new high rank update formulae is developed, that leads to a substantical improvement of performance compared to previous direct approaches. The algorithm is globally convergent, its local convergence is superlinear with an asymptotic convergence rate that is essentially independent of the meshsize used in the parameterization.

* The research was supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 72.

In view of practical applications, it is one of the most important properties of the new algorithm that it is completely derivative-free (due to internal numerical differentiation schemes). *This means, that any analytical preparations* (such as derivation of adjoint equations) *are strictly avoided.*

PARAMETERIZATION OF CONTROL PROBLEMS BY MULTIPLE SHOOTING

For ease of presentation, we consider the *constrained optimal control* problem ($x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^k$)

$$[P1] \quad J(x,u) = \int_{t_o}^{t_f} f(t,x(t),u(t))dt = \min, \quad (1)$$

$$\dot{x} = f(t,x(t),u(t)), \; t \in [t_o,t_f], \quad (2)$$

$$r(x(t_o),x(t_f)) = 0 \text{ or } \geq 0 , \quad (3)$$

$$g(t,u(t)) \geq 0, \quad t \in [t_o,t_f]. \quad (4)$$

The actual implementation of our method provides also a special treatment of free parameters and the free-end time case. Moreover, discontinuous right hand sides could be included as well as multipoint boundary conditions (b.c.)(and thus state constraints).

Multiple Shooting Parameterization

The basic concept of the new approach is the reduction of the infinite dimensional problem [P1] to a finite dimensional optimization problem by a *multiple shooting* technique. For a suitably chosen mesh

$$t_o < t_1 < \ldots < t_{m-1} < t_m = t_f, \quad (5)$$

the control vector is approximated on every subinterval by a finite set of parameters using given basis functions ϕ_j

$$u(t) \equiv \phi_j(t,q_j),\ q_j \in \mathbb{R}^{k_j},\qquad t \in I_j := [t_j,t_{j+1}],\ j=1(1)m-1, \tag{6}$$

e.g., piecewise polynomials. The state variables on every subinterval are replaced by the computed solution $x(t;s_j,q_j)$ of the initial value problems (IVP)

$$\dot{x} = f(t,x,\phi_j(t,q_j)),\qquad x(t_j) = s_j,\quad t \in I_j \tag{7}$$

as function of s_j and q_j. Problem [P1] is thus transformed to a constrained finite optimization problem with respect to the parameter vector $y:=(q_o,..,q_{m-1},s_o,..,s_m)$:

$$\text{[P2]}\quad \sum_{j=0}^{m-1} \int_{t_j}^{t_{j+1}} L(t,x(t;s_j,q_j),\phi_j(t,q_j))dt \tag{8}$$

$$=: J(y) = \min$$

$$r(s_o,s_m) = 0 \ \text{ or } \ \geq 0, \tag{9}$$

$$g_j(q_j) \geq 0 \tag{10}$$

with the additional *matching conditions*

$$x(t_{j+1};s_j,q_j)-s_{j+1}=:h_j(s_j,s_{j+1},q_j)=0 \tag{11}$$

which ensure continuity of the solution trajectory.

Up to now, finite dimensional nonlinear programs have been derived from [P1] exclusively by initial value simulation or *single shooting*, i.e. m=1. Advantages of increasing the size of the nonlinear program [P1] by the extra variables s_j are, firstly, those already proved theoretically and practically for *multiple shooting* in two- or multipoint BVP and parameter estimation problems - such as improved convergence in nonlinear problems and *numerical stability* by preventing the growth of the error introduced by poor initial data, discretization or round-off, and propagated by inherent instabilities of the o.d.e. system.

Secondly, there are also distinct advantages of multiple shooting over single shooting from the optimization point of view. This will be shown in the following.

A RECURSIVE QUADRATIC PROGRAMMING METHOD FOR MULTIPLE SHOOTING

Problem [P2] represents a relatively large, constrained optimization problem of the class

$$\text{[P3]}\quad F_1(y) = \min \tag{12}$$

$$F_2(y) = 0 \tag{13}$$

$$F_3(y) \geq 0, \tag{14}$$

which is solved by a Quasi-Newton method outlined in the following. A given estimate y^k is iteratively improved by

$$y^{k+1} = y^k + t_k\Delta y^k,\quad t_k \in [t_{min},t_{max}], \tag{15}$$

where the increment Δy^k is determined by the Kuhn-Tucker solution $(\Delta y^k,\lambda_1^k,\lambda_2^k)$ of a quadratic approximation to [P3].

$$\text{[P4]}\quad \frac{1}{2}\Delta y^T B^k \Delta y + \nabla F_1^k \Delta y \tag{16}$$

$$=: Q_k(\Delta y) = \min$$

$$F_2^k + \nabla F_2^k \Delta y = 0 \tag{17}$$

$$F_3^k + \nabla F_3^k \Delta y \geq 0. \tag{18}$$

F_i^k, ∇F_i^k are the functions and gradients,

$$F_i(y^k),\ \nabla F_i(y^k),\ (i=1,2,3), \tag{19}$$

evaluated at y^k. The $\hat{n}\times\hat{n}$ - matrix B^k

$$\hat{n} = (m+1)n + \sum_{j=0}^{m-1} k_j \tag{20}$$

is an approximation for the Hessian $\nabla_y^2 L$ of the Lagrange function

$$L(y,\lambda) = F_1(y) - \lambda_1^T F_2(y) - \lambda_2^T F_3(y), \tag{21}$$

which, starting from an initial estimate B^o, is revised in each step by an update procedure

$$B^{k+1} = B^k + U(B^k,\Delta y^k,\gamma^k) \tag{22}$$

$$\gamma^k = \nabla L(y^{k+1},\lambda_1^k,\lambda_2^k) - \nabla L(y^k,\lambda_1^k,\lambda_2^k).$$

Schemes of this type are known in nonlinear programming (cf. [1 , 9 ,12]). The special algorithm described here is also closely related with generalized Gauss-Newton methods for constrained nonlinear least squares problems developed for parameter estimation in o.d.e. ([3 , 6]).
The algorithm is well defined if the Jacobian of the active constraints of [P4] is non-singular throughout the iterations, and B^k is positive definite on its null-space (and under these conditions, global convergence can be shown).

From the details of the new method, two features are derived in the sequel: an efficient solution algorithm for [P4], and a high rate update procedure for (22), which are a consequence of the multiple shooting approach and of basic importance for the performance of the method.

A Condensing Algorithm for Recursive Solution of the Quadratic Program

Problem [P4] is typically much larger for multiple than for single shooting, but it has a sparse block structure. With the abbreviations (dropping the iteration index k)

$$G_j^s := \partial x(t_{j+1};s_j q_j)/\partial s_j,$$

$$G_j^q := \partial x(t_{j+1};s_j,q_j)/\partial q_j, \tag{23}$$

$$h_j := h_j(s_j,s_{j+1},q_j),\ j=1(1)m-1,$$

the linearized matching conditions are

$$\Delta s_{j+1} = G_j^s\Delta s_j + G_j^q\Delta q_j + h_j. \tag{24}$$

The explicit form of (24) permits an efficient *recursive* equivalence transformation of [P4] to a *condensed problem*

$$[\text{P5}]\ \frac{1}{2}\begin{pmatrix}\Delta s_o\\ \Delta q\end{pmatrix}^T \hat{B}\begin{pmatrix}\Delta s_o\\ \Delta q\end{pmatrix} + \hat{b}\begin{pmatrix}\Delta s_o\\ \Delta q\end{pmatrix} = \min \quad (25)$$

$$v_1 + E_1^s \Delta s_o + E_1^q \Delta q = 0 \quad (26)$$

$$v_2 + E_2^s \Delta s_o + E_2^q \Delta q \geq 0\ . \quad (27)$$

Here, $\hat{B}$, $\hat{b}$, E_i^s, E_i^p and v_i can be conveniently computed in a backward recursion. $\Delta q = (\Delta q_o,\ldots,\Delta q_{m-1})$ represents the control parameters. The condensed problem [P5] is now of the same size as for single shooting, and is solved by a stable and efficient QP-algorithm.

The remaining increments $s_1,\ldots,s_m$ and the Lagrange multipliers of the matching conditions - which are required for the Hessian update - can be computed by two recursions.

<u>Lemma</u>. Let $(\Delta s_o, \Delta q)$ denote the solution of the condensed problem [P5], and (μ_1,μ_2) the Lagrange multipliers corresponding to (26,27). With the supplementary definition

$$G_m^s := \partial r(s_o,s_m)/\partial s_m\ , \quad (28)$$

the Kuhn-Tucker-point $(\Delta y,\lambda)$ of [P4] is complemented by

Δs_j due to (24)

in a forward recursion for $j = 1(1)m$, and

$$\lambda_{1,j} = -\partial Q/\partial(\Delta s_{j+1}) + G_{j+1}^{s\ T}\lambda_{1,j+1} \quad (29)$$

in a backward recursion for $j = m-1(-1)0$, where $\lambda_{1,m}$ are the multipliers of the condensed problem associated with boundary conditions.

By use of this condensing algorithm the solution effort for the multiple shooting problem is of the same order as for single shooting.

<u>High rank constrained variable-metric updates</u>

For the sake of brevity, the boundary conditions are assumed to satisfy the separability condition

$$r(s_o,s_m) = r_1(s_o) + r_2(s_m) \quad (30)$$

otherwise, auxiliary parameters can be introduced). As a characteristic feature of the new approach, the Hessian B of L for [P2] can be shown to have the block structure

$$B = \begin{pmatrix} C_o & & \\ & \ddots & \\ & & C_m\end{pmatrix} \quad (31)$$

$$C_m = \partial^2 L(y)/\partial s_m^2\ ,$$

$$C_j = \partial^2 L(y)/\partial(s_j,q_j)^2\ ,\ j=0(1)m-1\ .$$

Application of any of the standard rank 2 update formulae U for revising B^k in (22) would destroy this a priori known structure, and lead to considerably increased storage requirements as well as to an increase in computing effort for the condensing algorithm. By the use of appropiate projections, however, *generalized update formulae*

$$B^{k+1} = B^k + \sum_{j=0}^{m} U(P^jB^kP^j, P^j\Delta y^k, P^j\gamma^k) \quad (32)$$

can be generated from U. (Here, P^j is the projection from y-space onto $(s_j q_j)$ or s_m-subspace). It is easy to show, that the new formula is of *rank 2(m+1)* and preserves the structure. If a generalized DFP-update is used, local *superlinear convergence* can actually be shown (similar to the classical case). The present version of the algorithm employs a generalized BFGS-update, with a modification due to [12] to maintain positive definiteness.

Extensive numerical tests as well as theoretical evidence indicate that the *asymptotic convergence rate* is considerably improved by the new high rank update procedure.

In fact, it is essentially independent of the number of meshpoints used for the control parameterization.

<u>Remark</u>. High rank generalized update formulae could also be constructed for BVP approaches other than multiple shooting, e.g. certain collocation methods.

GRADIENT GENERATION

In realistic problems, the main computational work is needed for the gradient and function evaluation step (9). In order to avoid human error and to minimize this effort, the implementation of the new method (MUSCOD) uses "internal numerical differentiation" schemes ([3 , 6]) which make the algorithm completely derivative-free.

NUMERICAL RESULTS

<u>Van der Pol (VDP) Problem</u>

The VDP problem is relatively simple and was chosen for comparison purposes.

$$J = \frac{1}{2}\int_0^5 (x_1^2 + x_2^2 + u^2)dt = \min \quad (33)$$

$$\dot{x}_1 = x_2 \qquad ,\ x_1(0)=1$$

$$\dot{x}_2 = -x_1 + (1-x_1^2)x_2 + u,\ x_2(0)=0$$

$$x_1(5) - x_2(5) + 1 = 0$$

Table 1 shows the results of MUSCOD for a piecewise linear control parameterization and for equidistant meshes of sizes m = 3,6,11,21. The initial guesses were $x_1(t_j) \equiv 1$, $x_2(t_j) \equiv 0$, and $u(t) = \phi_j(t,q_j) \equiv 0.5$

As could be expected, the value of the cost-function J decreases with increasing number of mesh points m. The number of iterations (GE), however, remains the same, or is even slightly reduced.

Due to the extremely simple differential equations, the computing time increases considerably as a consequence of the - dominating ! - overhead, and to some

TABLE 1 *VDP Problem (MUSCOD)*

Mesh#	J(x,u)	GE	FE	CPU-sec total	OH
3	2.0757	7	7	.174	.031
6	1.6907	7	8	.241	.060
11	1.6860	6	6	.305	.135
21	1.6857	6	6	.813	.524

GE: number of gradient evaluations (iterations)
FE: number of function evaluations
OH: overhead - calculations except function and gradient generation

extend also in the GE/FE - calculations, which are downgraded because the mesh-sizes for $m \geq 6$ already impose noticable stepsize restrictions on the integrator. In realistic applications, both effects are rather unlikely to happen.

Table 2 quotes the results of Pouliot, Pierson and Brusch [11] obtained by a single shooting nonlinear programming algorithm with the same control parameterization and same initial data.

Here, the number of iterations (GE) increases with m - up to five times more than needed by MUSCOD for the same accuracy - thus showing the characteristic behaviour of the initial value approach. A comparison of computing times is hardly possible, since Table 1 was computed on an IBM 3081, and Table 2 on an ITEL-AS/6, which is slower (although allegedly by less than a factor of 2).

Pouliot et al. also note an interesting comparison run with a (continuous) Gradient Projection method due to Pierson [10], which needed 33.5 sec on the same machine for a minimum of 1.6859 - which shows that their algorithm is already considerably fast for a direct method.

A comparison between direct and indirect approach is hampered by the fact that both work with hardly compatible input data. For the VDP problem, adjoint differential equations and transversality conditions are

$$\dot{\lambda}_1 = - x_1 + \lambda_2(1 + 2x_1x_2)$$
$$\dot{\lambda}_2 = - x_2 + \lambda_2(x_1^2 - 1) - \lambda_1$$
$$\lambda_1(5) + \lambda_2(5) = 0$$

and the maximum principle yields

$$u(x,\lambda) = - \lambda_2.$$

To obtain a similar starting guess, $\lambda_1(t_j) \equiv 0$, $\lambda_2(t_j) \equiv -0.5$ were chosen initially (m=11). The optimal solution was computed with a multiple shooting code ([2, 6]) also used for parameter estimation (PARFIT/OPCON). It took 5 iterations for an optimal value of $J = 1.68568$ and a computing time of 0.235 sec (including 0.085 sec overhead).

TABLE 2 *VDP Problem (Pouliot et al.)*

Mesh#	J(x,u)	GE	FE	CPU-sec total
3	2.0759	11	17	1.9
6	1.6907	13	19	2.83
11	1.6860	23	28	6.65
21	1.6857	31	36	15.02

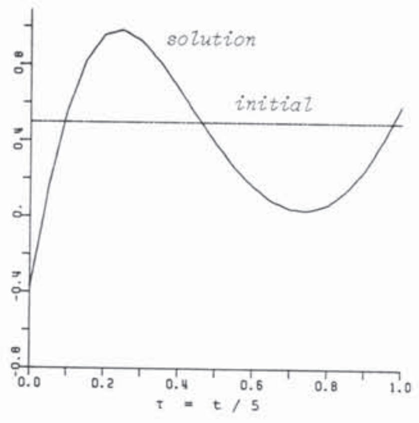

Fig. 1 VDP: control u, m = 21

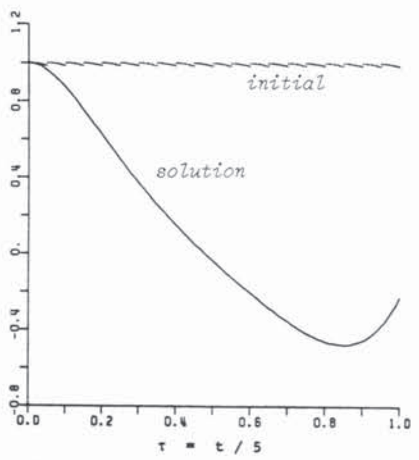

Fig. 2 VDP: state variable x_1, m = 21

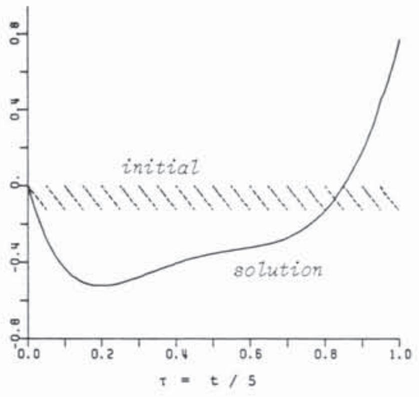

Fig. 3 VDP: state variable x_2, m = 21

These results, as well as other test runs, indicate that the direct multiple shooting algorithm MUSCOD performs similarly efficient as the indirect method for comparable initial data.

Since the relative merits of both approaches lie in different fields, they are ideally suited for a combined application. Note that MUSCOD also provides the estimates $\lambda_{1,j}$ for the adjoint variables $\lambda(t_j)$ of the continuous control problem by means of the recursion formula (29), that can be used for a subsequent call of the indirect method.

These estimates actually converge to $\lambda(t_j)$ for decreasing meshwidth. In the VDP-problem, these figures already agree to several decimals.

Reentry Problem

In order to illustrate the practical performance of the new method on a non-academic problem the numerical computation of the re-entry trajectory for an Apollo-type vehicle is presented. The problem is summarized in terms of $x = (v,\gamma,\xi)$ (cf.[13]):

$$J = \int_0^T c_o v^3 (\rho(\xi))^{\frac{1}{2}} dt = \min_{(x,u,T)} \qquad (34)$$

(convective heating)

$$\dot{v} = -c_1 \rho(\xi) v^2 C_D(u) - c_2 \frac{\sin\gamma}{(1+\xi)^2}$$

$$\dot{\gamma} = c_1 \rho(\xi) v C_L(u) - c_3 \frac{v\cos\gamma}{(1+\xi)} - \frac{c_2 \cos\gamma}{v(1+\xi)^2}$$

$$\dot{\xi} = c_4 v \sin\gamma , \qquad |u(t)| \leq \pi/2$$

and $\rho(\xi) = c_5 \exp(-c_6\xi)$, $C_D(u) = c_7 - c_8 \cos u$, $C_L(u) = c_9 \sin u$, $c_i > 0$. Both initial and end values are prescribed for every component of x. The free-end time problem is eqivalently transformed to a fixed-end time problem on [0,1], introducing T as a parameter.

The numerical results are shown in Fig.4-7. Starting with T = 230. and $u(t_j) \equiv 0$ (m=7), single shooting fails to converge (crash type initial trajectory). Multiple shooting, starting the trajectory by linear interpolation between initial and end values, converges within 33 iterations (33 GE/34FE) and 7.77 sec (including an overhead of 1.01 sec, i.e. less than 15%). The optimal value of J is 2.783, the end time T = 225.3 .

Fig. 4 - 7 △——△ initial trajectories
□——□ solution trajectories

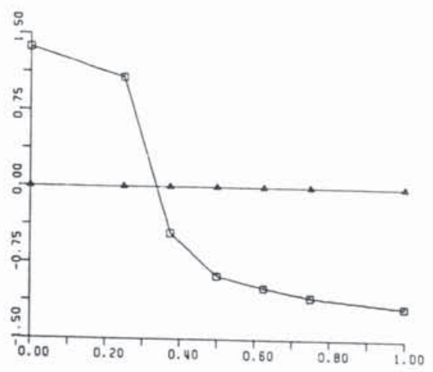

Fig. 4 Re-entry: brake cone u

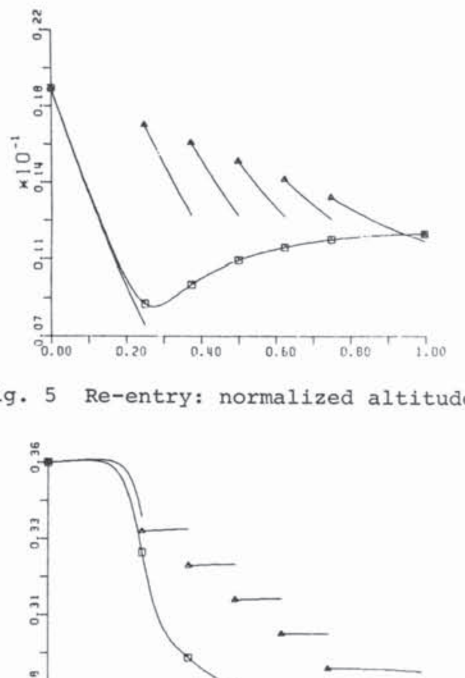

Fig. 5 Re-entry: normalized altitude ξ

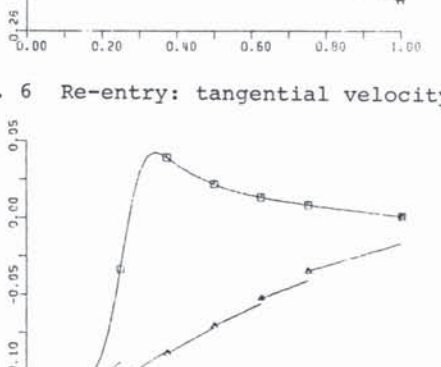

Fig. 6 Re-entry: tangential velocity v

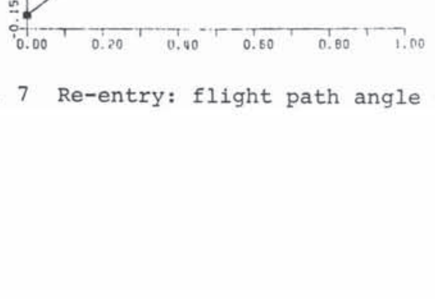

Fig. 7 Re-entry: flight path angle γ

REFERENCES

1. BIGGS, M.C. (1978). A numerical comparison between two approaches to the nonlinear programming problem. *Towards global optimization 2*, Dixon, Szego (Eds.), North-Holland, Amsterdam.
2. BOCK, H.G. (1978). Numerical Solution of Nonlinear Multipoint Boundary Value Problems with Application to Optimal Control, Z. Ang. Math. Mech. 58, 407.
3. BOCK, H.G. (1981). Numerical Treatment of Inverse Problems in Chemical Reaction Kinetics. *Modelling of Chemical Reaction Systems*, Ebert, Deuflhard, Jäger (Eds.), Springer, Berlin-Heidelberg-New York.
4. BOCK, H.G. (1981). Numerical Treatment of State-constrained and Chebyshev Control Problems (in German), Carl-Cranz-Gesellschaft: Techn. Rep., Heidelberg.
5. BOCK, H.G, P. KRÄMER-EIS (1981). An Efficient Algorithm for Approximate Computation of Feedback Control Laws in Nonlinear Processes, Z. Ang. Math. Mech. 61, 330.
6. BOCK, H.G. (1983). Recent advances in Parameter Identification Techniques for O.D.E., *Numerical Treatment of Inverse Problems*, Deuflhard, Hairer (Eds.), Birkhäuser, Boston.
7. BULIRSCH, R. (1971). The Multiple Shooting Method for Numerical Solution of Nonlinear Boundary Value Problems and Optimal Control Problems (in German), Carl-Cranz-Gesellschaft: Techn. Rep., Heidelberg.
8. DEUFLHARD, P. (1980). Recent Advances in Multiple Shooting Techniques. *Computational Techniques for Ordinary Differential Equations*, Gladwell, Sayers (Eds.), Academic Press, London - New York.
9. HAN, S.P. (1976). Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, Math. Prog. 11, 263.
10. PIERSON, B.L. (1975). Panel flutter optimization by gradient projection. International J. for Numerical Methods in Engineering, 9(2), 271.
11. POULIOT, M.R., B.L. PIERSON, R.G. BRUSCH (1981). Recursive quadratic programming solutions to minimum-time aircraft trajectory problems. 2nd IFAC Workshop on Control Appl. of Nonlinear Programming, Collected papers, Well (Ed.).
12. POWELL, M.J.D. (1978). A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. *Numerical Analysis*, Watson, G.A. (Ed.), Springer, Lecture Notes in Maths. No. 630, Berlin.
13. STOER, J., R. BULIRSCH (1973). Einführung in die numerische Mathematik II, Springer, Berlin-Heidelberg-New York.