

A Multimethod Technique for Solving Optimal Control Problem

Alexander I. Tyatyushkin

Abstract A multimethod algorithm for solving optimal control problems is implemented in the form of parallel optimization processes with the choice of the best approximation. The multimethod algorithm based on a sequence of different methods is to provide fast convergence to an optimal solution. Such a technology allows one to take into account some particularities of the problem at all stages of its solving and improve the efficiency of optimal control search.

Key words Optimal control • Multimethod algorithms • Parallel computations • Software packages • Numerical methods

1 Introduction

The technology of finding the numerical solution to the applied optimal control problems is based on universal software which has a well-developed interface and a rich arsenal of optimization methods. Such software allows one to take into account specific features of the problem under consideration by making the use of diverse algorithms of improvement at different stages of iteration process. Application of several numerical methods for solving a single optimization problem was suggested in many publications oriented to the software development [1–3, 6, 7, 11].

Principal difficulty in applying the multimethod algorithm lies in the fact that at each stage of the problem-solving process one has information about efficiency of the method applied at the present moment. To determine the efficiency of any optimization method at some stage of searching for solution to the given problems

A.I. Tyatyushkin (✉)

Institute for System Dynamics and Control Theory SB RAS, 134 Lermontov St.,

664033 Irkutsk, Russia

e-mail: tjat@icc.ru

it is necessary to perform one or several iterations. Therefore, to choose the method which is more appropriate for the given stage of problem solving, the operation of switching from one method to another is usually repeated.

Also, it is necessary to know about switching times. But this information can be easily obtained by tracking the current method measuring parameters characterizing its convergence.

Thus, the principal problem of the multimethod technology is the choice of method which allows one to continue effectively the optimization process from the moment when convergence of current method was impaired.

Modern operational systems provide a solution to the given problem by organizing parallel computational flows for simultaneous computation by several methods. In each flow, one can realize iterative process of one method from a collection of methods. Thus, a single problem can be solved by several methods. With the multiprocessor technology on hand, of course, it is convenient to use individual processor for accomplishing iteration of each method.

After finding the next approximation, each method is considered, for instance, evaluating an increment of the functional. More effective method is taken to continue the optimization. Next, the approximation obtained by this method is transferred to other methods as initial data to perform next iteration. Starting from this approximation, one or several iterations are again performed by all methods. Out of the obtained approximations again we take the one in which the functional has a smaller value.

Continue iterative process until the optimum criterion is met for the obtained approximation. After that we find an approximation solution to the problem under consideration. In this case the solution is found by the multimethod algorithm consisting of a sequence of steps of different methods attached to optimization process to accelerate its convergence. The advantage of the multimethod algorithm in comparison with each method separately lies in its greater adequacy in application to concrete problem. At each stage of searching for solution, the multimethod algorithm makes the use of optimization method which is more suitable in terms of specific features of a given problem (e.g., the ravines of function, specific character, and the structure of constraints).

In the graphic, the decrease in the functional $I(u)$, in iterations of the multimethod algorithm, is shown by the broken line which consists of the graphics of separate methods. Figure 1 shows the multimethod algorithm operation in the case, when two methods $M1$ and $M2$ are used. The graphics given show the decrease in the functional in the iterations of the methods $M1$ and $M2$. The graphic of decrease in the functional in the iterations of the multimethod algorithm is the curve ABC . It is constructed by using two graphics corresponding to the methods $M1$ and $M2$. Namely, the region BC is obtained by parallel translation of the region EL . According to this figure, the zero value of functional is achieved in k_1 iterations of the method $M1$, while the use of the methods $M1$ and $M2$ requires k_2 iterations.

The multimethod algorithm works, up to the \bar{k} th iteration, by the method $M1$ (the curve AB) and then by the method $M2$ (the curve BC). The reason is that

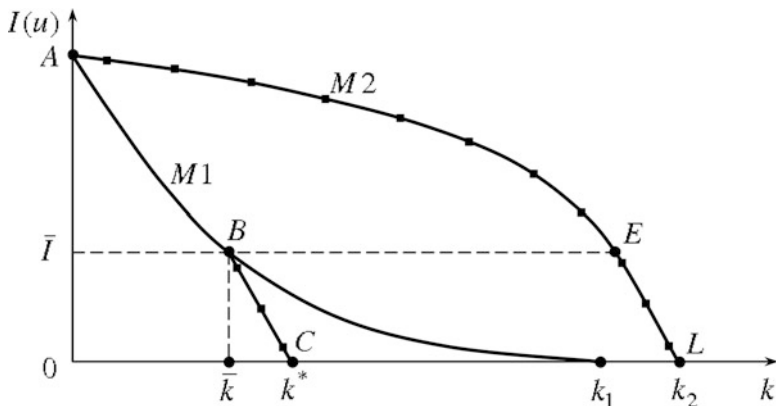


Fig. 1 Graphics of decrease of functional on the iterations of the methods $M1$, $M2$ and multi-method algorithm

beginning with the \bar{k} th iteration, the velocity of decrease in the functional during the use of the method $M2$ is higher. As a result, zero value of functional is achieved by the multimethod algorithm using k^* iterations. This is considerably less than in the case where the methods $M1$ and $M2$ are used individually.

2 Parallel Computations in the First-Order Methods

By different criteria for choosing the closest optimization method and also organizing in different ways parallel computations on the method's iterations, several different combinations of algorithms can be obtained to solve a single problem. Moreover, it is possible to construct the multimethod algorithms that do not contain repeated computations in the iterations of different optimization methods. For example, in the methods of gradient type [4, 13], laborious computations of the gradient, requiring an integration of the adjoint system, should be performed only once; then, the obtained gradient should be used in the iterative formulas of all methods. In this case, computational expenditures at one step of multi-flow algorithm are considerably reduced. Moreover, realization of the step by any of the methods is accomplished by using the same approximately obtained values. Then all optimization algorithms are applied as if to the one and the same approximate model. Thus, the criterion of new approximation is defined only by the optimization methods. Otherwise, because of computational errors, the same parameters used to estimate convergence of the methods may have different values which can lead to improper choice of the best method.

Let us consider an optimal control problem provided by the conditions in the form of equalities at the right trajectory end. The controlled process is described by the system

$$\dot{x} = f(x, u, t), \quad x(t_0) = x_0, \quad t \in T = [t_0, t_1], \quad x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^r \quad (1)$$

with terminal conditions

$$I_j(u) = \varphi^j(x(t_1)) = 0, \quad j = \overline{1, m} \quad (2)$$

and with phase constraints

$$J_i(u, t) = g^i(x, t) = 0, \quad i = \overline{1, s}, \quad t \in T. \quad (3)$$

The control is constrained through

$$u(t) \in U, \quad (4)$$

where U is a bounded closed set in \mathbb{R}^r . Vector functions $f(x, u, t)$ are assumed to be differentiable w.r.t. to x and u and continuous w.r.t. t ; $\varphi^j(x)$, $j = \overline{1, m}$ are assumed to be continuously differentiable w.r.t. x functions.

It is required to find the control among controls fulfilled by (3), such that provides the validity of conditions (2), for controlled process (1) and, on the other hand, provides the minimum of the functional

$$I_0(u) = \varphi^0(x(t_1)), \quad (5)$$

where $\varphi^0(x)$ is continuously differentiable function.

The gradients of functionals $I_j(u)$, $j = \overline{0, m}$ in terms of the functions $H^j(\psi_j, x, u, t) = \psi_j'(t)f(x, u, t)$ and adjoint system

$$\dot{\psi}_j = -f_x(x, u, t)' \psi_j(t), \quad \psi_j(t_1) = -\psi_x^j(x(t_1))$$

are given by the formula

$$\nabla I_j(u) = -H_u^j(\psi_j, x, u, t), \quad j = \overline{1, m}. \quad (6)$$

For each $t \in T$ one can calculate in much the same way the gradients $J_j(u, t)$, $j = \overline{1, s}$:

$$\nabla J_j(u, t) = -\overline{H}_u^j(\Phi_j, x, u, t, \tau), \quad t_0 \leq \tau \leq t \leq t_1, \quad (7)$$

where $\overline{H}_u^j(\Phi_j, x, u, t, \tau) = \Phi_j'(t, \tau)f(x, u, \tau)$, $\Phi_j(t, \tau)$, $j = \overline{1, s}$ are the solutions of the conjugate system

$$\frac{\partial \Phi_j(t, \tau)}{\partial \tau} = -\frac{\partial f(x, u, \tau)}{\partial x} \Phi_j(t, \tau), \quad \tau \in T,$$

with the boundary conditions $\Phi_j(t, t) = -\frac{\partial g^j(x(t))}{\partial x}$, $j = \overline{1, s}$.

2.1 Application of Gradient Methods

Gradient procedure of the minimization of the functional (4) without taking into account the constraints (2) and (3) is given through relation

$$u^{k+1} = u^k - \alpha_k \nabla I_0(u^k),$$

where α_k are chosen, for example, from condition of fastest decrease of the functional $I_0(u)$. The solution of the problem with terminal conditions (2) without constraints (3) can be obtained by parallel application of the linearization and penalization methods given, for example, in [7]. When using penalization method, penalty functional which consists of the functions (2) and (4) is minimized with the help of gradient procedure. Making use of the multimethod technology, one can also simplify each iteration of the linearization method in such a way that its working time will be close to the one of the penalization method. Then the algorithm will be as follows:

1. For given $u^k(t), t \in T$, system (1) is integrated; in the integration points, the phase coordinates of trajectory $x^k(t)$ are memorized.
2. $m + 1$ flows are organized, for parallel integration of adjoint system provided by different initial conditions $\psi_j(t_1) = -\psi_x^j(x(t_1)), j = \overline{0, m}$. In the process of integration, the solutions $\psi_j(t)$ are used to construct a linear system of algebraic equations

$$\sum_{i=1}^m \left(\int_{t_0}^{t_1} H_u^{j'} H_u^i dt \right) \lambda_i = I_j(u^k) - \int_{t_0}^{t_1} H_u^{j'} H_u^0 dt, \quad j = \overline{1, m}.$$

3. After solving this system, the values of variables $\lambda_i, i = \overline{1, m}$ are found.
4. A new approximation of the control

$$u^{k+1} = u^k + \alpha_k \delta u, \quad \delta u = H_u^0 + \sum_{i=1}^m \lambda_i H_u^i,$$

is constructed, where the parameters α_k satisfy inequality

$$I_0(u^k + \alpha_k \delta u) + \beta I_{j_0}(u^k + \alpha_k \delta u) \leq I_0(u^k) + \beta I_{j_0}(u^k) - \varepsilon \int_{t_0}^{t_1} \delta u' \delta u dt,$$

$$0 < \varepsilon < 1, \quad j_0 = \arg \max_{1 \leq j \leq m} |I_j(u^k)|, \quad \beta = \sum_{i=1}^m |\lambda_i|.$$

From this algorithm, as particular case (for $m = 0$), we obtain the ordinary gradient method.

2.2 The Methods for Solving Problem with Constraints on Control

Let us focus on the algorithms intended for solving the problems provided by constraints on control, but with free right end. Suppose that for some $u^k(t) \in U$, $t \in T$, one finds any solution to the system (1) $x^k(t)$, $t \in T$. Setting in (5) $j = 0$, integrate adjoint system from $t = t_1$ to $t = t_0$ when $u = u^k(t)$, $x = x^k(t)$. Calculate on its solution $\psi^k = \psi^0(t)$ the control using the maximum principle:

$$\bar{u}^k(t) = \operatorname{argmax}_{u \in U} H(\psi^k, x^k, u, t), \quad t \in T,$$

and find the value of scalar function

$$w_k(\bar{u}(t), t) = H(\psi^k, x^k, \bar{u}, t) - H(\psi^k, x^k, u^k, t), \quad t \in T.$$

Let $t = \tau_k$ the maximum point of this function being on T . Then, the necessary optimum condition of the control u^k will be

$$w_k(\bar{u}^k(\tau_k), \tau_k) = 0.$$

In the case when for given u^k and obtained x^k , ψ^k , \bar{u}^k , the maximum principle

$$w_k(\bar{u}^k(\tau_k), \tau_k) > 0$$

does not hold; iteration of the method [12] can be made to improve u^k .

Denote the point set, where maximum principle is broken by

$$T_\varepsilon = \left\{ t \in T : w_k(\bar{u}^k(t), t) \geq \varepsilon w_k(\bar{u}^k(\tau_k), \tau_k) \right\}, \quad \varepsilon \in [0, 1].$$

Observe that at $\varepsilon = 0$, we have $T_\varepsilon^k = T$, while at $\varepsilon = 1$, the set T_ε^k consists of the maximum points of the function $w_k(u(t), t)$.

By varying ε , we can find such value for which the control

$$u_\varepsilon^k = \begin{cases} \bar{u}^k(t), & t \in T_\varepsilon, \\ u^k(t), & t \in T \setminus T_\varepsilon \end{cases} \quad (8)$$

provides the least value of the objective functional $I_0(u)$, i.e.,

$$\varepsilon_k = \operatorname{argmin}_{\varepsilon \in [0, 1]} I_0(u_\varepsilon^k).$$

When searching for ε_k , several flows can be used for simultaneous integration of system (1) with controls (8), corresponding to different values of $\varepsilon \in [0, 1]$. In addition, at $t = t_1$, we have different phase space points $x_\varepsilon^k(t_1)$ and pertinent values $I_0(u_\varepsilon^k) = \varphi(x_\varepsilon^k(t_1))$. After the smallest value of the functional $I_0(u_\varepsilon^k)$ is chosen,

we verify the inequality $I_0(u_\varepsilon^k) < I_0(u^k)$, and if it holds, we assume $u^{k+1} = u_\varepsilon^k$. Otherwise, the subdivision of ε can be continued, and the values of functional for the following values can be found.

By virtue of structure of the controls generated by the iteration formula (8), the relaxation of an algorithm can be impaired even before the control satisfying the maximum principle is obtained. Therefore, to continue optimization process, it is necessary to apply another algorithm, in iteration of which the controls are constructed not only with boundary points but also with interior ones w.r.t. the set U as well. For example, the convergence can be restored by constructing a convex combination of two controls

$$u^{k+1}(t) = u^k(t) + \alpha [\bar{u}^k(t) - u^k(t)], \quad \alpha \in [0, 1]. \tag{9}$$

The calculations by the formulas (8) and (9) can be made simultaneously by choosing from the obtained approximations such u^{k+1} to which the smallest value of the functional corresponds. In the case, where the functional values are compared within several iterations, the values of increase in the functional, obtained in the neighboring iterations of each method, should be used as a criterion to compare the efficiency of the methods (8) and (9).

In practice, it is established that the application of the variations of two types, namely, “horizontal” (8) and “vertical” (9), allows us to avoid the effect of “control sticking to the boundaries” which is inherent in the algorithms based on the maximum principle [5, 12].

In the case in the iteration equation (9), the control $\bar{u}^k(t)$ is derived from the linearized maximum principle

$$\bar{u}^k(t) = \arg \max_{u \in U} H_u(\psi^k, x^k, u, t)' u(t), \quad t \in T; \tag{10}$$

we obtain the iterations of the conditional gradient method. It is evident that for the systems which are linear in control, the control function (10) coincides with that deduced from the maximum principle. Another algorithm of control improvement can be obtained by substituting, in the iteration formula (8), the interval T_ε for the following one:

$$T_\varepsilon^k = [\tau_k - \varepsilon(\tau_k - t_0^k), \tau_k + \varepsilon(t_1^k - \tau_k)], \quad \varepsilon \in [0, 1], \tag{11}$$

where t_0^k, t_1^k are the nearest left and right discontinuity points of the function $w(\bar{u}^k(t), t)$.

In the process of finding the value of parameter ε_k providing convergence of the algorithm, we can apply the above procedure along with parallel computations. This way of constructing the interval provides blowing down of its ends towards the point τ_k , in the case, if the function $w(\bar{u}^k(t), t)$ is constant within some neighborhood of the point τ_k , thus maintaining the convergence of the algorithm [12].

2.3 Linearization Method for Solving Problems with Phase Constraints

Let $u_k(t)$ be a current approximation of the control, and let $x_k(t)$ be the phase trajectory, corresponding to $u_k(t)$, $t \in T$. Using the gradients (6) and (7) we linearize the conditions (2) and (3) in the neighborhood of u_k :

$$I_i^L(u^k, u) = I_i(u^k) + \int_{T_0}^{T_1} \nabla I_i(u^k, t)' (u(t) - u^k(t)) dt = 0, \quad i = \overline{1, m}, \quad (12)$$

$$J_j^L(u^k, u, \tau) = J_j(u^k, \tau) + \int_{t_0}^{\tau} \nabla J_j(u^k, t)' (u(t) - u^k(t)) dt = 0, \quad (13)$$

$$j = \overline{1, s}, \quad \tau \in T.$$

Construct a modified Lagrange function for the problem (1)–(5) in the form:

$$\begin{aligned} L(u, u^k, \lambda^k, \mu^k) = & I_0(u) - \lambda^{k'} (I(u) - I^L(u^k, u)) \\ & - \int_{t_0}^{t_1} \mu^k(t)' (J(u, t) - J^L(u^k, u, t)) dt \\ & + \frac{\rho}{2} (I(u) - I^L(u^k, u))' (I(u) - I^L(u^k, u)) \\ & + \frac{\rho}{2} \int_{t_0}^{t_1} (J(u, t) - J^L(u^k, u, t))' (J(u, t) - J^L(u^k, u, t)) dt, \end{aligned} \quad (14)$$

where I, I^L — m -vectors; J, J^L — s -vectors; λ^k, μ^k are m - and s -dimensional Lagrange factors; $\rho \geq 0$ is a penalty coefficient.

In the $k + 1$ -st iteration of the considered method we solve the minimization problem of functional (14) on the solutions of the system (1) with the linear constraints (12), (13), and (4). By solving the problem, we determine new values of the Lagrange factors λ_i^{k+1} , $i = \overline{1, m}$, $\mu_i^{k+1}(t)$, $i = \overline{1, s}$, $t \in T$.

After determining u^{k+1} , λ^{k+1} , and μ^{k+1} , we again linearize constraints (2), and (3) in the neighborhood of u^{k+1} , construct the functional $L(u, u^{k+1}, \lambda^{k+1}, \mu^{k+1}, \rho)$, and repeat the iteration.

Thus, the algorithm for solving the formulated problem consists of the following main operations:

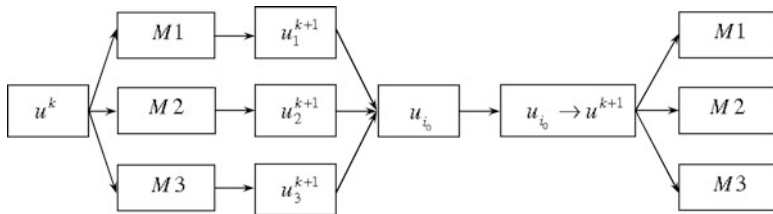


Fig. 2 The scheme of realization of $(k + 1)$ -th iteration by multimethod algorithm using three methods $M1$, $M2$, and $M3$

1. Linearizing the constraints and solving the auxiliary problem (1), (4), and (12)–(14). Calculating the Jacobian of the linear system (13) is computationally very expensive.
2. Verifying the optimal conditions for the solution obtained on the k th iteration.

When solving the problems with linear constraints the application of this algorithm is greatly simplified because it is not necessary to calculate the constrained Jacobian.

2.4 The Block Scheme of Multimethod Algorithm Operation

Summarizing the above said we see that by applying various iteration procedures and making use of different rules to construct the sets of varying controls, we obtain the collection of algorithm, each working effectively enough only in a certain situation. Thus, in the process of finding the optimal control, it is necessary to include several algorithms.

By organizing parallel computations to realize some collection of algorithms and applying the selection procedure to take the best approximation after simultaneous iterations by all methods, we are able to find effectively optimal control by the multimethod algorithm.

Figure 2 demonstrates how the multimethod algorithm works in the case when three methods are used. The block of selection of the best approximation finds u_{i_0} from a largest value of increment of the functional obtained in the $(k + 1)$ -th iteration

$$u_{i_0} = \operatorname{argmax}_{i \in (1, 2, 3)} (I(u^k) - I(u_i^{k+1})).$$

This approximation is passed to all methods $u_i^{k+1} = u_{i_0}$, $i = 1, 2, 3$, to perform the next iteration.

It should be noted that another multimethod algorithm can be generated from the collection of methods for solving another problem. The algorithm can be more adequate because of taking into account the specific features of this problem.

3 Implementation of Multimethod Algorithm

3.1 Solving the Adjoint Problem

The most labor-consuming operation performed at each step of all first-order algorithms is numerical integration of original and adjoint system of differential equations provided by some control. The solving of adjoint system in the iterations is used to find either the value of Hamiltonian or both to perform the calculation of gradients of functionals. Thus, the numerical integration of this system is accomplished at each step of any of the first-order methods. Since the multimethod technology enables the steps to be made simultaneously by all methods, the solution of adjoint system, which is obtained by single integration, is used in all iteration formulas simultaneously.

3.2 Computation of the Method Step

Realization of each step of the first-order method needs to find the value of method's step α or ε , for which new obtained approximation provides the smallest value of objective functional. The search for such a value of α requires multiple integration of initial system (1). By constructing control u^α by given iteration formula at different values of α , and integrating initial system, at $u = u^\alpha$, we obtain various trajectories $x^\alpha(t)$, $t \in T$ and find pertinent values of the functional $I(u^\alpha)$. By applying the one-dimensional search method it is possible by several such recalculations to find the approximate value of $\alpha = \alpha_k$ such that the minimum of $I(u^\alpha)$ w.r.t. parameter α is provided. If, in the process, the inequality $I(u^{\alpha_k}) < I(u^k)$ holds, the control u^{α_k} is taken as new approximation u^{k+1} . Otherwise, iteration process, using this method, comes to an end.

To recover the method convergence it is necessary to correct calculations, namely, diminish the integration step, increase the accuracy of one-dimensional search, and so on. The multimethod algorithm can be used to continue the process of control improvement by switching to another method, while the condition for termination of its work is impossibility to guarantee relaxation by none of the methods entering in the given collection.

Another scheme of search can also be used to find the value of α_k while using the multimethod algorithm. Its essence consists in the fact that in the process of subdivision of α , for example, by cutting in two, with each its value, it accomplishes test step by all methods. Since, in many algorithms, for finding α_k , it accomplishes subsequent subdivision in two until the inequality $I(u^{\alpha_k}) < I(u^k)$ holds, then, for any fixed α , the validity of this inequality is checked simultaneously for all methods. In addition, with given value of α , in correspondence with iteration formula of each method, it is constructed the control u^α integrated the system (1). Its solution is used to calculate the functional $I(u^\alpha)$. If, in the process, for some value of parameter α ,

for some of the methods, the required inequality holds, then u^{α_k} , obtained by this method, should be taken as a new approximation to be passed to all methods for continuation of iteration process.

3.3 Choice of Optimization Methods

For parameters to be used to estimate the efficiency of iteration process, one can take, for instance, the velocity of reduction of the residuals in optimization conditions, the value of increment of the objective functional, or the extent of violation of some important, for example, in the physical sense, constraints. The accuracy of calculations of these parameters provides the proper choice of the method and opportune transition to another optimization method.

3.4 The Methods of Approximation of Control

The principal error in modeling the control problem arising in the process of numerical solving is due to discrete approximation of controlled dynamic system and tabular representation of control function.

The methods of numerical integration allow for discretization of the system with given accuracy but provided that control functions are continuous and their values can be determined for arbitrary $t \in T$. However, in practice, piecewise constant approximation of control functions is often applied, and its values are defined only in given sites of the temporal network. These values are changed in iteration process, depending on optimization method to decrease minimized functional or to reduce the residuals for given terminal conditions. The values of control functions between the points, which are necessary, for example, to apply the numerical methods of Runge–Kutta type, as usual are taken to be equal either the value in the nearest left point or both are defined by linear interpolation by the values in left and right points. In this process, the error of numerical integration can be considerably extended. Then, on the obtained numerical solution of the system, the values of parameters, which are used to choose the method, could be incorrect.

To diminish the errors of calculation of trajectory, one can condense the temporal network in order to increase the number of desired values of control which implies to solve optimization problem of a big size. In the case if required control is smooth enough, then, with some number of points and with the help of interpolation formulas, the admissible accuracy of its approximation can be provided. However, in general case, if control functions are discontinuous, for example, of relay type, this approach may turn out to be inefficient, since, in this case, the condensation of network affects the approximation accuracy only in the neighborhood of the switching points. Therefore, in the process of solving the bang–bang optimal control problems, it is necessary to use the procedure of switching times correction to

provide the approximation for control functions with prescribed accuracy. In the absence of this procedure, the control is found in the form of the array of numbers defined on temporal network. Such a control may differ significantly from the optimal one both by the value of objective functional and by the accuracy of optimal conditions fulfillment.

3.5 *The Estimation of Accuracy of a Chosen Method*

The application of multimethod algorithm is correct only if all methods use the same approximation of the control and provide the same accuracy of integration of the systems. In this case, all optimization methods are used to solve the same finite dimensional problem obtained by discretization, while the values of parameters used to choose the method give a more correct estimate of the algorithm efficiency, providing thus a correct choice of the method to continue iteration process. As a result, the approximate solution is obtained in a lesser number of iterations, in comparison to individual use of each method from a given collection. This is because at each stage of solving the considered problem, a more effective algorithm is applied to be more adequate in this situation.

4 The Numerical Experiments

Let us present the results of the numerical experiments to demonstrate application of the multimethod technology. Two examples are considered: a test problem of the Rosenbrock function minimization and the problem of optimal control of the rocket flight.

4.1 *Example 1*

It is required to find the minimum of Rosenbrock function $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ with constraint $(x_1 - 2)^2 + (x_2 - 2)^2 \leq 1$ and initial approximation $x^0 = (2, 1)$.

It is known that the absolute minimum of this function is achieved at the point $(1, 1)$. It is equal to 0. In the problem under consideration, the point $(1, 1)$ is not feasible. Therefore the zero value of objective functional is not attained.

This problem was solved numerically by the method of conditional gradient, by the method of gradient projection, and also by the multimethod algorithm which contains these two methods. The smallest value of the function equal to 0.0358, for prescribed accuracy $\varepsilon = 10^{-4}$, was attained using 240, 341, and 170 iterations, respectively. Minimum point is $x^* = (1.189, 1.415)$.

4.2 Example 2

In this case, controlled process is described by the system

$$\begin{aligned}\dot{x}_1 &= -cpx_1^2[1.174 - 0.9\cos u_1] - g\sin x_2/(1+x_3)^2, \\ \dot{x}_2 &= 0.6cpx_1^2\sin u_1 + x_1\cos x_2/[R(1+x_3)] - g\cos x_2/[x_1(1+x_3)^2], \\ \dot{x}_3 &= x_1\sin x_2/R, \\ \dot{x}_4 &= x_1\cos x_2/[1+x_3],\end{aligned}$$

where $p = 0.002704\exp(-4.26Rx_3)$, $R = 209$, $g = 0.00032172$, and $c = 26600$.

The initial data $x(0) = (0.36, -0.141372, 0.019139, 0.0)$ and terminal conditions $x(t_k) = (0.27, 0.0, 0.011962, \text{unbounded})$ are given, while the parameter t_k is not fixed. It is required to find such control $u(t)$, $t \in [0, t_k]$ and such smallest value of parameter t_k that the fulfillment of terminal conditions is guaranteed, while the functional

$$I(u) = \int_0^{t_k} x_1[\exp(-4.26Rx_3)]^{1/3} dt$$

should attain the smallest value.

This problem was solved by the above linearization method and also by the projected Lagrangian method, in iteration of which the nonlinear Lagrangian with linearizable constraints is minimized [11]. Approximate solution (the accuracy w.r.t. boundary conditions is 10^{-3}) was found in 282 and 215 iterations, respectively. The solution with the same accuracy, that was obtained with the multimethod algorithm including these two methods, was found in 164 iterations. The value of parameter t_k equals 72.412, while the functional $I(u)$ attains the value equal to 24.59.

5 Conclusion

Thus, we can conclude that for each problem under consideration there exists an appropriate sequence of steps based on different methods which provides more effective search for the optimal control. In the multimethod algorithms the construction of such a sequence is accomplished automatically according to some given criterion estimating the efficiency of optimization process at each stage of the problem solving. The use of the technology described above is based on the application software, for example, [5–11, 14], which includes the methods of first and second order for solving the optimal control problems with constraints of different types.

References

1. Evtushenko, Yu.G.: Methods of solving of extremal problems and its application. Nauka, Moscow (1982).
2. Gurman, V.I., Dmitri'ev, M.G., Osipov, G.S.: Intellectual multimethod's technology for solving and analysis of control problems: Preprint of Institute of programmed systems of RAS. Pereslavl-Zallesskii (1996).
3. Ling, L., Xue G.: Optimization of Molecular Similarity Index with Applications to Biomolecules. *J. Glob. Optim.* **14**(3), 299–312 (1999)
4. Lyubushin, A.A., Chernous'ko, F.L.: The method of successive approximations for computation of optimal control. *Izv. AN SSSR. Tehn. kibernetika.* 2, 141–159 (1983).
5. Morzhin, O.V., Tyatyushkin, A.I.: An algorithm of the section method and program tools for reachable sets approximating. *J. of Computer and Systems Sciences International.* **47**(1), 1–7 (2008).
6. Tyatyushkin, A.I.: Package KONUS for optimization of continuous controlled systems. The packages of applied softwares: The experience of using. Nauka, Moscow (1989).
7. Tyatyushkin, A.I.: Numerical methods and software for optimization of controlled systems. Nauka, Novosibirsk (1992).
8. Tyatyushkin, A.I.: Numerical methods for optimization of controlled systems. *J. Stability and control: Theory and Appl.* **3**(2), 150–174 (2000)
9. Tyatyushkin, A.I.: Parallel computations in Optimal control problems. *Siberian J. of Number.Mathematics (Sib.Branch of Russ. Acad. of Sci).* **3**(2), 181–190, Novosibirsk (2000).
10. Tyatyushkin, A.I.: Many-Method Technique of Optimization of Control Systems. Nauka, Novosibirsk (2006).
11. Tyatyushkin, A.I., Zholudev, A.I., Erincbek, N.M.: The program system for solving optimal control problems with phase constraints. *Intern. J. of Software Engineering and Knowledge Engineering.* **3**(4), 487–497 (1993).
12. Vasil'ev, O.V., Tyatyushkin, A.I.: On some method of solving of optimal control problems based on maximum principle. *J. vychisl. matem. i mat. fiziki.* **21**(6), 1376–1384 (1981).
13. Yuan, G.: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. *Optim. Lett.* **3**(1), 11–21 (2009).
14. Zholudev, A.I., Tyatyushkin, A.I., Erincbek, N.M.: Numerical optimization methods of controlled systems. *Izv. AN SSSR. Tehn. kibernetika.* 4, 14–31 (1989).