U.S.S.R. Comput. Maths. Math. Phys., Vol.19, No.2, pp. 99–119

A LIBRARY OF PROGRAMS FOR SOLVING OPTIMAL CONTROL PROBLEMS[1]

N.I. GRACHEV and YU.G. EVTUSHENKO

Moscow

(Revised 20 March 2003)

THE MATHEMATICAL basis of a library of standard programs for the numerical solution of the general optimal control problem with mixed constraints is described. An example of the solution of an elementary problem of flight dynamics is given.

The present paper gives the mathematical basis of a library of standard programs for solving optimal control problems with mixed constraints. The methods employed utilize reduction of the initial problem to a problem of nonlinear programming. The programs are designed to solve a wide class of problems including time-optimal problems, problems with moving right-hand end, problems with non-differentiable functionals, and optimization with respect to a control vector and control parameter, etc. The library is part of the unified package of applied programs, developed at the Computing Centre of the Academy of Sciences of the USSR, for solving problems of unconstrained minimization of functions of several variables, and nonlinear programming and optimal control problems.

1. Basic computational formulae

Of the vast literature on numerical methods for solving optimal control problems, we shall only mention a few monographs [1, 2, 3] and two survey papers [4, 5]. All the methods described are based on reduction of the initial problem to a problem of nonlinear programming. This approach has been developed by various authors (see e.g. [2, 4]). It has proved unusually effective for several reasons: first, as a result of it, many previously hypothetical heuristic algorithms become obvious and can be extended; second, it has enabled the rich stock of methods of nonlinear programming and unconstrained minimization to be drawn on; and third, it has created a constructive basis for devising methods for optimizing systems, integrable by schemes with a high order of accuracy.

Let the controlled process be described by the system of ordinary differential equations

$$\frac{dx}{dt} = f(x,u,t), \qquad 0 \le t \le T, \tag{1.1}$$

where t is the independent variable, x is the phase vector, and u the control vector. We shall not be specific about the vector dimensionalities since they are not vitally important. Constraints are imposed on the controlled process (1.1) of the equation and inequality types:

$$\Gamma^1(x(t),u(t),t) = 0, \qquad \Gamma^2(x(t),u(t),t) \le 0. \tag{1.2}$$

[1] *Zh. vȳchisl. Mat. mat. Fiz.* Vol.19, No.2, pp. 367–387, 1979.

We are also given the terminal constraints

$$\Gamma^3(x(T),T)=0, \qquad \Gamma^4(x(T),T)\le 0. \tag{1.3}$$

The functional to be minimized is

$$I=\int_0^T F(x(t),u(t),t)dt+F_T(x(T),T). \tag{1.4}$$

The optimal control problem may be stated as follows.

Problem I. To choose the control $u(t)$ such that functional I is minimized in the fixed interval $[0,T]$, $u(t)$ being chosen from all the possible controls such that conditions (1.2) and (1.3) are satisfied for the corresponding solutions $u(t)$ and $x(t)$ of system (1.1).

Section 4 below is devoted to various generalizations of this problem. At present, we shall simplify the problem as much as possible in order to facilitate the treatment.

We divide the interval $[0,T]$ into $k-1$ subintervals. Let h_i be the length of the i-th subinterval; also, we put

$$t_1=0, \qquad t_i=\sum_{s=1}^{i-1}h_s, \qquad x_i=x(t_i), \qquad u_i=u(t_i), \qquad t_k=T,$$
$$z_i=(x_i,u_i,t_i), \qquad f(z_i)=f(x_i,u_i,t_i), \qquad i=1,2,\dots,k.$$

In each i-th subinterval we integrate (1.1) by a scheme belonging to the family of Runge–Kutta methods:

$$x_{i+1}=x_i+h_i\sum_{s=1}^{r}g_sf(z_{is}); \tag{1.5}$$

here, $z_{is}=(x_{is},u_{is},t_{is})$, $u_{i1}=u_i$, $u_{is}=u(t_{is})$,

$$t_{i1}=t_i, \qquad t_{i,j+1}=t_i+\beta_jh_i, \qquad x_{i1}=x_i, \qquad x_{i,j+1}=x_i+\beta_jh_if(z_{ij}), \tag{1.6}$$

where g_s, β_j is a collection of numbers, and all the β_j lie between 0 and 1. In (1.5) and (1.6), and throughout the present section unless stipulated otherwise, the subscripts i, s, j take integral values from the intervals $[1,k-1]$, $[1,r]$, $1,r-1]$, respectively.

In the simplest case, when $r=1$, (1.5) transforms to the Euler integration scheme

$$x_{i+1}=x_i+h_if(x_i,u_i,t_i). \tag{1.7}$$

On evaluating the quadrature in (1.4) according to expressions (1.5), (1.6), we get

$$I(\bar{x},w)=F_T(z_k)+\sum_{i=1}^{k-1}h_i\sum_{s=1}^{r}g_sF(z_{is}), \tag{1.8}$$

where $\bar{x}=(x_{11},\dots,x_{1r},x_{21},\dots,x_k)$, $w=(u_{11},\dots,u_{1r},u_{21},\dots,u_k)$.

On specifying the vector of controls w, we can uniquely determine the phase vector $\bar{x}$ with the aid of relations (1.5) and (1.6). In view of this connection, we shall write $\bar{x}=x(w)$.

We replace the constraints (1.2), (1.3) by their discrete analogues

$$\bar{\Gamma}^1(\bar{x},w)=0, \qquad \bar{\Gamma}^2(\bar{x},w)\le 0. \tag{1.9}$$

Here, we combine the constraints of the equation and inequality types along a trajectory, and the terminal constraints:

$$\begin{aligned}\bar{\Gamma}^1(\bar{x}, w) &= [\Gamma^1(z_{11}), \Gamma^1(z_{21}), \ldots, \Gamma^1(z_{i1}), \ldots, \Gamma^1(z_k), \Gamma^3(z_k)],\\ \bar{\Gamma}^2(\bar{x}, w) &= [\Gamma^2(z_{11}), \Gamma^2(z_{21}), \ldots, \Gamma^2(z_{i1}), \ldots, \Gamma^2(z_k), \Gamma^4(z_k)].\end{aligned}$$

We replace the initial problem I by the following problem of nonlinear programming.

Problem II. To find the control vector w such that $\bar{x} = x(w)$ and w satisfy the constraints (1.9), and such that functions $I(\bar{x}, w)$ is minimized.

We shall assume that problems I and II have a solution. The question of the closeness of their solutions when, instead of (1.5), an approximation according to Euler's scheme is used, has been examined in several papers (see, e.g., [4, 6]). It appears that these results extend to the more general case of schemes of type (1.5).

Assume that functions f, Γ^1, Γ^2, F, F_T are differentiable with respect to x and u. Later, expressions will be needed to evaluate the derivatives of a differentiable function of the type

$$B(\bar{x}, w) = \sum_{i=1}^{k-1} h_i \sum_{s=1}^{r} g_s R(z_{is}) + \alpha(z_k). \tag{1.10}$$

We define the vectors

$$p_i = \frac{dB}{dx_i}, \qquad p_{ij} = \frac{dB}{dx_{ij}}, \qquad p_k = \frac{dB}{dx_k}, \qquad p_{ir} = \frac{dB}{dx_{ir}} \tag{1.11}$$

of the same dimensionality as x. Recalling dependences (1.5) and (1.6), defining the vectors x_{i+1}, $x_{i,j+1}$ as differentiable functions of the "previous" vectors, we obtain the recurrence relations

$$p_i = \frac{\partial B}{\partial x_i} + \sum_{s=1}^{r} \frac{\partial x_{is}}{\partial x_i} \frac{dB}{dx_{is}} + \frac{\partial x_{i+1}}{\partial x_i} \frac{dB}{dx_{i+1}} = p_{i+1} + \sum_{s=1}^{r} p_{is}; \tag{1.12}$$

$$p_{ij} = \frac{\partial B}{\partial x_{ij}} + \frac{\partial x_{i,j+1}}{\partial x_{ij}} \frac{dB}{dx_{i,j+1}} + \frac{\partial x_{i+1}}{\partial x_{ij}} \frac{dB}{dx_{i+1}},$$

$$p_{ir} = \frac{\partial B}{\partial x_{ir}} + \frac{\partial x_{i+1}}{\partial x_{ir}} \frac{dB}{dx_{i+1}} = \frac{\partial B}{\partial x_{ir}} + h_i g_r f_x(z_{ir}) p_{i+1}. \tag{1.13}$$

We introduce the function

$$H(z_{is}, p_{i+1}, p_{i,s+1}) = h_i[g_s R(z_{is}) + (f(z_{is}), g_s p_{i+1} + \beta_s p_{i,s+1})]$$

Here, (a, b) is the scalar product of the vectors a and b. Noting that $\beta_r = 0$, $p_{i,r+1} = 0$, we can rewrite relations (1.13) in the compact form

$$p_{is} = H_x(z_{is}, p_{i+1}, p_{i,s+1}). \tag{1.14}$$

Here and throughout, the subscripts x and u denote the derivatives with respect to x and u, respectively. The following basic expression for evaluating the gradient of the function B is obtained in a similar way:

$$\frac{dB(x(w), w)}{du_{is}} = \frac{\partial B}{\partial u_{is}} + \frac{\partial x_{i,s+1}}{\partial u_{is}} \frac{dB}{dx_{i,s+1}} + \frac{\partial x_{i+1}}{\partial u_{is}} \frac{dB}{dx_{i+1}} = H_u(z_{is}, p_{i+1}, p_{i,s+1}). \tag{1.15}$$

The present method of derivation is much simpler than that used to obtain system (1.7) in [2], due to the introduction of the auxiliary vectors by means of relations (1.11).

On specifying the control vector w, we can use relations (1.5) and (1.6) to find successively the phase vector $\bar{x}$ and evaluate the function B. In the case of a continuous system, this corresponds to integration of (1.1) "from left to right". Next, with the aid of (1.12) and (1.14), we evaluate the sequence p_i, p_{is}. These calculations correspond, in optimal control, to integration of the equations for the pulses "from right to left". After this, we use expression (1.15) to find the derivatives.

The following circumstance must be emphasized. The use of integration schemes of type (1.5) with $r \geq 2$ presupposes that, in each interval of integration, the function $f(x, u, t)$ has bounded derivatives with respect to all its arguments. If the control $u(t)$ varies substantially (by an amount ~ 1) inside an interval of integration, the calculations lose some of their accuracy. Hence, in the optimal control search process, we need to monitor the position of the mesh base-points with respect to t, and if necessary, either change their disposition or else limit the control variation at each integration step. It is simplest to regard the control as constant within each interval. This device is justified when solving many practical problems, in which the system (1.1) has to be integrated with high accuracy, while the optimal control can be evaluated more crudely. This is bound up with the fact that the optimal control can usually only be realized approximately, and the step for specifying the control does not need to be particularly small.

If we put the control constant in an interval of integration, then $u_i = u_{i1} = \ldots = u_{ir}$, the vector w can be regarded as consisting of the set $[u_1, \ldots, u_k]$, and for the derivatives of B with respect to u_j we have the expression

$$\frac{dB(\bar{x}, w)}{du_i} = \sum_{s=1}^{r} H_u(z_{is}, p_{i+1}, p_{i,s+1}).$$

Great simplifications are possible in the resulting relations, due to the fact that the gradients dB/du are only needed to find the approximate optimal controls, and the accuracy of computing them needs to be matched with the accuracy of solving optimization problem II. As a result of this, small terms in the relations for the gradients and pulses can be disregarded.

Let us consider some particular cases of integration schemes (1.1).

1. In the case of Euler's method (1.7), we have

$$\begin{aligned} B &= \sum_{i=1}^{k-1} h_i R(z_i) + \alpha(z_k), & H(z_i, p_{i+1} &= h_i[R(z_i) + (f(z_i), p_{i+1})], \\ p_i &= p_{i+1} + H_x(z_i, p_{i+1}, & p_k &= \alpha_x(z_k), \\ \frac{dB}{du_i} &= H_u(z_i, p_{i+1}, & \frac{dB}{du_k} &= \alpha_u(z_k). \end{aligned} \tag{1.16}$$

To prove the convergence, it is important to be able to evaluate the second derivative of the function B. If we assume that the functions defining problem I are twice differentiable with respect to x and u, we can obtain the relations for the second derivatives of B. We introduce the square symmetric matrices of second derivatives

$$P_i = \frac{d^2 B(\bar{x}, w)}{dx_i dx_i}, \qquad G_{ij} = \frac{d^2 B(\bar{x}, w)}{du_i du_j}.$$

It is easy to obtain the following recurrence relations:

$$G_{ii} = H_{uu}(z_i, p_{i+1} + \Phi_u(z_i)P_{i+1}\Phi_u^\top(z_i), \qquad G_k = B_{u_k u_k},$$
$$P_i = H_{xx}(z_i, p_{i+1} + \Phi_x(z_i)P_{i+1}\Phi_x^\top(z_i), \qquad P_k = B_{x_k x_k},$$
$$G_{ij} = \Phi_u(z_j)D^i_{j+1}[H_{xu}(z_i, p_{i+1} + \Phi_x(z_i)P_{i+1}\Phi_u^\top(z_i)],$$
$$D^i_{j+1} = \frac{\partial x_i}{\partial x_{j+1}} = \Phi_x(z_{j+1})\Phi_x(z_{j+2})\ldots\Phi_x(z_{i-1}), \qquad \Phi(z_i) = x_i + h_i f(z_i);$$

here, $j < i$, D^i_i is the identity matrix, and the superscript "$\top$" denotes transposition.

These relations can be generalized for the case (1.5), but since they become unwieldy, we shall not quote the generalizations. Notice that the relations obtained in [7] for the second derivatives are incorrect, since they contain no terms with second derivatives of the function H; they give a correct result only in the case of linear systems.

2. For Euler's method with recalculation

$$\begin{aligned} r &= 2, \quad g_1 = 0, & g_2 &= 1, \quad \beta_1 = 1/2, \\ x_{i+1} &= x_i + h_i f(z_{i2}), & x_{i2} &= x_i + 0.5h_i f(z_{i1}), \\ p_{i2} &= h_i[R_x(z_{i2}) + f_x(z_{i2})p_{i+1}], & p_{i1} &= 0.5h_i f_x(z_{i1})p_{i2}, \\ \frac{dB}{du_{i1}} &= 0.5h_i f_u(z_{i1})p_{i2}, & \frac{dB}{du_{i2}} &= h_i[R_u(z_{i2}) + f_u(z_{i2})p_{i+1}], \end{aligned}$$
$$p_i = p_{i+1} + p_{i1} + p_{i2}.$$

By comparison with Euler's scheme, the number of points at which the control vector is sought is here doubled. We can put $u_i = u_{i1} = u_{i2}$ throughout, and note that, when Euler's scheme with recalculation is used, terms of order h_i^3 are discarded at each step of integration. We can determine the vectors p_i and derivatives to the same accuracy. As a result, we obtain the expressions

$$\begin{aligned} p_i &= p_{i+1} + h_i[f_x(z_{i2})\bar{p}_{i+1} + R_x(z_{i2})], \\ \bar{p}_{i+1} &= p_{i+1} + 0.5h_i[f_x(z_{i2})p_{i+1} + R_x(z_{i2})], \\ \frac{dB}{du_i} &= h_i[f_u(z_{i2})\bar{p}_{i+1} + R_u(z_{i2})]. \end{aligned}$$

Here, the derivatives with respect to x and u have to be taken into account with an error of order h_i^2.

3. The very popular scheme of the Runge–Kutta method is obtained on putting $r = 4$, $g_1 = g_4 = 1/6$, $g_2 = g_3 = 1/3$, $\beta_1 = \beta_2 = 1/2$, $\beta_3 = 1$, $\beta_4 = 0$. There is no difficulty in writing all the necessary expressions in this case. We shall, therefore, confine ourselves to the case when the control is constant in the interval of integration. The relations for recalculating the pulses will be quoted with an error $O(h_i^3)$, and the relations for the derivatives, with an error $O(h_i^2)$:

$$\begin{aligned} p_i &= p_{i+1} + h_i\sum_{s=1}^{4} g_s[R_x(z_{is}) + f_x(z_{is})p_{i+1}] + \\ &+ h_i^2\sum_{s=2}^{4} g_s\beta_{s-1}f_x(z_{i,s-1})[R_x(z_{is}) + f_x(z_{is})p_{i+1}], \\ \frac{dB}{du_i} &= h_i\sum_{s=1}^{4} g_s[R_u(z_{is}) + f_u(z_{is})p_{i+1}]. \end{aligned}$$

The relations for other schemes of integration can be obtained in a similar way.

2. Necessary conditions for an extremum

We introduce the dual vectors λ^1, λ^2, having the same dimensionalities as $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, respectively. We form the Lagrange function for problem II:

$$L(w,\lambda) = I(x(w),w) + (\lambda^1, \bar{\Gamma}^1(x(w),w)) + (\lambda^2, \bar{\Gamma}^2(x(w),w)). \tag{2.1}$$

Here, λ is the union of vectors λ^1 and λ^2.

Using well-known results from nonlinear programming, we can state the necessary and sufficient conditions for an extremum for problem II. Let w_* be the solution of problem II, let $x_* = x(w_*)$, and let the constraints (1.9) satisfy the regularity condition; then, by the Kuhn–Tucker theorem, it is necessary that vectors λ^1_* and λ^2_* exist, such that

$$\frac{dL(w_*,\lambda_*)}{dw} = 0, \qquad \lambda^{2\ell}_* \bar{\Gamma}^{2\ell}(x_*,w_*) = 0, \qquad \lambda^{2\ell}_* \geq 0, \tag{2.2}$$

where $\lambda^{2\ell}$, $\bar{\Gamma}^{2\ell}$ are, respectively, the ℓ-th coordinates of vectors λ^2 and $\bar{\Gamma}^2$. These conditions hold for all components of the vectors. The derivative of L with respect to w in (2.2) is found from relations (1.11) – (1.15) on taking expression (2.1) for B in them. In particular, (1.15) is rewritten as

$$\begin{aligned} \frac{dL(w_*,\lambda_*)}{du_{is}} &= \frac{\partial}{\partial u_{is}} H(z_{*is}, p_{*i+1}, p_{*i,s+1}, \lambda_{*i}^{1}, \lambda_{*i}^{2}), \\ H(z_{is}, p_{i+1}, p_{i,s+1}, \lambda_{*i}^{1}, \lambda_{*i}^{2}) &= h_i[g_s F(z_{is}) + (f(z_{is}), g_s p_{i+1} + \beta_s p_{i,s+1})] + \\ &+ \ (\lambda_{*i}^{1}, \bar{\Gamma}^1_i(z_{i1})) + (\lambda_{*i}^{2}, \bar{\Gamma}^2_i(z_{i1})). \end{aligned}$$

Here, $\bar{\Gamma}^1_i$, $\bar{\Gamma}^2_i$ are the constraint vectors, evaluated at the i-th step, and λ_{*i}^{1}, λ_{*i}^{2} are the corresponding vectors of Lagrange multipliers.

With certain assumptions, the solution of problem II can be replaced by a search for the max-min

$$\max_{\lambda^1} \max_{\lambda^2 \geq 0} \min_{w} L(w, \lambda^1, \lambda^2). \tag{2.3}$$

The arguments underlying the transformation from problem II to problem (2.3) are similar to those in nonlinear programming.

Denote by S the set of all indices ℓ for which $\bar{\Gamma}^{2\ell}(x(w_*), w_*) = 0$.

If vectors w_*, λ^1_*, λ^2_* exist, such that conditions (2.2) are satisfied, and for any non-zero vector w for which

$$\frac{d\bar{\Gamma}^1(x(w_*),w_*)}{dw} w = 0, \qquad \left(\frac{d\bar{\Gamma}^{2\ell}(x(w_*),w_*)}{dw},\ w = 0 \right), \qquad \ell \in S,$$

we have the inequality

$$w^\top \frac{d^2 L(w_*,\lambda_*)}{dw^2} w > 0,$$

then w_* is an isolated local solution of problem II.

This sufficient condition can be used to prove the convergence of the methods given below for solving problem II. The sufficient conditions for convergence will not be quoted, however, since they are no more than a repetition, in slightly changed terms, of the well-known conditions of nonlinear programming.

Notice that our approach, based on the techniques of nonlinear programming, does not lead to "discontinuities" in the pulses, as distinct from the results obtained for continuous systems in [8]. Our scheme can be extended to continuous systems by obtaining for the latter propositions similar to the maximum principle. By developing this approach, we arrive at the necessary conditions for an extremum obtained by a different method in [9], or with certain assumptions about convexity, at the sufficient conditions quoted in [10].

3. Description of the algorithms

At present, the library compiled by the authors contains 16 algorithms for solving optimal control problems. Many of the familiar algorithms are excluded. In particular, there are no algorithms that are not adapted for taking account of phase constraints, and no algorithms based on variations in the state space, etc.

In most of the methods outlined below, problem II is solved according to a general scheme: the vector w is specified, the values of I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$ are computed, and on the basis of these, a function B of type (1.10) is formed; then B is minimized with respect to w using any method from the library of unconstrained minimization (u.m.) algorithms.

The u.m. algorithms can be divided into 3 groups: (1) those not requiring the existence of derivatives of the minimized function, (2) those using the existence of the first derivative, and (3) those using the existence of first and second derivatives. By strengthening the conditions imposed on the minimized function, we can construct more efficient algorithms. Hence the highest rate of convergence can be obtained by means of algorithms of group (3); but to realize them, we are obliged to search for the matrices of second derivatives of the minimized function. In optimal control problems, when the dimensionality of the vector w reaches $200 \div 500$, such computations demand a large computer memory. When solving practical problems, therefore, methods of group (2) were basically used (the method of conjugate gradients, or steepest descent, etc.).

Problem II is a particular case of a nonlinear programming problem. Its most important feature is the comparative simplicity of evaluating the derivatives of function B with respect to the components of vector w. This affords great scope for effectively utilizing various methods of nonlinear programming, based on the use of first derivatives.

The various methods of solving problem II differ from each other in the ways of constructing the functions B, and the rules for varying the functions during the iterations. The choice of scheme for integrating system (1.1) only influences the rules for computing I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, and the derivatives of B with respect to w, and is unimportant from the point of view of describing the algorithms. This property was made use of when compiling the library. The procedures for computing I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, and seeking the derivatives of B, were formed as separate modules. When replacing the integration schemes in the library algorithms, only these modules were changed, while the other units remained unchanged. To simplify the treatment as much as possible, we shall assume that the integration is performed by Euler's scheme.

Let us briefly describe the main algorithms.

1. *The method of penalty functions [11].* We form the penalty function

$$B(w,\tau) = I(x(w),w) + \tau[\varphi(\bar{\Gamma}^1(x(w),w)) + \psi(\bar{\Gamma}^2(x(w),w))].$$

Here, $0 < \tau$ is the penalty coefficient; in the case of a scalar argument, the functions $\varphi(y)$ and $\psi(y)$ are such that $\varphi(0) = \psi(0) = 0$; $\varphi(y) > 0$ for $y \neq 0$; if $y < 0$, then $\psi(y) = 0$; if $y > 0$, then $\psi(y) > 0$. If y is an ℓ-dimensional vector with components $y^1, \ldots, y^\ell$, we put

$$\varphi(y) = \sum_{i=1}^{\ell} \varphi(y^i), \qquad \psi(y) = \sum_{i=1}^{\ell} \psi(y^i).$$

The method is as follows: for some monotonically increasing sequence $\tau_1 < \tau_2 < \ldots$ we construct a sequence of vectors $w(\tau_1), w(\tau_2), \ldots$, found by solving the problem of unconstrained minimization of B with respect to w:

$$B(w(\tau_s),\tau_s) = \min_w B(w,\tau_s), \qquad s = 1,2,\ldots. \tag{3.1}$$

Under certain conditions, the sequence $w(\tau_s)$ converges as $s \to \infty$ to the solution of problem II.

In order to use methods of differentiable minimization, it is advisable to take as φ and ψ sufficiently smooth functions. Taking as our guide the use of the method of conjugate gradients, we used the following twice continuously differentiable functions:

$$\varphi(y) = y^2, \qquad \psi(y) = \begin{cases} 0, & y \le 0, \\ k_1 y^3, & 0 < y \le R_1, \\ y^2 + k_2 y + k_3, & y > R_1, \end{cases}$$

$k_1 = 1/(3R_1)$, $k_2 = -R_1$, $k_3 = R_1^2/3$, where $0 < R_1$ is a fairly small number (usually, $R_1 = 10^{-4}$).

On next using expression (1.15) to compute the derivatives of the function B, we solve the approximate problem (3.1). The minimization process stops as soon as a vector w is found such that

$$\|B_w(w, \tau)\| \le \varepsilon(\tau), \tag{3.2}$$

where $\varepsilon(\tau) \to 0$ as $\tau \to \infty$. In the numerical realization of the method, the sequences $\{\tau_s\}$, $\{\varepsilon(\tau_s)\}$ were constructed as follows:

$$\begin{array}{lll} \tau_{s+1} = \nu\tau_s, & \varepsilon(\tau_{s+1}) = b/(1 + \ln \tau_{s+1}), & \text{if } \tau_s[\varphi(\bar\Gamma^1) + \psi(\bar\Gamma^2)] > \tilde e, \\ \tau_{s+1} = \tau_s, & \varepsilon(\tau_{s+1}) = \mu\varepsilon(\tau_s), & \text{if } \tau_s[\varphi(\bar\Gamma^1) + \psi(\bar\Gamma^2)] \le \tilde e. \end{array}$$

Here, the numbers τ, ν, μ, $\tilde e$, b are such that $0 < \tau_1$, $1 < \nu$, $0 < b$, $0 < \mu < 1$, $\tilde e > 0$.

A modification of the method was devised, in which we took as φ and ψ non-differentiable functions, with the result that there was no need for τ to tend to infinity. However, to solve problem (3.1) we then had to use the less effective methods of non-differentiable optimization; this considerably slows the computations. The move to non-differentiable penalty functions is, therefore, only justified when, among the functions defining problem I, some are non-differentiable. The method of penalties is most useful when the initial approximation is roughly known. The computations usually start from this, and the initial values of the dual variables are found. The drawback of the method is the low rate of convergence close to the solution.

2. *The dual method [12].* As B we take the function

$$B(w, \lambda^1, \lambda^2) = L(w, \lambda^1, \lambda^2) + \tau[\varphi(\bar\Gamma^1(x(w), w)) + \psi(\bar\Gamma^2(x(w), w))],$$

where L is defined in Section 2, and τ is a fixed positive parameter. Assume that λ_s^1, λ_s^2 are known; then the s-th step is performed according to the relations

$$\begin{aligned} &B(w_s, \lambda_s^1, \lambda_s^2) = \min_w B(w, \lambda_s^1, \lambda_s^2), \\ &\lambda_{s+1}^1 = \lambda_s^1 + c\bar\Gamma^1(x(w_s), w_s), \\ &\lambda_{s+1}^{2\ell} = \lambda_s^{2\ell} + 4c\bar\Gamma_s^{2\ell} \times \begin{cases} e_1, & \text{if } \lambda_s^{2\ell} < e_1,\ \bar\Gamma_s^{2\ell} > 0, \\ \lambda_s^{2\ell} & \text{otherwise.} \end{cases} \end{aligned} \tag{3.3}$$

Here, $\bar\Gamma_s^{2\ell} = \bar\Gamma^{2\ell}(x(w_s), w_s)$, is the descent step, and $0 < e_1$ is a fairly small number (usually, $e_1 = 10^{-4}$). The u.m. process (3.3) stops when a condition similar to (3.2) is satisfied.

3. *The method of modified Lagrange functions.* A lot has been published about this method (see, e.g., [13]). We form the generalized Lagrange function as

$$B(w, \lambda^1, \lambda^2) = I(x(w), w) + [\|\lambda^1 + \tau\bar\Gamma^1\|^2 + \|(\lambda^2 + \tau\bar\Gamma^2)_+\|^2]/2\tau,$$

where $\|\cdot\|$ is the Euclidean vector norm; the components of the vector a_+^i are expressed in terms of the components of the vector a^i by $a_+^i = a^i$, if $a^i > 0$; otherwise, $a_+^i = 0$.

The passage to the $(s+1)$-th step is performed according to the relations

$$\begin{aligned} &B(w_s, \lambda_s^1, \lambda_s^2) = \min_w B(w, \lambda_s^1, \lambda_s^2), \\ &\lambda_{s+1}^1 = \lambda_s^1 + \tau\bar{\Gamma}^1(x(w_s), w_s), \qquad \lambda_{s+1}^2 = (\lambda_s^2 + \tau\bar{\Gamma}^2(x(w_s), w_s))_+. \end{aligned} \tag{3.4}$$

The parameter τ has to be fairly large, in order to ensure that problem (3.4) is solvable and the method is convergent.

4. *The method of simple iteration [14].* The generalized Lagrange function is constructed in the form

$$B(w, \lambda^1, \lambda^2) = I(x(w), w) + [\|\lambda^1 + \tau\bar{\Gamma}^1\|^2 + \psi(\tau\bar{\Gamma}^2) + 2(\lambda^2, \gamma(\tau\bar{\Gamma}^2))]/2\tau;$$

here, $\psi(y)$ is a fairly smooth penalty function, e.g. $\psi(y) = y_+^i$. The vector function $\gamma(a)$ is such that its ℓ-th component γ^ℓ is expressed in terms of the ℓ-th component a^ℓ of the vector a according to the rule: if $a^\ell \leq 0$, then $\gamma^\ell = (1 - a^\ell)^{-q}$; otherwise,

$$\gamma^\ell = 1 + qa^\ell + \frac{q(q+1)}{2}(a^\ell)^2 + \frac{q(q+1)(q+2)}{3!}(a^\ell)^3;$$

here, q is any positive number (usually, $q = 1$).

The method is as follows:

$$B(w_s, \lambda_s^1, \lambda_s^2) = \min_w B(w, \lambda_s^1, \lambda_s^2), \quad \lambda_{s+1}^1 = \lambda_s^1 + \tau\bar{\Gamma}^1(x(w_s), w_s), \quad \lambda_{s+1}^{2\ell} = \frac{\partial B(w_s, \lambda_s^1, \lambda_s^2)}{\partial \bar{\Gamma}^{2\ell}}.$$

This method provided a very high rate of convergence for a number of test problems. Its drawback, shared with the two preceding methods, is that a knowledge of the dual variables is required. To determine the latter, one of methods $1, 5$, or 6 was employed.

5. *Morrison's method [15].* We form the function

$$B(w, \eta) = (I(x(w), w) - \eta)^2 + \varphi(\bar{\Gamma}^1) + \psi(\bar{\Gamma}^2),$$

where φ and ψ are constructed in the same way as in the penalty method, and η is a parameter. We assume that, at the start of the computations, a lower bound is known for the solution of problem II, i.e. if $I_* = I(x(w_*), w_*)$, then we can indicate $\eta_0 < I_*$. At the $(s+1)$-th step we find η_{s+1} from the relations

$$B(w_s, \eta_s) = \min_w B(w, \eta_s), \qquad \eta_{s+1} = \eta_s + [B(w_s, \eta_s)]^{1/2}. \tag{3.5}$$

6. *The modified Morrison method [14].* This differs from Morrison's method in the way that the parameter η is varied

$$\eta_{s+1} = \eta_s + \frac{B(w_s, \eta_s)}{(I(x(w_s), w_s) - \eta_s)}. \tag{3.6}$$

The method is justified only in the case of convex programming problems. It is easily shown that, when solving a u.m. problem exactly, the condition $\eta_0 \leq I_*$ implies $\eta_s \leq I_*$ for all s. The value of η_{s+1} given by (3.5) is less than or equal to the η_{s+1} obtained from (3.6); this ensures a higher convergence rate in the latter case. A serious drawback of the two methods is the requirement that the unconstrained minimization be performed with high accuracy. It is only then that the necessary condition for operation of the methods, namely, $\eta_s \leq I_*$, is satisfied during the computations. This requirement is particularly important in the second method, since the increase of η is faster.

Both methods closely resemble the method of penalties after solving the u.m. problem at the s-th step, and finding a point w_s, a value of τ can be found for which the solution of the u.m. problem for the penalty method gives precisely the same result. The advantage of these methods, therefore, lies in the fact that the variation of η takes place automatically, whereas in the penalty method a sequence of τ_s has to be specially assigned.

7. *The method of linearization.* The method described in [16] for solving problems of non-linear programming, after being justified in [17], was used in [18] for optimal control problems.

At the s-th step, after linearizing the function to be minimized and the constraints, we obtain

$$\begin{aligned} I(w_s + \delta w) &= I(w_s) + (I_w(w_s), \delta w) + O(\|\delta w\|^2), \\ \bar{\Gamma}^1(w_s + \delta w) &= \bar{\Gamma}^1(w_s) + \bar{\Gamma}^1_w(w_s)\delta w + O(\|\delta w\|^2), \\ \bar{\Gamma}^2(w_s + \delta w) &= \bar{\Gamma}^2(w_s) + \bar{\Gamma}^2_w(w_s)\delta w + O(\|\delta w\|^2). \end{aligned}$$

Here, for typographical simplicity, we have omitted the dependence of the functions on x.

We pose the following problem of quadratic programming: to find the minimum with respect to δw of the function

$$(I_w(w_s), \delta w) + a\|\delta w\|^2 \tag{3.7}$$

under the constraints

$$\bar{\Gamma}^1(w_s) + \bar{\Gamma}^1_w(w_s)\delta w = 0, \qquad \bar{\Gamma}^2(w_s) + \bar{\Gamma}^2_w(w_s)\delta w \le 0, \tag{3.8}$$

where a is a positive coefficient. After finding the optimal value δw_* we put

$$w_{s+1} = w_s + c\delta w_*.$$

There are several ways of choosing the step c. Following [17], we find c from the condition that the penalty function decreases at each step:

$$W = I(w_{s+1}) + N \max[0, |\bar{\Gamma}^1(w_{s+1})|, \bar{\Gamma}^2(w_{s+1})];$$

here, N is a fairly large number; $\bar{\Gamma}^2$ and $|\bar{\Gamma}^1|$ denote, respectively, the set of components of $\bar{\Gamma}^2$ and the set of moduli of components of $\bar{\Gamma}^1$. To solve (3.7), (3.8), a transformation to the dual problem is performed. If we put $a = 0$, then (3.7), (3.8) becomes a problem of linear programming; to (3.8) we have to add the requirement that the moduli of the components of the vector δw be bounded.

Of all the methods contained in the library, the present one is the most laborious and demands the largest computer memory for the computations. This is due to the need to store the gradients of the "active" constraints. It is, therefore, best to use this method only when the number of essential phase constraints is reasonably small.

8. *The method of Krylov–Chernous'ko iterations [19].* This was originally proposed to solve problems in which there are no phase and terminal constraints. We will show that the method can be adapted to solve problem I (in this case it has to be regarded as an algorithm for solving u.m. problems). Put

$$\begin{aligned} \bar{H}(x, w, p) &= \sum_{i=1}^{k-1} H_i(z_i, p_{i+1}), \qquad p_k = \frac{\partial B}{\partial x_k}, \\ H_i(z_i, p_{i+1}) &= h_i[R(z_i) + (f(z_i), p_{i+1})]. \end{aligned}$$

After specifying a control w_s, we find from system (1.7) $\bar{x}_s = x(w_s)$ and from (1.16), the vector $\bar{p}_s = p(w_s)$. In accordance with Section 1, every vector $\bar{w}$ minimizing the function $B(x(w))$ is a stationary point of the function $\bar{H}$, i.e.

$$\frac{\partial H(x(\bar{w}), \bar{w}, p(\bar{w}))}{\partial w} = 0. \tag{3.9}$$

We replace the problem of minimizing $B(x(w), w)$ by the problem of finding the stationary points of the function $\bar{H}$ (i.e. the points $\bar{w}$ that satisfy (3.9)). If

$$\frac{\partial H(x(w_s), w_s, p(w_s))}{\partial w} = a,$$

where $\|a\| \neq 0$, then, on solving Eq. (3.9) with respect to w_s, that appears explicitly in it, we obtain

$$w_s = \rho(x(w_s), p(w_s), a).$$

For solving (3.9) we use the method of simple iteration, putting

$$w_{s+1} = \rho(x(w_s), p(w_s), 0).$$

If $\|w_{s+1} - w_s\| < e$, where e is the accuracy of the computations, we assume that the stationary point of the function $\bar{H}$ has been found. Assume that it coincides with the minimum point of the function B. We then vary function B in accordance with any of the above algorithms, and again replace the search for the minimum of B by the search for the stationary point of $\bar{H}$, etc.

This scheme has an attractive computational simplicity: instead of minimizing B with respect to the vector w of high dimensionality, we solve a large number of problems of finding stationary points of a function of a vector of small dimensionality. Unfortunately, this fragmentation does not apply in the very commonly encountered case of optimization with respect to the vector w and a supplementary vector of control parameters (we shall discuss this in Section 4). Moreover, the method is often divergent and can be used only when w_s is fairly well known. Initially, therefore, minimization of B with respect to w is performed by some u.m. method, and only then do we go over to the described procedure. In order to somewhat weaken the divergence of the method, we took $w_{s+1} = w_s$ and carried out the variation of w_{s+1} "from right to left" till the norm $\|w_{s+1} - w_s\|$ was sufficiently small. After this, the variation of w was stopped, new values of x and p were computed, and w_{s+1} was again corrected "from right to left".

9. *Reduction to a boundary value problem [1, 5].* Assume that w_s, $x(w_s)$, $p(w_s)$ are known at the s-th iterative step. Denote by ξ the vector identical with vector $p_1(w_s)$. We integrate (1.7), (1.16) "from left to right". The sequence of p_i is here found from the implicit scheme. The control at each step of the integration is found from condition (3.9). At the last k-th integration step, we obtain the vectors $\bar{x}_{ks}$, $\bar{p}_{ks}$. If

$$\bar{p}_{ks} = \frac{\partial B(\bar{x}_{ks}, u_{ks})}{\partial x_k}, \tag{3.10}$$

the auxiliary problem of finding the stationary point of the function $\bar{H}$ with respect to w is solved. Assume that these points coincide with the minimum point of B with respect to w. Using any of algorithms 1–7, we vary B and pass to the new $(s+1)$-th step. Otherwise, if (3.10) does not hold, we seek the vector ξ in such a way that this condition is satisfied. Usually, Newton's method is employed to find ξ. When realizing the method, various numerical difficulties connected with the divergence, are encountered (for more details see [5]).

It was shown in [20] that, with certain assumptions, a necessary condition for an extremum, similar to Pontryagin's maximum principle, holds in problem II:

$$H_i(x_i^*, u_i^*, p_{i+1}^*, \lambda_*^1, \lambda_*^2) = \min_{u_i} H_i(x_i^*, u_i, p_{i+1}^*, \lambda_*^1, \lambda_*^2); \tag{3.11}$$

here, the asterisk means that the relevant vector is evaluated for the optimal solution of problem II. The function H_i is defined in Section 2.

Condition (3.11) enables us to replace the search for stationary points of function H_i by the need to find the minimum of H_i with respect to u_i. Hence the search for the minimum of function B splits into the solution of a sequence of problems of minimizing H_i with respect to a vector u_i of low dimensionality. However, this device only proves successful close to the solution, and is, therefore, only used towards the end of the computations.

To conclude this section, notice that many other methods of nonlinear programming can be extended in a similar way to solve problem I. We only quote here the methods which form part of our library and have been tested by us for solving a number of practical problems.

4. Some generalizations

The algorithms in the library are designed to solve problems of a more general type than are described in Section 1. Let us indicate the main generalizations.

1. Instead of (1.1) we consider systems that contain the vector ξ of control parameters:

$$\dot{x} = f(x, u, t, \xi) = f(z, \xi). \tag{4.1}$$

The functions F, F_T, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$ may also depend on ξ. In problem I we have to choose the control $u(t)$ and vector ξ in such a way that (1.2) and (1.3) are satisfied, and I is minimized. The gradient of the function B with respect to ξ is evaluated as follows:

$$\frac{dB}{d\xi} = \frac{\partial \alpha(z_k, \xi)}{\partial \xi} + \sum_{i=1}^{k-1} \sum_{s=1}^{r} H_\xi(z_{is}, \xi, p_{i+1}, p_{i,s+1}). \tag{4.2}$$

We combine the control vector and vector of parameters in a single symbol, putting $\bar{w} = (w, \xi)$. Then, all the expressions and methods described above can be used, after replacing w in them by the vector $\bar{w}$. The optimization process with respect to w and ξ takes place simultaneously.

2. For system (1.1), part of the components of vector x may not be defined at $t = 0$. We refer them to the components of vector ξ. To evaluate the derivatives, we use expression (3.2), after adding to it the term $(\partial x_1/\partial \xi)p_1$. The components of the vector x, specified at $t = T$, have to be referred to the terminal constraints (1.3).

Computing experience shows that problems are often unusually sensitive to the accuracy of solving the boundary value problem for system (1.1). It is, therefore, advisable to start the computations with optimization with respect to ξ for fixed $u(t)$, and then go over to simultaneous optimization with respect to ξ and $u(t)$, and finally perform supplementary optimization with respect to ξ for fixed $u(t)$. The library of optimization methods was compiled in such a way that these computations are carried out according to the same algorithms without any transformations.

3. The time optimal problem can easily be reduced to the class considered. Assume that we seek for system (1.1) the control such that $u(t)$ and the corresponding trajectory everywhere satisfactory (1.2) and the terminal set

$$\Gamma^3(x(t), u(t), t) = 0, \qquad \Gamma^4(x(t), u(t), t) \leq 0 \tag{4.3}$$

is reached for the least possible value of t. As ξ we take a $(k-1)$-dimensional vector in which every i-th component is the same as the size h_i of the integration step in the i-th interval. Then, the minimized function is equal to the sum

$$I = \sum_{i=1}^{k-1} \xi^i.$$

We fix the number of discretization intervals k and seek ξ, w such that conditions (1.9) and (4.3) are satisfied, and I is minimized. We have to evaluate the derivative of B with respect to ξ from the expression

$$\frac{dB}{d\xi^i} = \sum_{s=1}^{r} \frac{\partial H(z_{is}, \xi^i, p_{i+1}, p_{i,s+1})}{\partial h_i}.$$

The problem now has the standard form and any of the methods described above can be used.

If the integration steps are constant everywhere, $h = h_1 = \ldots = h_{k-1}$, it is sufficient to introduce a scalar ξ, put $I = \xi(k-1)$, and evaluate the derivative of B from the expression

$$\frac{dB}{d\xi} = \sum_{i=1}^{k-1} \frac{dB}{d\xi^i}.$$

4. If the vector function $f(x, u, t)$ is differentiable with respect to x and u everywhere except for a finite number of points $\{t_j\}$, where there is a discontinuity of the first kind, then the step sizes h_j need to be chosen in such a way that all the points $\{t_j\}$ are at base-points of the main mesh for t. The expressions given in Section 1 remain unchanged. Similarly, if, at some points $\{t_i\}$ of the main t mesh, the phase trajectory has a given discontinuity, independent of x and u, then the expressions of Section 1 for evaluating the derivatives are again unchanged. If the discontinuity depends on x and u, i.e.

$$x(t_{i+}) = \psi(x(t_{i-}), u(t_i)),$$

where

$$x(t_{i+}) = \lim_{t \to t_{i+0}} x(t), \qquad x(t_{i-}) = \lim_{t \to t_{i-0}} x(t),$$

then we have to recalculate the pulses at these points. When finding p_{i-1}, instead of p_i we have to take the vector $(\partial\psi(x(t_{i-}), u(t_i))/\partial x)p_i$ and in the expression for the derivative dB/du_i we have to add the term $(\partial\psi(x(t_{i-}), u(t_i))/\partial u)p_i$.

5. If system (1.1) is linear, we arrive at an extremely simple particular case of problem I. All the algorithms are still applicable; they are merely simplified, since the derivatives of the right-hand sides of (1.1) are constant.

6. In many problems there are constraints of the "parallelepiped" type:

$$a \leq w \leq b. \tag{4.4}$$

It is not advisable to refer such constraints to constraints (1.9). It is simpler to allow for them when solving the unconstrained minimization problems. For, it can be shown that all the above expressions remain in force if, instead of the usual unconstrained minimization of B with respect to w, we solve the problem of minimizing B with respect to w on the set (4.4). Hence the library of u.m. programs was compiled in two versions: with and without an allowance for the constraints (4.4). In the linearization method, the vector δw is sought under constraints (3.7) and the condition $a \leq w_s + \delta w \leq b$.

7. In some problems an additional constraint is imposed on the rate of change of the control:

$$|du(t)/dt| \leq c. \tag{4.5}$$

If t_i and t_{i+1} are adjacent mesh points for the integration with respect to t, then, in the discrete version, constraint (4.5) is rewritten in the form

$$-ch_i \leq u(t_{i+1}) - u(t_i) \leq ch_i.$$

These two constraints can be referred to constraints Γ^2. When using the expressions of Section 1 for the derivatives, we have to allow for extra terms due to the fact that the constraints at the i-th step depend on u_{i+1} as well as on u_i.

Notice that constraints (4.5) can be introduced artificially for regularizing the problem. Delay conditions in system (1.1) can be taken into account in a similar way.

8. When seeking structures of optimal strength, it is required that the control $u(t)$ be constant in certain given intervals $A_j = [t_j, t_j + \tau_j]$. To allow for this factor, we have to specify the initial control satisfying this requirement, and then in later computations, take as dB/du_{js}, where $t_{js} \in A_j$, the sum of the derivatives dB/du_{js}, where the summation is over all indices such that $t_{js} \in A_j$.

9. According to [5], it becomes much more difficult to use modern methods for solving problem I if the functional has the form

$$I_1 = \max_{0 \leq t \leq T} v(x(t), u(t), t), \qquad I_2 = \int_0^T |v(x(t), u(t), t)| dt.$$

In spite of the fact that v is a differentiable function of its arguments, the functionals I_1 and I_2 are only directionally differentiable. In the first case, we introduce an auxiliary control parameter ξ, with respect to which the minimization is performed, and a new constraint, putting

$$I_1 = \xi, \qquad v(x(t), u(t), t) \leq \xi.$$

In the second case, we use the usual device: we introduce a new control $\tilde{u}(t)$ and two constraints, putting

$$I_2 = \int_0^T \tilde{u}(t) dt, \qquad -v(x(t), u(t), t) \leq \tilde{u}(t) \geq v(x(t), u(t), t).$$

The minimization is performed with respect to $u(t)$ and $\tilde{u}(t)$. In both cases the problems are reduced to type I. No non-differentiable functions appear, and all the methods can be applied without any supplementary modifications. Elementary min-max problems can be reduced in a similar way [21].

5. Example of numerical computations

Of the many varied practical problems that have been solved with the aid of the library, we shall just quote one, concerning optimization of the motion are given by the system

$$\begin{gathered} \dot{x} = V\cos\theta\cos\psi, \qquad \dot{y} = V\sin\theta, \qquad \dot{z} = -V\cos\theta\sin\psi, \\ G\dot{V} = g(k_1 P\cos\alpha - C_x qS - G\sin\theta), \qquad \dot{G} = -C_s, \\ V\dot{\theta} = g(k_2 N\cos\gamma - \cos\theta), \qquad \dot{\psi}V\cos\theta = -gk_2 N\sin\gamma. \end{gathered} \tag{5.1}$$

Here, x, y, z are the Cartesian coordinates of the f.v., V is the modulus of the velocity vector, θ is the angle of inclination of the trajectory, ψ is the heading angle, G is the weight, α is the angle of attack, γ is the rolling angle, k_1 is the motor thrust, referred to maximum thrust P, k_2 is the acceleration g, referred to maximum value N, k_3 is the relative braking force, S is the characteristic area of the f.v., C_x is the diag coefficient, $g = 9.81\,\mathrm{m/sec}^2$, q is the form drag, and C_s is the fuel consumption per sec.

We utilized the following dependences:

$$\begin{aligned}
&q = \rho(y)V^2/2, \quad \rho(y) = 3.3\cdot 10^{-10}y^2 - 1.155\cdot 10^{-5}y + 0.125,\\
&P = [10 + V^2/a^2(y)](2500 - y)/12.5, \qquad a(y) = 340.3 - 4.08\cdot 10^{-3}y,\\
&\alpha = \frac{k_2 NG}{k_1 P + 4.6qS}, \quad C_s = [0.7 + 2(k_1 - 0.3)^2]k_1 P/3600, \quad S = 55\,\mathrm{m}^2,\\
&N = \min\left(\frac{qS}{G}, \frac{150000}{G}, 8\right), C_x = 0.02 + 3.174\alpha^2 + 0.03k_3.
\end{aligned}$$

The following constraints were imposed on the controls:

$$\begin{aligned}
&0.05 \le k_1 \le 1, \qquad 0.01 \le k_2 \le 1, \qquad 0 \le k_3 \le 1,\\
&|\dot{k}_1| \le 0.2, \qquad |\dot{k}_2| \le 0.25, \qquad |\dot{k}_3| \le 1, \qquad |\dot{\gamma}| \le 1.57\,\mathrm{rad/sec}.
\end{aligned} \tag{5.2}$$

The initial values of the phase coordinates and controls were

$$\begin{aligned}
&x = z = 0, \qquad y = 5000\,\mathrm{m}, \qquad V = 300\,\mathrm{m/sec}, \qquad \psi = \theta = 0,\\
&G = 20000\,\mathrm{kG}, \qquad k_1 = 1, \qquad k_2 = 1/N, \qquad k_3 = 0, \qquad \gamma = 0.
\end{aligned} \tag{5.3}$$

For system (5.1) the controls $k_1(t)$, $k_2(t)$, $k_3(t)$, $\gamma(t)$ were sought, such that they satisfy conditions (5.2) and move the f.v. in the shortest time from (5.3) to the domain specified by the conditions

$$y = 7000\,\mathrm{m}, \qquad \theta = 0, \qquad \psi = -\pi, \qquad k_2 = 1/N, \qquad \gamma = 0. \tag{5.4}$$

As the initial control we took

$$k_1(t) \equiv 1, \qquad k_2(t) \equiv 0.3, \qquad k_3(t) \equiv 0, \qquad \gamma(t) = 1.5\sin(\pi t/T_1), \qquad T_1 = 21\,\mathrm{sec}.$$

The problem was solved by five methods. In order to trace the convergence process more clearly, the same conjugate gradients method was always used to solve the u.m. problem; the computations ended either when (3.2) is satisfied, or when the number of steps exceeded the amount D. We denote by T the running value of the functional during the computations (the time for moving from (5.3) to (5.4)). In Fig. 1 we show the variation of T as a function of the computing time t_1 on the BESM-6 computer. The points indicate the values of T at the instant of ending each s-th iteration; interpolation was performed in the intervening intervals. The numbers of the curves correspond to the numbers of the algorithms quoted in Section 4. Methods 3 and 4 demanded a knowledge of the dual variables. Hence they were only used after making one step by the penalty method.

Let us emphasize that the computing process is strongly influenced by a successful choice of the parameters of the methods. The authors did not perform any special optimization with respect to the parameters, nor did they attempt to obtain "record" results. Usually, 2-3 runs were made with each method, and the best result is quoted in Fig. 1. The following parameters were taken:

For method 1: $R_1 = 10^{-4}$, $\tau_1 = b = 1$, $\nu = 1.5$, $\mu = 0.9$, $D = 100$, $\tilde{e} = 10^{-4}$.

For method 3: $\tau = 10$, $\varepsilon(s) = 0.5/(1 + 0.5s)$, $D = 100$.

For method 4: $\tau = 10$, $\varepsilon(s) = 0.5/(1 + 0.05s)$, $D = 100$.

For method 5: $\eta_0 = 10$, $\varepsilon(s) = 0.1/(1 + s)$, $D = 100$.

For method 6: $\eta_0 = 10$, $\varepsilon(s) = 0.01/(1 + s)$, $D = 200$.

In every case a uniform mesh with respect to the independent variable with $k = 51$ was used, the integration was performed according to Euler's scheme, and the initial problem was reduced to a problem of nonlinear programming, in which the minimum was sought with respect to 205 variables in the presence of 702 constraints of the inequality type, and 5 constraints of the equation type. Constraints of the "parallelepiped" type were allowed for in the unconstrained minimization method, while other constraints were referred to constraints of type (1.9). The number of them was 405. The vector of dual variables λ had the same dimensionality. The computations by each method were stopped if the difference between two consecutive values of T proved to be less than 10^{-4}. It can be seen from Fig. 1 that methods 4 and 3 have the fastest rate of convergence. An acceptable accuracy was reached after 25 mins. The penalty method required over an hour to achieve closely similar results. When using method 6 the accuracy of solving the u.m. problem had to be increased as compared with the case of using method 5, since otherwise a value of T exceeding the optimal value was obtained after the second step. In spite of serious efforts, the authors were not able to obtain results by the linearization method that could compete with those obtained with other methods; they are, therefore, not included in Fig. 1.

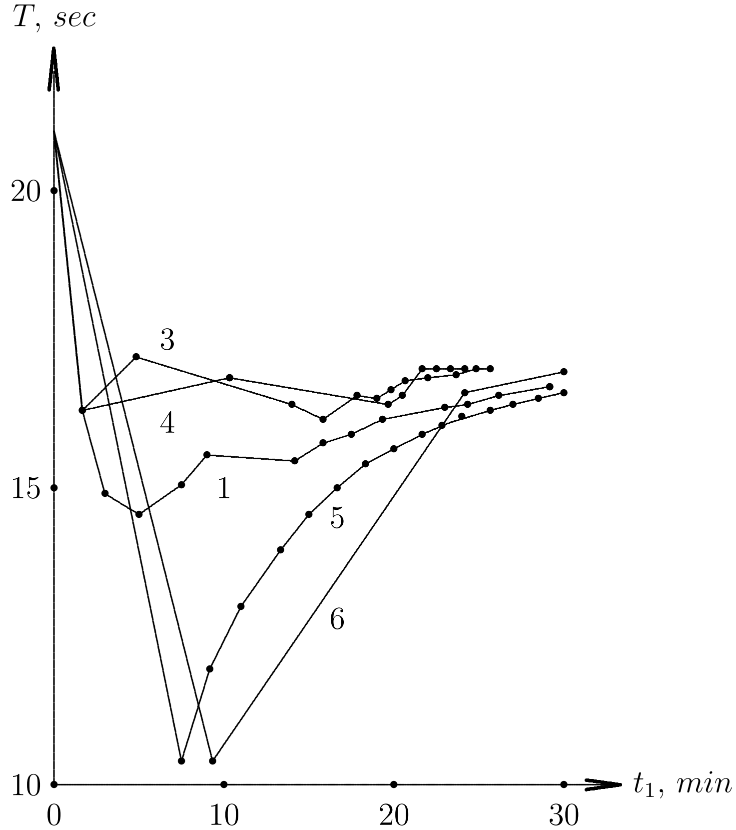

Fig.1

Our library of programs has been used to solve many other problems. In particular, in [22] we describe the solution of problems of optimal design of power structures, and solve Isaac's game dolichobrachistochrone problem.

REFERENCES

1. MOISEEV, N.N., *Numerical methods in the theory of optimal systems* (Chislennye metody v teorii optimal'nykh sistem), Nauka, Moscow, 1971.

2. POLAK, E., *Numerical optimization methods, A unified approach* (Chislennye metody optimizatsii. Edinyi podkhod), Mir, Moscow, 1974.

3. VASIL'EV, F.P., *Lectures on methods of solving extremal problems* (Lektsii po metodam resheniya ekstremal'nykh zadach), Izd-vo MGU, Moscow, 1974.

4. ERMOL'EV, Yu.M., and GULENKO, V.P., The finite difference method in optimal control problems, *Kibernetika*, No.3, 1–20, 1967.

5. FEDORENKO, R.P., Approximate solution of modern variational problems, in: *Mathematical programming and allied topics. Computational methods* (Matem. programmirovanie i smezhnye voprosy. Vȳchisl. metody), TsEMI Akad. Nauk SSSR, Moscow, 28–97, 1976.

6. BUDAK, B.M., BERKOVICH, E.M., and SOLOV'EVA, E.N., On the convergence of difference approximations for optimal control problems, *Zh. vȳchisl. Mat. mat. Fiz.*, **9**, No.3, 522–547, 1969.

7. OUTRATA, I.V., Second-order methods in discrete optimal control problems, *Kybernetika*, **12**, No.1, 38–45, 1976.

8. DUBOVITSKII, A.Ya., and MILYUTIN, A.A., Constrained extremum problems, *Zh. vȳchisl. Mat. mat. Fiz.*, **5**, No.3, 395–453, 1965.

9. SMOL'YAKOV, E.R., The maximum principle for problems with phase constraints, in: *Operations research* (Issl. operatsii), No.2, 136–155, VTs Akad. Nauk SSSR, Moscow, 1971.

10. MANGASARIAN, O.L., Sufficient conditions for the optimal control of non-linear systems, *SIAM J. Control*, **4**, No.1, 139–158, 1966.

11. FIACCO, A., and McCORMICK, G., *Nonlinear programming*, Wiley, 1968.

12. EVTUSHENKO, Yu.G., Numerical methods for solving problems of non-linear programming, *Zh. vȳchisl. Mat. mat. Fiz.*, **16**, No.2, 307–324, 1976.

13. ROCKAFELLAR, R.T., Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM J. Control*, **12**, No.2, 268–285, 1974.

14. GOLIKOV, A.I., and EVTUSHENKO, Yu.G., A class of methods for solving problems of nonlinear programming, *Dokl. Akad. Nauk SSSR*, **239**, No.3, 519–522, 1978.

15. MORRISON, D., Optimization by least squares, *SIAM J. Numer. Analys.*, **5**, No.1, 83–88, 1968.

16. GRIFFITH, R.E., and STEWART, R.A., A nonlinear programming technique for the optimization of a continuous processing system *Manag. Sci.*, **7**, No.4, 379–392, 1961.

17. PSHENICHNYI, B.N., and DANILIN, Yu.M., *Numerical methods in extremal problems* (Chislennye metody v ekstremal'nykh zadachakh), Nauka, Moscow, 1975.

18. FEDORENKO, R.P., Approximate solution of some optimal control problems, *Zh. vȳchisl. Mat. mat. Fiz.*, **4**, No.6, 1045–1064, 1964.

19. KRYLOV, I.A., and CHERNOUS'KO, F.L., Algorithm of the method of successive approximations for optimal control problems, *Zh. vȳchisl. Mat. mat. Fiz.*, **12**, No.1, 14–34, 1972.

20. DAVYDOV, E.G., Distribution of resources in graphs, in: *Systems of distribution of resources in graphs* (Sistemy raspredeleniya resursov na grafakh), 31–48, VTs Akad. Nauk SSSR, Moscow, 1970.

21. MAGNANTI, T.L., Nonlinear programming and the maximum principle for discrete time optimal control problems, *Rev. franc. Automat. inform. Rech. oper.*, No.3, 75–91, 1975.

22. GRACHEV, N.I., and EVTUSHENKO, Yu.G., Packet of programs for solving optimal control problems, VTs Akad. Nauk SSSR, Moscow, 1978.