The Technology of Calculating the Optimal Modes of the Disk Heating (Ball)

Yury Alexandrovich Kostikov a, Alexander Mikhailovich Romanenkov b*

a Candidate of Physical and Mathematical Sciences, Head of the Department 812, Moscow Aviation Institute (National Research University), 125993, Volokolamskoye shosse, 4, Moscow, A-80, GSP-3, Russia.

b Candidate of Technical Sciences, Associate Professor, Department 812, Moscow Aviation Institute (National Research University), 125993, Volokolamskoye shosse, 4, Moscow, A-80, GSP-3, Russia.

Received 05 February 2019; Accepted 04 June 2019

Abstract

The paper considers the problem of optimal control of the process of thermal conductivity of a homogeneous disk (ball). An optimization problem is posed for a one-dimensional parabolic type equation with a mixed-type boundary condition. The goal of the control is to bring the temperature distribution in the disk (ball) to a given distribution in a finite time. To solve this problem, an algorithm is proposed that is based on the gradient method. The object of the study is the optimal control problem for a parabolic boundary value problem. Using the discretization of the original continuous differential problem, difference equations are obtained for which a numerical solution algorithm is proposed. Difference approximation of a differential problem is performed using an implicit scheme, which allows to increase the speed of calculations and provides the specified accuracy of calculation for a smaller number of iterations. An approximate solution of a parabolic equation is constructed using the one-dimensional sweep method. Using differentiation of the functional, an expression for the gradient of the objective functional is obtained. In this paper, it was possible to reduce the multidimensional heat conduction problem to a one-dimensional one, due to the assumption that the desired solution is symmetric. A formula is obtained for calculating the variation of a quadratic functional that characterizes the deviation of the current temperature distribution from the given one. The flowcharts and implementations of the algorithm are presented in the form of Matlab scripts, which clearly demonstrate the process of thermal conductivity and show the computation and application of optimal control in dynamics.

Keywords: Optimal Control; Parabolic Equation; Gradient Method; Software Complex for Calculating Optimal Modes.

1. Introduction

Optimization problems are found in almost all spheres of human activity, since any activity must be effective in a certain sense. That is, an action plan must be chosen that ensures optimality, according to the chosen criterion. The search for optimal solutions led to the creation of special mathematical methods and the mathematical foundations of optimization (calculus of variations, numerical methods, etc.) were laid already in the 18th century. However, until the second half of the 20th century, optimization methods were used very rarely in many areas of science and technology, since the practical use of mathematical optimization methods required tremendous computational work, which was extremely difficult to implement without a computer, and in some cases impossible. If we do not take into account the economic, physical, chemical or other content of these tasks, then all tasks are reduced to the following optimization problem: to find the minimum (or maximum) of a function or functional on some admissible set of a given functional

*Corresponding author: romanaleks@gmail.com

http://dx.doi.org/10.28991/cej-2019-03091340

© 2019 by the authors. Licensee C.E.J, Tehran, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).
space. That is, the value of the functional expresses the quality of management, and the allowable set is determined by the constraints on resources, the possibilities of economic or other processes in the system under study [1-5].

This paper is devoted to the problem of building a numerical solution and developing a script in the Matlab language to determine the optimal control of the process of heating a homogeneous disk. One of the main goals and important results of the work is the description of an actually universal algorithm for solving similar optimization problems. When solving specific tasks, it is necessary to change the initial and boundary conditions, difference expressions, and the objective function, but the sequence of actions performed is preserved [6-8].

2. Formulation of the Problem

The problem of optimal control of a non-stationary process of heat conduction in a disk and in a ball is considered. There is axial symmetry in the first case, central symmetry in the second [9-11].

![Figure 1. Axial symmetry](image1.png)

![Figure 2. Central symmetry](image2.png)

In this case, the model of the process of heat conduction is described by a one-dimensional parabolic equation:

$$\frac{\partial \varphi}{\partial t} = \frac{1}{r^n} \frac{\partial}{\partial r} \left( r^n k(r) \frac{\partial \varphi}{\partial r} \right)$$  \hspace{1cm} (1)

Where $n = 1$ – for a flat case (see Figure 1), $n = 2$ – for the three-dimensional case (see Figure 2). $\varphi(t,r)$ – body temperature $r$ at a point in time $t$, $r = R\sqrt{x_0 + y_0}$, $(0 \leq x_0 \leq 1)$, $(0 \leq y_0 \leq 1)$, $t_k$ – final moment of time $(0 \leq t \leq t_k)$, $R$ – cylinder or ball radius $(0 \leq r \leq R)$, $k$ – coefficient of thermal conductivity.

When $t = 0$, the initial condition is set:

$$\varphi(0, r) = \varphi_0(r)$$  \hspace{1cm} (2)

When $r = R$ the boundary condition is:

$$\frac{\partial \varphi(t,R)}{\partial r} + \alpha \varphi(t, R) = \alpha U(t)$$  \hspace{1cm} (3)

$U(t)$ - The temperature at the border, which is completely at our disposal. In this work, $U(t)$ is a control function depending on time, and $\alpha$ is a heat transfer coefficient.

When $r = 0$, the condition of limitation is set, that is $|\varphi(t,0)| < \infty$, which is a consequence of the continuity and differentiability of the equation solution (1):

$$\lim_{r \to 0} r^n k(r) \frac{\partial \varphi(t,r)}{\partial r} = 0$$  \hspace{1cm} (4)

In this paper, we consider the problem of bringing the disk temperature to a given temperature to a finite point in time. It is required to find the function $\varphi(t,r)$, which is a solution to Equation (1) and satisfies conditions (2) - (4). Find the admissible control $U(t)$, the function $U(t)$, satisfying the constraints of the form:

$$0 < U^- \leq U(t) \leq U^+$$  \hspace{1cm} (5)

Where $U^+$ and $U^-$ - given constants, so that by a given time point $t_k$ the temperature distribution in the region should be made as close as possible to the given distribution $\varphi_0(r)$, $0 \leq r \leq R$.

Consider the functional characterizing the temperature deviation at a given time from the target temperature. It is necessary to define the function $U(t)$, which delivers a minimum to this functional:
\[
J = \int_0^R r^n [\varphi(t_k, r) - \varphi_g(r)]^2 dr + \beta \int_0^{t_k} U(t)^2 dt \tag{6}
\]

The construction of the required control function \(U(t)\) will be carried out based on the condition that the first variation of the functional \(J\) is equal to zero. To do this, we hover the control \(U(t)\):

\[
U(t) \rightarrow U(t) + \delta U(t)
\]

Then the function \(\varphi(t, r) \rightarrow \varphi + \delta \varphi\), and the functional \(J + \delta J\) change. In this case, Equation (1) for the new function \(\varphi\) will remain the same:

\[
\frac{\partial (\varphi + \delta \varphi)}{\partial t} = \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial (\varphi + \delta \varphi)}{\partial r} \right) \tag{7}
\]

And the functional will take the form:

\[
J + \delta J = \int_0^R r^n [\varphi(t_k, r) + \delta \varphi(t_k, r) - \varphi_g(r)]^2 dr + \beta \int_0^{t_k} (U(t) + \delta U(t))^2 dt
\]

Next, subtract from Equation 7 to Equation 1:

\[
\frac{\partial (\varphi + \delta \varphi)}{\partial t} \bigg|_{t=0} = \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial \varphi}{\partial r} \right)
\]

From here, we obtain for \(\delta \varphi(t, r)\) an equation of the following form:

\[
\frac{\partial \delta \varphi}{\partial t} = \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial \delta \varphi}{\partial r} \right) \tag{8}
\]

With appropriate boundary and initial conditions:

\[
\delta \varphi(0, r) = 0, \quad \lim_{r \to 0} \left( r^n \frac{\partial \delta \varphi}{\partial r} \right)(t, r) = 0, \tag{9}
\]

\[
\frac{\partial}{\partial r} \delta \varphi(t, R) + \alpha \delta \varphi(t, R) = \alpha \delta U(t)
\]

After simple calculations, it is possible to obtain an expression for the variation of the functional \((6)\), which has the following form:

\[
\delta J = \int_0^R 2r^n \left[ \varphi(t_k, r) - \varphi_g(r) \right] \delta \varphi(t_k, r) dr + 2 \beta \int_0^{t_k} U(t) \delta U(t) dt \tag{10}
\]

Next, we define the adjoint system to the problem (1) - (3):

\[
\frac{\partial \psi(t, r)}{\partial t} = - \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial \psi(t, r)}{\partial r} \right) \tag{11}
\]

With initial and boundary conditions:

\[
\psi(t_k, r) = 2 \left[ \varphi(t_k, r) - \varphi_g(r) \right] \tag{12}
\]

\[
\lim_{r \to 0} r^n k \frac{\partial \psi(t, r)}{\partial r} = 0, \quad \frac{\partial \psi(t, R)}{\partial r} + \alpha \psi(t, R) = 0 \tag{13}
\]

It turns out that the variation of the functional \((10)\) is linearly expressed through the increment variation and the following fact holds.

**3. Statement**

There is equality:

\[
\int_0^R 2r^n \left[ \varphi(t_k, r) - \varphi_g(r) \right] \delta \varphi(t_k, r) dr = akR^n \int_0^{t_k} \psi(t, R) \delta U(t) dt,
\]

Where \(\psi(t, r)\) - adjoint function, which is the solution of problem (11) - (13) in inverse time.

**Evidence.** Perform a chain of calculations:

\[
\int_0^R 2r^n \left[ \varphi(t_k, r) - \varphi_g(r) \right] \delta \varphi(t_k, r) dr = \int_0^R \psi(t_k, r) \delta \varphi(t_k, r) dr = \int_0^R \int_0^{t_k} \frac{\partial}{\partial t} \left( \psi(t, R) \delta \varphi(t, r) \right) dt dr
\]

\[
= \int_0^R \delta \varphi \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial \varphi}{\partial r} \right) + \psi \frac{k}{r^n} \frac{\partial}{\partial r} \left( r^n \frac{\partial \delta \varphi}{\partial r} \right) dt dr = \int_0^{t_k} \left( \int_0^R k r^n \frac{\partial \delta \varphi}{\partial r} \frac{\partial \varphi}{\partial r} dr - \int_0^R k r^n \frac{\partial \varphi}{\partial r} \frac{\partial \delta \varphi}{\partial r} dr \right) dt = \int_0^{t_k} k r^n \frac{\partial \delta \varphi}{\partial r} \psi_{t_k} dt - \int_0^{t_k} k r^n \frac{\partial \psi}{\partial r} \delta \varphi_{t_k} dt = akR^n \int_0^{t_k} \psi(t, R) \delta U(t) dt
\]
Substitute the resulting ratio in the expression for the variation of the functional (10):
\[ \delta f = \int_0^1 \left( akR^n \psi (t, R) + 2 \beta U(t) \right) \delta \theta (t) \, dt \]  
(14)

It is worth noting that many researchers are studying the processes of thermal conductivity. In papers [12, 13], more complex models are considered and the question of the properties of discredited problems is investigated, namely, the questions of convergence of the proposed methods and difference schemes are studied.

4. Numerical Method

We introduce dimensionless variables as follows:
\[ r' = \frac{r}{R}, \quad t' = \frac{kt}{R^2} \]

And define the area \( D \):
\[ D = (0 \leq r \leq 1.0, 0 \leq t \leq t_k) \]

Next, we introduce the grid:
\[ \omega_h = \{ r_i = \frac{i}{N} h, \quad i = 0, 1, \ldots, N \} \]
\[ \omega_t = \{ t_j = \frac{j}{N} \tau, \quad j = 0, 1, \ldots, j_0 \} \]

With steps \( h = \frac{1}{N} n \) and \( \tau = \frac{1}{N} \delta \), Denote \( \varphi_i^1 \) the value in the node \((r_i, t_j)\) of the grid function \( \varphi \) defined on \( D \). Performing the standard replacement of the derivatives \( \frac{\partial \varphi}{\partial t} \) and \( \frac{\partial \varphi}{\partial r} \) with difference expressions, we construct an implicit central difference scheme for Equation 1.

\[ \frac{\partial \varphi}{\partial t} = \frac{\varphi_{i+1}^j - \varphi_i^j}{\tau} \]
\[ \frac{\partial \varphi}{\partial r} = \frac{\varphi_{i+1}^j - \varphi_i^j}{h} \]  
(15)

Applying the operator \( \frac{\partial}{\partial r} \) to \( r^n \frac{\partial \varphi}{\partial r} \), we will have:
\[ \frac{\partial}{\partial r} \left( r^n \frac{\partial \varphi}{\partial r} \right) = \left( \frac{r_i+1 + r_i}{2} \right)^n \frac{\varphi_{i+1}^j - \varphi_i^j}{h^n} - \left( \frac{r_i + r_{i-1}}{2} \right)^n \frac{\varphi_{i+1}^j - \varphi_i^j}{h^n} \]  
(16)

Therefore, the difference equation for determining the desired grid function will be as follows:
\[ \frac{\varphi_i^{j+1} - \varphi_i^j}{\tau} = \frac{1}{h^2} \left( \left( \frac{r_i+1 + r_i}{2} \right)^n \frac{\varphi_{i+1}^j - \varphi_i^j}{h^n} - \left( \frac{r_i + r_{i-1}}{2} \right)^n \frac{\varphi_{i+1}^j - \varphi_i^j}{h^n} \right) \]  
(17)

In the following, for convenience, we will omit the superscript \( (j + 1) \), that is, we assume that \( \varphi_i = \varphi_i^{j+1} \). Imagine the scheme (17) as follows [14, 15]:
\[ A_i \varphi_{i-1} + B_i \varphi_{i+1} - C_i \varphi_i = \frac{\varphi_i^j}{\tau} \]  
(18)

Where:
\[ A_i = \left( \frac{1-1}{h^2} \right)^n; \quad B_i = \left( \frac{1+1}{h^2} \right)^n; \quad C_i = \left[ \left( \frac{1+1}{h^2} \right)^n + \frac{1}{\tau} \right] \]  
(19)

Here: \( i = 1, \ldots, N - 1 \).

Further, it is worth paying attention to the boundary conditions (3) and (4) in differential form. It is known that the order of approximation of the difference problem (17) is two. Therefore, the boundary conditions in the difference form must also have a second order of approximation.

We obtain an approximation of condition (3). To do this, we introduce a fictitious point \( i = N + 1 \). Then condition (3) can be written as follows:
\[ \frac{\varphi_{N+1} - \varphi_N}{2h} + a \varphi_N = aU(t_{j+1}) \]  
(20)

In this case, the point with the number \( N \) becomes an internal point, which means that Equation 18 is fulfilled in it, that is,
\[ A_N \varphi_{N-1} + B_N \varphi_{N+1} - C_N \varphi_N = \frac{\varphi_i^j}{\tau} \]  
(21)
From Equation 20, we can express \( \varphi_{N+1} \) and substitute it into Equation 21. As a result, we will have:

\[
\varphi_{N+1} = \varphi_{N-1} - 2ah\varphi_{N} + 2ahU
\]

Here \( U = U(t_{j+1}) \). Substitute this expression for \( \varphi_{N+1} \) into the Equation 21:

\[
(A_N + B_N)\varphi_{N-1} - (2ahB_N + C_N)\varphi_{N} = \frac{\varphi_{j}}{\tau} - 2ahB_N U
\]

Or:

\[
\varphi_{N-1} = \bar{L}_N\varphi_{N} + \bar{M}_N
\]

(22)

Where:

\[
\bar{L}_N = \frac{(2ahB_N + C_N)}{(A_N + B_N)}
\]

\[
\bar{M}_N = \frac{\varphi_{j} - 2ahB_N U}{(A_N + B_N)}
\]

Thus, Equation 23 is an approximation of condition (3). To solve Equation 18, we will use the standard sweep method. The following fact should be noted. To solve the optimal control problem, it will be necessary to solve the direct problem and the adjoint problem (11)-(13). To do this, it is also possible to use an implicit central difference scheme. The solution of this equation occurs in reverse time, i.e. from \( t = t_{k} \) to 0, while the "-" sign on the right side turns into "+".

Since the control of \( U \) depends on time, we will approximate it with piecewise constant functions of the form [15, 16]:

\[
U(t) = U_i, \quad t_i \leq t \leq t_{i+1}, i = 0,1, ... ,j^0 - 1
\]

(23)

Where \( U_i \) is a constant in the interval equal to the length of the time step. Then the minimized functional becomes a function of \( j^0 \) variables, namely:

\[
J = J(U_1, U_2, ..., U_{j^0})
\]

(24)

And to minimize it, well-known optimization methods, including gradient descent methods, can be used.

It should be noted that in this case, the optimal control problem is reduced to the conditional optimization problem, since the variables \( U_1, U_2, ..., U_{j^0} \) satisfy the constraint (5).

Imagine a variation of the minimizing functional (14) in new dimensionless variables:

\[
\delta J = \int_{0}^{t_{k}} (a\psi(t, 1) + 2\beta U(t)) \delta U(t) dt
\]

(25)

Replacing integration by finite summation, we have:

\[
\delta J = \sum_{t=0}^{N-1} \left( a\frac{\psi(t)+\psi(t+1)}{2} + 2\beta U_j \right) \delta U_j \tau
\]

We get from here that:

\[
\frac{\delta J}{\delta U_j} = \left( a\frac{\psi(t)+\psi(t+1)}{2} + 2\beta U_j \right) \tau
\]

(26)

\[
j = 0,1, ... ,j^0 - 1
\]

Knowing quantities (26), it is easy to write formulas for the gradient descent method

\[
U_j^{(k+1)} = U_j^{(k)} - \gamma \frac{\delta J}{\delta U_j}
\]

\[
U_j = \begin{cases} U^+, & \text{if} \quad U_j^{(k+1)} > U^+ \\ U^-, & \text{if} \quad U^+ \leq U_j^{(k+1)} \leq U^- \\ U^-, & \text{if} \quad U_j^{(k+1)} < U^-
\end{cases}
\]

(27)

Here, \( \gamma \) is the step along the gradient \( (\gamma > 0) \). Figure 3 (a) shows the temperature graph \( \varphi(10\tau, r) \). Figure 3 (b) shows the graph of the solution of the adjoint problem \( \psi(10\tau, r) \).
Consider the algorithm for solving the direct problem. Figure 4 shows the block diagram of this algorithm.

Figure 4. Block diagram of the direct problem solution
At the beginning, the number of points in the partition along the radius \( N \), the maximum temperature \( bT \) and the constants \( \gamma \), \( \alpha \), which are necessary for organizing the calculations of not only the direct system, but also the whole task, are specified. Further, the initial temperature distribution \( f_i(r) \) - which is a function and described in a separate m-file, is specified. Setting these variables in the script looks like this:

```matlab
N=50;
bT=5;
gamma=0.8;
alpha = 0.4;
function [ out ] = fi( r )
    out= 2*r;
end
```

At the next stage, the values of \( h, taw, r, f_i \text{old} \) are calculated, the initial temperature distribution is set at \( t = 0 \). In the script, this stage has the form:

```matlab
h=1/(N-1);
taw=gamma*h*h;
f_i\text{old}=fi(r);
r=0:h:1;
t=0;
l=zeros(1,N);
m=zeros(1,N);
u_mass\_old=zeros(1,1+N);
```

Calculations are organized in an iterative way and implemented in a while loop. The script code directly implements the sweep method, which is described by the formulas (17)-(22):

```matlab
while t<bT
    l(1) = 4/((h^2)*4/(h^2)+1/taw);
    m(1) = fi\_old(1)/(taw*(4/(h^2)+1/taw));

    for i=1:N-1
        den=1/(1/taw + (4*i-(2*i-1)*l(i))/(2*i*h^2));
        l(i+1)=((2*i+1)/(2*i*h^2))*den;
        m(i+1)=(fi\_old(i)/taw + ((2*i-1)*m(i))/(2*i*h^2))*den;
    end

    fi(N) = (alpha*h*u_mass\_old(count)+m(N))/(1-l(N)+h*alpha);
    i=N;

    while i>1
        fi(i-1)=l(i)*fi(i)+m(i);
        i=i-1;
    end

    fi\_old=fi;
    t=t+taw;
    count = count+1;
end
```

The solution of the conjugate problem is determined similarly to the solution of the direct problem. However, we note that the adjoint problem is solved in reverse time. Figure 5 shows the block diagram of the solution of the adjoint problem.
Figure 5. Block diagram of the solution of the conjugate problem

We describe the algorithm for constructing optimal control. The zero values of the arrays grad1 and grad2, which are necessary to determine the parameter \( \alpha \), which is the proportionality coefficient, which connects the controls on the current layer and the previous one, are initialized with zero values. Further, in the external while loop, the main calculation is performed. The condition for exit from the cycle is a small difference of controls.

In the loop body, the \( u_{mass\_old} \) and \( u_{mass} \) values are exchanged to go to the current layer. This is followed by a block that is responsible for solving the direct system. In this block, the method of solving a boundary value problem is directly implemented. After receiving the solution of the direct system, the solution of the adjoint system is searched. At the same time, in the adjoint system, the \( grad2 \) values are determined through the values of \( \psi(N) \) and \( \psi(N-1) \), which is used to determine the values of \( u_{mass} \) via \( u_{mass\_old} \). Before exiting the loop, the obtained values of \( u_{mass} \) are compared with the minimum (\( u_{minus} \)) and maximum (\( u_{plus} \)) values. The process is repeated until \( t < bT \). After passing this cycle, we proceed to the next iteration of the external while loop with the condition \( \text{abs}(u_{mass\_old}(M) - u_{mass}(M)) > \epsilon \). It should be noted that \( u_{mass\_old} \) appears in the direct system, which leads to its change and
approximation to the objective function $f_i\_g (r)$. As a result, we provide the gradient graphics and the constructed optimal control: grad2 and $u\_mass$ (Figure 6).

Figure 6. Block diagram of the algorithm for constructing optimal control
This block diagram formed the basis of the software package for calculating the optimal modes of disk heating. The main algorithmic component of the algorithm for finding optimal control is presented in the script.

```matlab
while abs(u_mass_old(M-1)-u_mass(M-1)) > 0.00001
    u_mass_old=u_mass;
    count = 1;
    fi_old=fu(r);
    while t<bT
        l(1) = 4/(h*h)*(4/(h*h)+1/taw));
        m(1) = fi_old(1)/(taw*(4/(h*h)+1/taw));
        for i=1:N-1
            den=1/(1/taw + (4*i-(2*i-1)*l(i))/(2*i*h^2));
            l(i+1)=((2*i+1)/(2*i*h^2))*den;
            m(i+1)=(fi_old(i)/taw + ((2*i-1)*m(i))/(2*i*h^2))*den;
        end
        fi(N) = (alpha*h*u_mass_old(count)+m(N))/(1-l(N)+h*alpha);
        i=N;
        while i>1
            fi(i-1)=l(i)*fi(i)+m(i);
            i=i-1;
        end
        fi_old=fi;
        t=t+taw;
        count = count+1;
    end
    subplot(1,4,1);
    plot(r,fi_old,'b');
    xlabel('r');
    ylabel('fi_old(r)');
    grid on;
    drawnow;
    l=zeros(1,N);
    m=zeros(1,N);
    t = 0;
    psi_old = (fi_old - fi_g(r));
    subplot(1,4,2);
    plot(r,fi_g(r));
    xlabel('r');
    ylabel('fi_g(r)');
    count2 = 1;
    while t<bT
        l(1) = 4/(h*h)*(4/(h*h)+1/taw));
        m(1) = psi_old(1)/(taw*(4/(h*h)+1/taw));
        for i=1:N-1
            den=1/(1/taw + (4*i-(2*i-1)*l(i))/(2*i*h^2));
            l(i+1)=((2*i+1)/(2*i*h^2))*den;
            m(i+1)=(psi_old(i)/taw + ((2*i-1)*m(i))/(2*i*h^2))*den;
        end
        psi(N) = m(N)/(1-l(N)+alpha*h);
        i=N;
        while i>1
            psi(i-1)=l(i)*psi(i)+m(i);
            i=i-1;
        end
        grad2(count2) = (alpha*(psi(N)+psi_old(N))/2+ 2*betta*u_mass_old(count2))*taw;
        u_mass(count2) = u_mass_old(count2)-(2/(abs(grad1(count2)-grad2(count2))+0.8))*grad2(count2);
        if u_mass(count2) > u_plus
            u_mass(count2) = u_plus;
        elseif u_mass(count2) < u_minus
            u_mass(count2) = u_minus;
        end;
        grad1=grad2;
        psi_old=psi;
    end
```

1404
t=t+taw;
count2 = count2+1;
subplot(1,4,3);
plot(grad2);
xlabel('M');
ylabel('grad');
end
subplot(1,4,4);
plot(u_mass,'b');
xlabel('M');
ylabel('u_mass');
end

Figure 7. Graph $\frac{\partial f}{\partial u_j}$ and $u_j$

5. Conclusion

The paper proposes a method for solving an extremal problem with a quadratic integral functional based on the gradient method for a one-dimensional heat equation with mixed boundary conditions. A formula for the first variation of the integral functional is derived, a numerical algorithm for constructing an approximate solution of a one-dimensional parabolic boundary value problem is implemented. A flowchart of the algorithm and the implementation of this algorithm in a script in the Matlab language are presented. The type of optimal control obtained based on the proposed algorithms is given.

6. Conflicts of Interest

The authors declare no conflict of interest.

7. References


