Numerical research of problems with variable interface

Anastasia Mikhailovna Onishkova
aonishkova@gmail.com

Abstract

A numerical algorithm for solving certain problems in mathematical physics (one- and two-dimensional), are to determine the minimum of a quadratic functional defined in a region containing the previously unknown surfaces. The latter is determined from the minimal functional with unknown functions. We consider the problem for a plane area. Two-dimensional problem is solved by the method of grids. Interface position is defined from a minimum condition. The presented new algorithm uses various methods to search minimum, in particular, genetic algorithms are implemented.

Keywords: algorithm, functional, at least, genetic algorithm, grid, function, area, interface

1 Introduction

Mathematical models of many physical processes and the phenomena lead to the boundary-value problems of mathematical physics containing in advance unknown surfaces, subject to determination during the problem solution. An example of such problem is the model of melting of ice - the so-called Stefan’s problem. Since Gibbs works, for the solving of problems with unknown interfaces the variation methods are widely used. The idea of the solution of such problem consists of in determination of the extremal or stationary points of corresponding functional. A feature of this class of problems is that the variation should be viewed not only the unknown function, but also the position of an unknown interface. Thus, the mathematical challenge is to find such

$$u^*, \tilde{\Gamma} : I \left( u^*, \tilde{\Gamma}^* \right) = \min_{u \in H, \Gamma} I(u, \Gamma),$$

where $u$ is some of the functions of a certain space $H$, and $\Gamma$ is the position of an unknown interface.

The mathematical theory of this class of problems in a certain degree of development [8]. However, the numerical study of such problems are encountered considerable difficulties. In this paper, we propose a numerical algorithm for solving problems with unknown interfaces. The idea of the method is as follows.

Assume that we know any provision of $\tilde{\Gamma}$. Then, solving the problem of finding $u \min I(u, \Gamma)$, can be found $\tilde{u}$ corresponding to the $\tilde{\Gamma}$. Substituting $\tilde{u}$ in $I$, we obtain a functional that depends only on $\Gamma$:

$$\tilde{I}(\Gamma) = I(\tilde{u}(\Gamma), \Gamma).$$

Steps of the algorithm can be roughly summarized as follows:

1. Set the initial position of the border: $\Gamma_0$. 
2. A decision is determined $u_0$.
3. Calculate $I_0 = I(u_0, \Gamma_0)$.
4. $\Gamma_n$ shall be verified the following approximation.
5. Find a solution $u_n : \min_u I(u, \Gamma_n)$.
6. Calculate $I_n = I(u_n, \Gamma_n)$.
7. Check the condition of convergence.

The key questions are:

a) Choice of the next iteration $\Gamma$;

b) Choice of conditions for convergence of the minimization of the sequence $I_n$.

2 Problem Statement

In the rectangular area, where given an equation $k \pm \Delta u = f$, where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and Dirichlet boundary conditions, it is necessary to determine the position of the unknown interface $\Gamma$, which are given matching conditions $k_+ \frac{\partial u_+}{\partial n} = k_- \frac{\partial u_-}{\partial n}$.

![Figure 1: Formulating the problem](image)

Interface $\Gamma$ is found by minimizing a functional

$$I = \frac{1}{2} \int_{V_+} k_+ \nabla u_+^2 dV_+ + \frac{1}{2} \int_{V_-} k_- \nabla u_-^2 dV_- - \int_{\Gamma \cup \partial V_-} fudV$$

3 Genetic algorithm

Note that the result we solve the problem of minimum of several variables. For two-dimensional problem the number of variables is twice the number of nodes used to approximate the unknown boundary. Therefore, it is essential to the issue of choosing a method of finding the minimum. Promising class of algorithms are the so-called genetic algorithms. What is a genetic algorithm? Genetic algorithm - a method for solving optimization problems based on natural selection, just as it is in the process of biological evolution. In the genetic algorithm is repeated modification of a family of individual decisions. At each step in the genetic algorithm were selected randomly chosen subjects from the resulting current solution, called the parent and which is used to generate the next generation
child. Through successive generations of selection is "evolution" toward an optimal solution. Genetic algorithm (GA) can be viewed as a type of random search, which is based on mechanisms that resemble natural selection and reproduction.

Unlike existing techniques, the GA starts with a random set of initial solutions, called population. Each element of the population is called a chromosome and represents a solution to a first approximation. Chromosome is a string of characters of some nature, not necessarily binary. Chromosomes evolve through many iterations, called generation (or generations). During each iteration of the chromosome is evaluated using some fitness function. To create the next generation of new chromosomes, called offspring, are formed either by crossover of two chromosomes - the parents of the current population, either by random changes (mutations) in one chromosome. New population is formed by (a) selection according to the match function of some parents and offspring and (b) removal of remaining in order to maintain a constant population size.

Chromosomes with higher fitness function are more likely to be selected (to survive). After several iterations, the algorithm converges to the best chromosome, which is either optimal or near-optimal solution.

![Figure 2: Generalized structure of the genetic algorithm.](image)

Thus, two kinds of operations:
1. Genetic operations: crossover and mutation;
2. Evolutionary operation: a choice.
Algorithm for solving

1. Set $a$, $b$, $h$, $k_+$, $k_-$ and the type of interface (until I considered only the case of straight boundaries).

2. Building a mesh: $n_1$ is number of points on $[0, a]$; $n_2$ is number of points on $[0, b]$; $x_i = (i - 1) \cdot h, i = 1...n_1$; $y_j = (j - 1) \cdot h, j = 1...n_2$.

3. In arrays $x_G$, $y_G$ put the mesh nodes, through which the border. $x_G$ stores the $x$ coordinate interface and $y_G$ is coordinate $y$.

4. Plotted on a grid and the resulting interface.

5. Obtain sets $V_+$ and $V_-$. $V_+$ will be stored in an array $xVP$ - sites of $x$ and $yVP$ are nodes on $y$; $V_-$ will be stored in an array $xVM$ - sites of $x$, and $yVM$ - sites in $y$.

6. Determine the location of the interface, to remember the coordinates of the interface in arrays $gNy$ and $gNx$.

7. Assign the interface unknown constant $a$, the array of unknowns.

8. Looking up, a solution to the $V_+$:
   (a) Write the boundary conditions $up( ; 1) = 0, up(1 ; ) = 0, up(n_2 ; ) = 0$.
   (b) Compute $f(x, y)$ at grid points on the $V_+ - f_{ij}$.
   (c) For internal nodes equates using difference formula $U_+ \delta_{up} = f(x, y)$ takes the form $up_{i+1j} - 2up_{ij} + up_{i-1j} + up_{ij+1} - 2up_{ij} + up_{ij-1} - f_{ij}h^2/k_+ = 0$.

9. Solve the resulting system of equations.

10. Obtain the solution $up$, which depends on $a$, where $a = up(gNy(i), gNx(i))$.

11. Recall condition. If we assume that the boundary - a straight line, then the normal $n = [h \ 0]$. If we denote the set of nodes $V - um$, then we can write the expression for the gradients of the normal (written for the nodes within the mesh)

\[
\begin{align*}
grup &= \left[ (up(gNy(i), gNx(i)) - up(gNy(i), gNx(x(i) - 1)))/(h)(up(gNy(i), gNx(i)) - up(gNy(i) - 1, gNx(i)))/(h) \right] \\
grum &= \left[ (um(gNy(i), gNx(i) + 1) - um(gNy(i), gNx(i)))/(h)(um(gNy(i) + 1, gNx(i)) - um(gNy(i), gNx(i)))/(h) \right]
\end{align*}
\]

The difference equation for the condition has the form

\[
kp*(n(1)*grup(1) + n(2)*grup(2)) - km*(n(1)*grum(1) + n(2)*grum(2)) = 0.
\]
Because \( n = [h 0] \), we have
\[
kp * (up(gNy(i), gNx(i)) - up(gNy(i), gNx(i) - 1)) - km * (um(gNy(i), gNx(i) + 1) - um(gNy(i), gNx(i))) = 0.
\]
For convenience \( gNy(i) = i \), \( gNx(i) = j \). Obtain
\[
kp * up(i, j) - kp * up(i, j - 1) - km * um(i, j + 1) + km * um(i, j) = 0.
\]
It is known that \( up(i, j) = um(i, j) \). Then
\[
kp * up(i, j) - kp * up(i, j - 1) - km * um(i, j + 1) + km * um(i, j) = 0.
\]
\((kp + km) * up(i, j) - kp * up(i, j - 1) - km * um(i, j + 1) = 0
\]
\( um(i, j + 1) = ((kp + km) * up(i, j) - kp * up(i, j - 1)) / km. \)

It turns out that we have expressed through the nodes set up \( V_- \), are in the column that follows the border (marked with blue dots).

![Figure 3: The column that follows the border](image)

For the remaining nodes \( um \):

(a) Write the boundary conditions \( um(:, n_a) = 0 \), \( um(1, :) = 0 \), \( um(n_b, :) = 0 \).

(b) For internal nodes equates using difference formulas. Equation \( k_- \delta um = f(x, y) \) takes the form
\[
um_{i+1,j} - 2um_{ij} + um_{i-1,j} + um_{ij+1} - 2um_{ij} + um_{ij-1} = f_{ij} h^2 / k_-= 0
\]

(c) Compute \( f(x, y) \) at grid \( V_- - f_{ij} \).

12. Obtain a solution \( um \), that depends on \( a = up(gNy(i), gNx(i)) \).

13. Searching of the functional

Term becomes

1. Term \( \frac{1}{2} \int_{V_+} k_+ \nabla u^2 dV_+ \) becomes
\[
\frac{1}{2} \sum m \sum n \int_{x_n}^{x_{n+1}} \int_{y_m}^{y_{m+1}} (ux^2 + uy^2) dx dy = \frac{1}{2} \sum m \sum n (ux^2 + uy^2) (x_{n+1} - x_n) (y_{m+1} - y_m) = sp
\]

There are
\[
x_n, y_m \in V_+, ux = \frac{\partial u}{\partial x} = \frac{u^n_m - u^{n-1}_m}{h}, uy = \frac{\partial u}{\partial y} = \frac{u^n_m - u^{n}_m}{h}, \nabla u^2 = ux^2 + uy^2.
\]
2. Term \(\frac{1}{2} \int_{V_-} k_- \nabla u^2 dV_-\) becomes

\[
\frac{1}{2} \sum_m \sum_n \int_{x_n}^{x_{n+1}} \int_{y_m}^{y_{m+1}} \left( u x^2 + u y^2 \right) dx dy = 
\]

\[
= \frac{1}{2} \sum_m \sum_n \left( u x^2 + u y^2 \right) (x_{n+1} - x_n) (y_{m+1} - y_m) = sm
\]

There are \(x_n, y_m \in V_-\).

3. Term \(\frac{1}{2} \int_{V_+ \cup V_-} f u dV\) becomes \(sop + som\), where

\[
sop = \frac{1}{2} \sum_i \sum_j f_{ij} u_{ij} (x_{j+1} - x_j) (y_{i+1} - y_i), x_j, y_i \in V_+
\]

\[
som = \frac{1}{2} \sum_k \sum_l f_{kl} u_{kl} (x_{k+1} - x_k) (y_{l+1} - y_l), x_k, y_l \in V_- / \Gamma.
\]

4. Functional \(I = sp + sm-so\). This functionality is expressed through \(u_{ij}\), because \(u_{im}\) was also expressed by \(u_{ij}\). The condition that the derivatives on the boundary with coefficients \(k_p\) and \(k_m\) is also taken into account.

5. Run for the functional genetic algorithm.

5 Test cases

As a test case the results of solving several model problems.

**Example 1**

We consider the solution of the model problem for elliptic equations

\[
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2y + 2y^2 - 2x + 2x^2
\]

In the rectangle centered at the origin, unit height and width equal to one. On the sides of the rectangle placed homogeneous Dirichlet conditions. Exact solution is known

\[
u(x, y) = xy - x^2 y - y^2 x + x^2 y^2.
\]

Decision on the interface \((i = 1)\) 0.03197 Exact solution \(minFI = 0.0333\). Thus, the error can also be considered good. Now take a step less than \(h = 0.1\), thus the number of nodes increases.

**Example 2**

We consider the solution of the model problem for elliptic equations [21]

\[
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -5\pi \sin \pi x \cdot \sin 2\pi y
\]

In the rectangle centered at the origin, unit height and width equal to two. On the sides of the rectangle placed homogeneous Dirichlet conditions. Exact solution is known.

\[
u(x, y) = \sin \pi x \cdot \sin 2\pi y.
\]
6 Conclusions

For two-dimensional problems with free (pre-unknown) boundary developed numerical algorithm for finding its position, which is determined from the minimality of the corresponding functional.
Numerical research of problems with variable interface

Figure 10: The multitudes

Figure 11: The multitudes

Figure 12: The functional

References


