

Identification of the heat transfer coefficient in phase change problems

DAMIAN SŁOTA*

*Institute of Mathematics, Silesian University of Technology, Kaszubska 23,
44-100 Gliwice, Poland*

Abstract In this paper, an algorithm will be presented that enables solving the two-phase inverse Stefan problem, where the additional information consists of temperature measurements in selected points of the solid phase. The problem consists in the reconstruction of the function describing the heat transfer coefficient, so that the temperature in the given points of the solid phase would differ as little as possible from the predefined values. The featured examples of calculations show a very good approximation of the exact solution and stability of the algorithm.

Keywords: Inverse Stefan problem; Solidification

Nomenclature

- b – length of domain, m
- c – specific heat, J/(kg K)
- d – parameter in mutation operator
- D – domain of the problem
- D_l – subset of domain D
- e – relative percentage error, %
- J – functional
- L – heat of solidification, J/kg

*E-mail address: d.slota@polsl.pl

n_{pop}	–	population size
M	–	number of control points
N	–	number of generations
N_1	–	number of sensors
N_2	–	number of measurements from each sensor
p_c	–	crossover probability
p_m	–	mutation probability
r	–	random number
t	–	time, s
t_k	–	time of end of solidification, s
t^*	–	time of end of problem, s
T	–	temperature, K
T_0	–	initial temperature, K
T_∞	–	surrounding temperature, K
T^*	–	temperature of melting point, K
U	–	temperature measurements, K
V, V_α	–	sets of functions
x, z	–	space variable, m

Greek symbols

α	–	heat transfer coefficient, W/(m ² K)
γ	–	regularization parameter
Γ_{ij}	–	boundary of domain
Γ_g	–	phase change moving interface (freezing front)
λ	–	thermal conductivity, W/(m K)
ρ	–	mass density, kg/m ³
σ	–	standard deviation
σ^p	–	standard deviation in percentage of the average value
ξ	–	function describing position of the freezing front
τ	–	current generation number

1 Introduction

The two-phase Stefan problem is a mathematical model of solidification of pure metals, where the distribution of temperature in solid and liquid phases is described by a heat conduction equation with initial and boundary conditions. The position of the freezing front (moving interface) is described by Stefan condition and the condition of temperature continuity. The Stefan problem consists in the determination of temperature distribution within a domain and the position of the freezing front when the initial condition, boundary conditions and thermophysical properties of a body are known.

The inverse Stefan problem consists in the determination of the initial condition, boundary conditions or thermophysical properties of a body. Lack of a portion of input information is compensated for with additional

information about the effects of the initial conditions operation. In the inverse Stefan problem, it is most often assumed that the additional information is a partial knowledge of the moving interface position, its velocity in a normal direction or temperature in selected points of a domain. The problems where the additional information is the position of the moving interface tend to be called the design problems.

It is possible to find an exact analytical solution of the inverse Stefan problem only in few simple cases. In other cases we are left with approximate solutions only (see for example [1–9]). In papers [3, 6], the authors used the Adomian decomposition method or variational iteration method for solving a single-phase inverse design Stefan problem. An advantage of those methods was obtaining a solution in the form of continuous functions and absence of the requirement of domain discretization. Their disadvantage was that they could only be applied to solve a single-phase inverse design Stefan problem. In paper [9], a solution of the inverse design Stefan problem is found in a linear combination form of functions satisfying the heat conduction equation. The coefficients of the combination are determined by the least square method to minimize the maximal defect in the initial-boundary data. This method requires that the interface position is known and therefore, it can only be applied in design problems and cannot be used where we only know the temperature measurement in a considered domain. The method described in paper [2] consists in minimization of a functional, whose value is the norm of difference between the given positions of phase-change front and the positions reconstructed based on the selected function describing the convective heat transfer coefficient. To find the functional's minimum, the Nelder-Mead method [10, 11] was used.

In this paper, an algorithm will be presented that enables solving the two-phase inverse Stefan problem, where the additional information consists of temperature measurements in selected points of the solid phase. The problem consists in the reconstruction of the function describing the heat transfer coefficient, so that the temperature in the given points of the solid phase would differ as little as possible from the predefined values. Based on the given information about temperature measurement, a functional was built defining the error of an approximate solution. To find the functional's minimum, a genetic algorithm was used [12–14]. Genetic algorithms, based on mechanisms which rule the living creatures' evolution, are a very useful tool for solving the global optimization problems, including ones with a large number of variable decisions. Recently, genetic algorithms

have found growing applications in solving of direct and inverse problems for different types of partial differential equations [15–19]. The application of a genetic algorithm for the inverse design Stefan problem is considered in papers [20, 21]. The application of genetic algorithms improve the accuracy of obtained results, compared to the results obtained in paper [2] when using the Nelder-Mead method (see also [21]). To solve a direct Stefan problem, the alternating phase truncation method was applied [22, 23].

The inverse Stefan problem belongs to the ill-posed problems, i.e. its solution is unstable due to errors of input data. This means that errors that are small at the beginning may cause huge problems at the end. In order to avoid such behaviour, appropriate stabilizing procedures are applied. The most frequent ones are: the function specification method [24] and the Tikhonov regularization method [25, 26]. In this paper, the Tikhonov regularization method has been used due to the accuracy and stability of the results obtained. To determine the regularization parameter, the discrepancy principle, proposed by Morozov, has been used [25, 26].

2 Governing equations

Let us consider a vertical device for continuous casting, working in an undisturbed cycle, assuming that the cross-section of an ingot is a circle with a radius of b . In addition, let us assume that the cooling conditions, changing in relation to the ingot casting direction, are identical throughout the ingot circumference. Let us also assume that the heat flux takes place only in the direction perpendicular to the ingot axis. This assumption results from the fact that the amount of heat conducted in the ingot's motion direction, compared to the amount of heat conducted in a direction perpendicular to the ingot axis, is scanty [27, 28]. With such assumptions, as well as due to thermal symmetry of the ingot's domain, the heat exchange process is described by a two-phase Stefan problem, where the time, t , and ingot casting speed, v , are bound by the relation: $t = z/v$, where z is a spatial variable along the ingot's length.

Let the boundary of domain $D = \Omega \times [0, t^*]$ be divided into five components (Fig. 1a):

$$\begin{aligned}
 \Gamma_{11} &= \{(0, t); t \in [0, t_k]\}, & \Gamma_{21} &= \{(b, t); t \in [0, t_p]\}, \\
 \Gamma_{12} &= \{(0, t); t \in [t_k, t^*]\}, & \Gamma_{22} &= \{(b, t); t \in [t_p, t^*]\}, \\
 \Gamma_0 &= \{(x, 0); x \in [0, b]\},
 \end{aligned}$$

for which an initial condition and boundary conditions are predefined. Let D_1 denote the subset of domain D , which is occupied by a liquid phase, and let D_2 denote the domain occupied by a solid phase. The freezing front will be denoted as Γ_g . Let us assume that it is described by function $x = \xi(t)$.

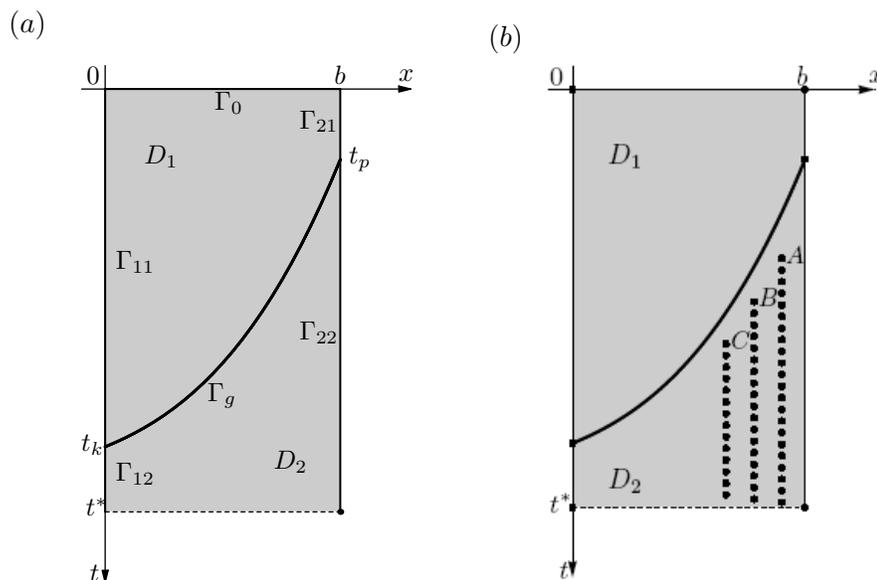


Figure 1. Domain of the two-phase problem (a) and positions of measurement points (b).

With the known values of temperatures in selected points of the solid phase $((x_i, t_j) \in D_2)$:

$$T_2(x_i, t_j) = U_{ij}, \quad i = 1, 2, \dots, N_1, \quad j = 1, 2, \dots, N_2, \quad (1)$$

where N_1 denotes the number of sensors and N_2 denotes the number of measurements from each sensor, function $\alpha(t)$ defined on boundaries Γ_{2k} ($k = 1, 2$) is to be determined, and function $\xi(t)$ describing the freezing front position and the distribution of temperatures T_k in domains D_k ($k = 1, 2$), which inside domains D_k (for $k = 1, 2$) fulfil the heat conduction equation:

$$c_k \rho_k \frac{\partial T_k}{\partial t}(x, t) = \frac{1}{x} \frac{\partial}{\partial x} \left(\lambda_k x \frac{\partial T_k}{\partial x}(x, t) \right), \quad (2)$$

on boundary Γ_0 , they fulfil the initial condition ($T_0 > T^*$):

$$T_1(x, 0) = T_0, \quad (3)$$

on boundaries Γ_{1k} ($k = 1, 2$), they fulfil the homogeneous second kind boundary conditions:

$$\frac{\partial T_k}{\partial x}(x, t) = 0, \quad (4)$$

on boundaries Γ_{2k} ($k = 1, 2$), they fulfil the third kind boundary conditions:

$$-\lambda_k \frac{\partial T_k}{\partial x}(x, t) = \alpha(t) (T_k(x, t) - T_\infty), \quad (5)$$

whereas on the freezing front Γ_g , they obey the temperature continuity condition and the Stefan condition:

$$T_1(\xi(t), t) = T_2(\xi(t), t) = T^*, \quad (6)$$

$$L \varrho_2 \frac{d\xi(t)}{dt} = -\lambda_1 \frac{\partial T_1(x, t)}{\partial x} \Big|_{x=\xi(t)} + \lambda_2 \frac{\partial T_2(x, t)}{\partial x} \Big|_{x=\xi(t)}, \quad (7)$$

where c_k , ϱ_k and λ_k are the specific heat, mass density and thermal conductivity in the liquid phase ($k = 1$) and solid phase ($k = 2$), respectively, α is the heat transfer coefficient, T_0 is initial temperature, T_∞ is the surrounding temperature, T^* is temperature of melting point, L is heat of solidification, and t and x refer to time and spatial location, respectively.

The direct Stefan problem resulting from equations (2)–(7) for a given heat transfer coefficient was solved via the alternating phase truncation method [22, 23]. As a result, the temperature distribution in the solid phase was obtained, constituting the reference point for a comparison of results. From the distribution, temperatures U_{ij} , simulating the temperature measurements, are obtained. Further in the paper, the so obtained temperatures will be treated as accurate.

Function $\alpha(t)$, describing the heat transfer coefficient, will be sought in the form of a function dependent (in a linear or non-linear way) on n parameters:

$$\alpha(t) = \alpha(t; \alpha_1, \alpha_2, \dots, \alpha_n). \quad (8)$$

Let V denotes a set of all functions in the form of (8), where $\alpha_i \in [\alpha_i^l, \alpha_i^u]$ for $i = 1, \dots, n$. For the determined function $\alpha(t) \in V_\alpha$, the problem (2)–(7) becomes a direct Stefan problem, the solution of which allows finding the courses of temperatures $T_{ij} = T_2(x_i, t_j)$ corresponding to function $\alpha(t)$. By taking advantage of the calculated temperatures T_{ij} and the given temperatures U_{ij} , we can build a functional which will determine the error of the approximate solution:

$$J(\alpha(t)) = \|T - U\|^2 + \gamma \|\alpha(t)\|^2, \quad (9)$$

where γ is the regularization parameter and

$$\|T - U\|^2 = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (T_{ij} - U_{ij})^2 \quad \text{and} \quad \|\alpha(t)\|^2 = \int_0^{t^*} (\alpha(t))^2 dt.$$

To determine the regularization parameter, the discrepancy principle proposed by Morozov was used, according to which the regularization parameter is determined from the equality:

$$\|T - U\| = \delta, \quad (10)$$

where δ is the error estimation of the input data U . In practice, for a selected set of values γ_j , $j = 0, 1, \dots, n$ of the regularization parameter, there is element $u_{\gamma_j}^\delta$ minimizing the Tikhonov functional (9). Next, as the sought regularization parameter value, such value of γ_{j_0} is selected, for which equation (10) is satisfied with the required accuracy.

3 Genetic algorithm

For minimization of the Tikhonov functional (9), genetic algorithms were used. These algorithms, based on mechanisms which rule the living creatures' evolution, are a very useful tool for finding the extremes of the function [12–14]:

$$f : D \rightarrow \mathbb{R}, \quad D \subset \mathbb{R}^n. \quad (11)$$

Function f is also called an objective function and the set D , a feasible set. Genetic algorithms process a set of chains of a determined length, called the chromosomes. A single element of the chain is called the gene. In a classic algorithm, genes take on one of the values: zero or one. This is the so-called binary encoding. In this case, the length of the chromosome depends on the size of the domain D and on the accuracy to which we want to find the solution. Also, floating-point encoding (real number representation) is applied, where chromosomes are chains with as many components as the number of variables in the objective function, with each gene assuming real values from a relevant interval. In this case, the length of the chromosome depends on the number of variables of the objective function only, and does not depend on the accuracy to which we want to find the solution. The accuracy of the solution only depends on the accuracy of the floating-point representation used by the applied compiler. In the case of binary encoding, the chromosome is longer (except where decision variables may assume two values

at the most) than in the case of floating-point encoding, which results in the algorithm performing slower [13]. A set of all processed chromosomes is called a population. At the beginning of execution of algorithm, the initial population is created at random, based on a feasible set D (domain of function f). Next, for each chromosome, a fitness function value is calculated. This value reflects how good an approximation of the sought solution is the decision variables vector (i.e. the vector composed of function f arguments) represented by a given chromosome. The fitness function can be equal to the objective function (11) or it may depend on the objective function in a different way [13, 14]. Based on the fitness function value, selection of chromosomes for further processing takes place. The selection consists of selecting a set of chromosomes from the actual population. A set of a size identical to that of the initial population is selected (some chromosomes may be selected many times). During the selection process, each chromosome is first assigned a probability of being selected to the new population and next, based on such probability, a new population is selected (at random).

Later on, the sampled chromosomes are subjected to crossover and mutation operators. These operators correspond to the processes taking place during the evolution of living creatures: crossover of parents' chromosomes and mutation of genes. The purpose of the crossover is an exchange of part of the genetic material between the chromosomes, which may lead to obtaining a new chromosome with a better fitness function value. The purpose of the mutation is a change of the value of a single gen, meaning an introduction of a new "genetic material" to the population, which material could not be obtained via using a crossover operator alone. In a genetic algorithm, two numbers from the $[0, 1]$ interval are determined, the so-called crossover probability, p_c , and the mutation probability, p_m . These numbers reflect the probability of applying a crossover or mutation operator to chromosomes or their genes. A genetic algorithm may also adopt an elitist model, consisting in the protection of the best chromosomes from the previous population. As a result of such operations, a new population is created, which is then subjected to the same transformations. The performance of an algorithm is stopped at the moment a suitable criterion has been fulfilled (e.g. an appropriate function value or number of populations has been achieved, etc.).

There are lots of different selection methods and crossover or mutation operators described in the literature [12–14]. In the genetic algorithm applied in the calculations, for the representation of the decision variables

vector, a chromosome was used in the form of a vector of real numbers (real number representation). A tournament selection and the elitist model were applied in the algorithm. This selection is carried out so that two chromosomes are drawn and the one with better fitness goes to a new generation. There are as many draws as individuals that the new generation is supposed to include. In the elitist model, the best individual of the previous generation is saved and, if all individuals in the current generation are worse, the worst of them is replaced with the saved best individual from the previous population.

As the crossover operator, arithmetical crossover was applied, where as a result of crossing of two chromosomes $\alpha^1 = (\alpha_1^1, \alpha_2^1, \dots, \alpha_n^1)$ and $\alpha^2 = (\alpha_1^2, \alpha_2^2, \dots, \alpha_n^2)$, their linear combinations are obtained:

$$\alpha^{1'} = r \alpha^1 + (1 - r) \alpha^2, \quad (12)$$

$$\alpha^{2'} = r \alpha^2 + (1 - r) \alpha^1, \quad (13)$$

where parameter r is a random number with a uniform distribution from the domain $[0, 1]$.

In the calculations, a nonuniform mutation operator was used as well. During mutation, the α_i gene from chromosome $\alpha = (\alpha_1, \dots, \alpha_i, \dots, \alpha_n)$ is transformed according to the equation:

$$\alpha'_i = \begin{cases} \alpha_i + \Delta(\tau, \alpha_i^u - \alpha_i), \\ \alpha_i - \Delta(\tau, \alpha_i - \alpha_i^l), \end{cases} \quad (14)$$

and a decision is taken at random which from the above formulas should be applied, where:

$$\Delta(\tau, x) = x \left(1 - r^{(1 - \frac{\tau}{N})d}\right), \quad (15)$$

and r is a random number with a uniform distribution from the domain $[0, 1]$, τ is the current generation number, N is the maximum number of generations and d is a constant parameter (in the calculations, $d = 2$ was assumed). A flow diagram of the applied genetic algorithm is presented in Fig. 2.

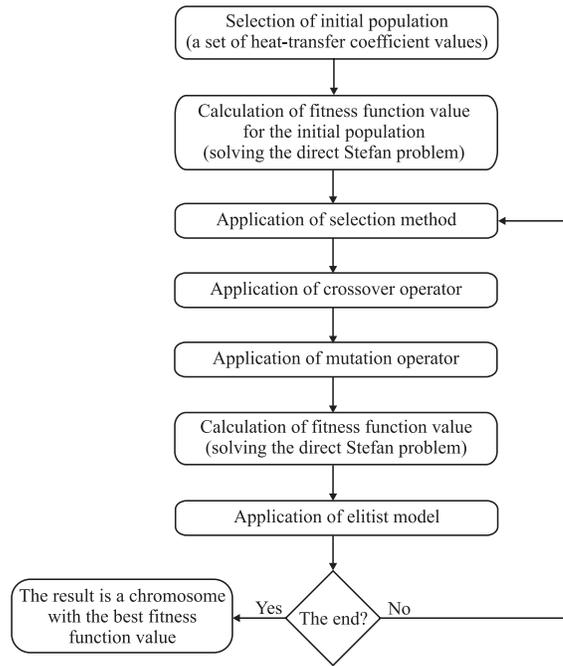


Figure 2. Flowchart of the genetic algorithm.

In calculations parameters used for the genetic algorithm are as follows:

- population size $n_{pop} = 70$,
- number of generations $N = 500$,
- crossover probability $p_c = 0.7$
- mutation probability $p_m = 0.1$.

Calculations were carried out for ten different initial settings of a random numbers' generator. The operators and the values of the genetic algorithm parameters applied in the calculations were selected on the basis of a number of numerical experiments carried out for an design inverse Stefan problem [29–31].

4 Numerical example

Let us present the examples illustrating the exactness and stability of the discussed algorithm. The following parameter values were assumed:

$b = 0.08$ m, $\lambda_1 = 33$ W/(m K), $\lambda_2 = 30$ W/(m K), $c_1 = 800$ J/(kg K), $c_2 = 690$ J/(kg K), $\rho_1 = 7000$ kg/m³, $\rho_2 = 7500$ kg/m³, $L = 270000$ J/kg, $T^* = 1773$ K, $T_\infty = 323$ K and $T_0 = 1813$ K.

In the alternating phase truncation method, the finite difference method was utilized, and the calculations were carried out on a grid with discretization steps equal $\Delta t = 0.1$ and $\Delta x = b/500$. A change of the grid density did not have any significant influence on the results obtained.

Function $\alpha(t)$ was sought in the form:

$$\alpha(t) = \begin{cases} \alpha_1 & \text{for } t \in [0, t_{\alpha_1}], \\ \alpha_2 & \text{for } t \in (t_{\alpha_1}, t_{\alpha_2}], \\ \alpha_3 & \text{for } t \in (t_{\alpha_2}, t^*], \end{cases} \quad (16)$$

where $t_{\alpha_1} = 38$ s, $t_{\alpha_2} = 93$ s. The set of constraints V_α was determined in the following way:

$$V_\alpha = \left\{ \alpha(t); \alpha_1 \in [1000, 1500], \alpha_2 \in [500, 1000], \alpha_3 \in [100, 500] \right\}.$$

The exact values of the sought coefficients α_i were:

$$\alpha_1 = 1200, \quad \alpha_2 = 800, \quad \alpha_3 = 250.$$

Table 1. Number of control points.

Position/Time	1 s	5 s	10 s
<i>A</i>	387	78	39
<i>B</i>	375	75	38
<i>C</i>	361	73	37

It was also assumed in the calculations that there was one thermocouple in the considered domain ($N_1 = 1$) placed, respectively, at the distance of 5 mm (*A*), 10 mm (*B*) and 15 mm (*C*) from the domain boundary (Fig. 1b). The temperature readings were taken every 1 s, 5 s or 10 s. The number of measurements from sensor (N_2 ; also called the number of control points) is presented in Tab. 1. The calculations were based on the exact temperature values and on the values disturbed by a random error with normal distribution equal to 0.5%, 1% and 2%.

Tables 2 and 3 compile the calculation results for the case of placing the sensor in position *A* and, successively, in *C*. The mean (derived from ten start-ups of the genetic algorithm) values of the reconstruction of the heat transfer coefficient, relative percentage error, standard deviation expressed as the weighted average percentage are shown in the tables. Accordingly, in the case of the input data given without perturbation, the heat transfer coefficient is reconstructed very well and with a minimal error, which may be further reduced by increasing the maximal number of generations in the genetic algorithm; however, such procedure is time consuming. Also, the distribution (scatter) (measured by standard deviation) of the obtained results is very small. For the input data burdened with errors, the derived results contain an error that is generally considerably smaller than the input data errors and never exceeds their values. Still, the scatter of the results is not big and only in four cases its percentage value exceeds a half of the input data errors percentage value. The accuracy of reconstruction of the heat transfer coefficient is insignificantly decreased with the sensor moved further from the domain boundary. Such response is consistent with the expectations, as the impact of boundary conditions is smaller at farther distance from the boundary. For the sensor in position *B*, similar results were obtained.

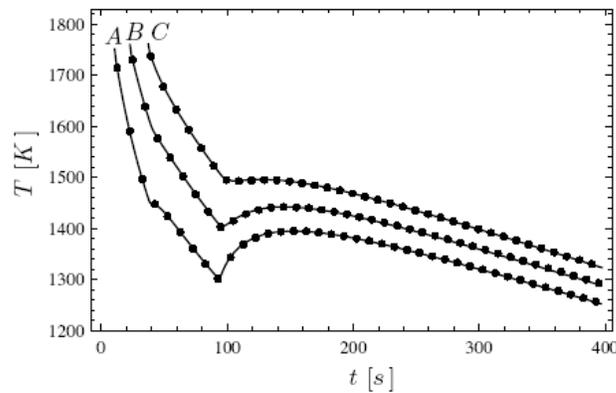


Figure 3. Exact (solid line) and reconstructed (dots) distributions of the temperature in the measurement points for temperature control every ten seconds and for perturbation equal to 2%.

Table 2. Results of the calculations (position *A*, α – reconstructed values of the heat transfer coefficient, e – relative percentage error, σ – standard deviation, σ^p – standard deviations in percent of mean value).

	Per.	α	e [%]	σ	σ^p [%]
1 s	0%	1200.00	0.000073	0.002690	0.000224
		800.00	0.000195	0.005465	0.000683
		250.00	0.000200	0.001969	0.000787
	0.5%	1196.50	0.291535	1.377100	0.115094
		800.39	0.048948	0.595400	0.074389
		250.22	0.089267	0.109240	0.043657
	1%	1199.69	0.025521	1.075164	0.089620
		800.90	0.112969	0.850492	0.106191
		250.16	0.066133	0.135278	0.054075
	2%	1200.29	0.023979	0.221830	0.018481
		802.37	0.296188	0.435958	0.054334
		249.99	0.004000	0.046710	0.018685
5 s	0%	1199.69	0.026056	1.215977	0.101358
		800.26	0.032708	0.765004	0.095594
		249.95	0.018700	0.163492	0.065409
	0.5%	1195.25	0.395910	0.320031	0.026775
		799.78	0.027146	1.194997	0.149415
		250.01	0.003800	0.150159	0.060061
	1%	1194.98	0.418431	0.068284	0.005714
		801.57	0.195760	0.766809	0.095664
		250.05	0.020033	0.200388	0.080139
	2%	1199.30	0.058292	1.646662	0.137302
		801.54	0.192042	1.285823	0.160420
		249.06	0.377100	0.185303	0.074402
10 s	0%	1199.69	0.025840	1.065134	0.088784
		800.42	0.052719	0.825494	0.103132
		249.92	0.031800	0.174393	0.069779
	0.5%	1199.61	0.032375	1.194603	0.099582
		799.62	0.046937	0.582495	0.072846
		250.30	0.118933	0.154738	0.061822
	1%	1195.50	0.375063	0.234386	0.019606
		800.39	0.049208	0.438346	0.054766
		250.07	0.029467	0.126111	0.050430
	2%	1195.26	0.394743	0.102938	0.008612
		805.05	0.631583	0.656153	0.081504
		248.83	0.469767	0.163678	0.065780

Table 3. Results of the calculations (position *C*; notations the same as in Tab. 2 above).

	Per.	α	ϵ [%]	σ	σ^P [%]
1 s	0%	1200.60 799.38 250.05	0.049667 0.077875 0.021733	1.736398 1.809117 0.203791	0.144628 0.226316 0.081499
	0.5%	1202.18 796.85 250.28	0.181431 0.393563 0.111200	3.536819 2.903134 0.189426	0.294201 0.364326 0.075686
	1%	1201.91 795.16 250.38	0.159278 0.605167 0.151233	2.787576 2.527743 0.225868	0.231929 0.317892 0.090211
	2%	1199.85 793.03 250.38	0.012799 0.871167 0.153033	0.053556 1.365088 0.219896	0.004464 0.172136 0.087824
5 s	0%	1200.03 799.81 250.03	0.002847 0.023979 0.013933	0.076617 1.466149 0.328218	0.006385 0.183313 0.131269
	0.5%	1199.99 798.93 250.37	0.000382 0.134344 0.149733	0.236845 1.054698 0.136987	0.019737 0.132015 0.054713
	1%	1200.04 793.97 250.87	0.003465 0.753854 0.346633	0.131360 1.024561 0.247203	0.010946 0.129043 0.098540
	2%	1200.93 811.05 247.75	0.077826 1.380698 0.898900	2.534864 2.728191 0.433034	0.211074 0.336379 0.174785
10 s	0%	1200.01 799.97 249.97	0.000917 0.004156 0.010567	0.070804 1.380651 0.224098	0.005900 0.172589 0.089649
	0.5%	1200.07 797.62 250.54	0.005556 0.296969 0.214533	0.099598 1.282471 0.233734	0.008299 0.160786 0.093293
	1%	1199.99 802.37 249.46	0.000813 0.296781 0.217333	0.025862 0.770827 0.156865	0.002155 0.096068 0.062883
	2%	1208.36 800.34 248.84	0.696757 0.043042 0.462900	11.24376 8.100852 0.646168	0.930497 1.012171 0.259669

Figure 3 illustrates the exact and reconstructed temperature distribution at the control points when the temperature was read every ten seconds, with perturbation equal to 2%. Accordingly, the calculations rendered very big consistency between the exact and the reconstructed temperature distribution. The mean percentage error of temperature reconstruction (in comparison with the exact data), in the case of placing the sensor in position *A* was 0.07%. In such case, the maximal percentage error of temperature reconstruction at a single control point was 0.14%, mean absolute error was 0.77 K, whereas the maximal absolute error equalled 1.46 K. For the location of the sensor in position *B* the errors were: 0.14%, 0.3%, 1.54 K and 3.09 K. For the position *C*: 0.05%, 0.57%, 0.6 K and 8.46 K. In each case, the maximal percentage error was considerably smaller than the input data errors, which in the discussed case, equalled to 2%. Nonetheless, the mean errors were very small. The increase of maximal errors occurred when the sensor moved further from the domain boundary. A similar response was also observed in the reconstruction of the freezing front. In this case, the mean percentage errors were: 0.12% when the sensor was in position *A*, 0.31% for position *B* and 0.34% for position *C*. In Fig. 4 the exact and reconstructed position of the freezing front are shown for the calculations involving the sensor in position *C*, temperature control performed every five and ten seconds, and perturbation equal to 2%. In other cases, the temperature distributions and location of the freezing front were reconstructed very well.

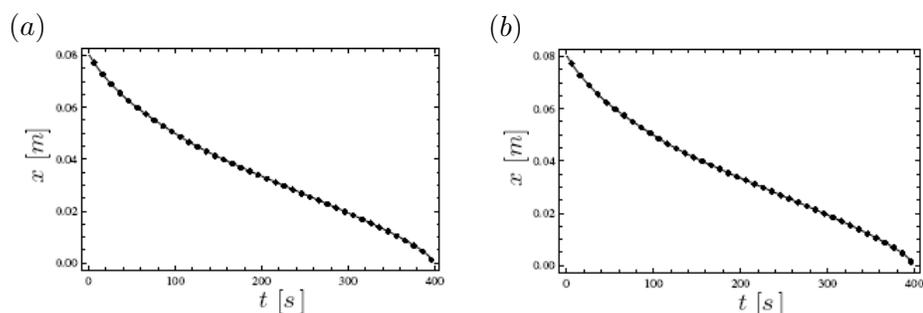


Figure 4. Exact (solid line) and reconstructed (dots) position of the freezing front for calculation for sensor in position *C*, temperature control every five (a) and ten seconds (b) and for perturbation equal to 2%.

5 Conclusions

In this paper, an algorithm has been presented that enables solving the two-phase inverse Stefan problem where the additional information consists of temperature measurements in selected points of the solid phase. The problem consists in the reconstruction of the function describing the heat transfer coefficient, so that the temperature in the given points of the solid phase would differ as little as possible from the predefined values. In numerical calculations, the alternating phase truncation method, the genetic algorithm and the Tikhonov regularization were used.

The featured examples of calculations show a very good approximation of the exact solution and stability of the algorithm in terms of the number of control points and the input data errors. Another important thing is a small scatter of the results obtained during calculations for different initial settings of the pseudorandom numbers' generator.

Acknowledgements The author wishes to thank the reviewers for their valuable criticisms and suggestions, leading to the present improved version of my paper. The research was financed from resources allocated in years 2007–2009 under the research project No. N N512 3348 33.

Received 23 April 2008

References

- [1] ANG D.D., DINH A.P.N., THANH D.N.: *Regularization of an inverse two-phase Stefan problem*, *Nonlinear Anal.* **34** (1998), 719–731.
- [2] GRZYMKOWSKI R., SŁOTA D.: *Numerical method for multi-phase inverse Stefan design problems*, *Arch. Metall. Mater.* **51** (2006), 161–172.
- [3] GRZYMKOWSKI R., SŁOTA D.: *One-phase inverse Stefan problems solved by Adomian decomposition method*, *Comput. Math. Appl.* **51** (2006), 33–40.
- [4] LIU J., GUERRIER B.: *A comparative study of domain embedding methods for regularized solutions of inverse Stefan problems*, *Int. J. Numer. Methods Engrg.* **40** (1997), 3579–3600.
- [5] REN H.-S.: *Application of the heat-balance integral to an inverse Stefan problem*, *Int. J. Therm. Sci.* **46** (2007), 118–127.
- [6] SŁOTA D.: *Direct and inverse one-phase Stefan problem solved by variational iteration method*, *Comput. Math. Appl.* **54** (2007), 1139–1146.
- [7] ZABARAS N.: *Inverse finite element techniques for the analysis of solidification processes*, *Int. J. Numer. Methods Engrg.* **29** (1990), 1569–1587.

- [8] ZABARAS N., KANG S.: *On the solution of an ill-posed design solidification problem using minimization techniques in finite- and infinite-dimensional function space*, Int. J. Numer. Methods Engrg. **36** (1993), 3973–3990.
- [9] GRZYMKOWSKI R., SŁOTA D.: *Multi-phase inverse Stefan problems solved by approximation method*. In: Parallel Processing and Applied Mathematics, R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Waśniewski, (eds.), LNCS, **2328**, Springer, Berlin 2002, 679–686.
- [10] BUNDAY B. D.: *Basic Optimisation Method*, Edward Arnolds Publ., London 1984.
- [11] NELDER J. A., MEAD R.: *A simplex method for function minimization*, The Comp. Journal **7** (1965), 308–313.
- [12] CHAMBERS L.: *The Practical Handbook of Genetic Algorithms, Applications*, Chapman & Hall/CRC, Boca Raton 2001.
- [13] MICHAŁEWICZ Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin 1996.
- [14] OSYCZKA A.: *Evolutionary Algorithms for Single and Multicriteria Design Optimization*, Physica-Verlag, Heidelberg 2002.
- [15] BURCZYŃSKI T., DŁUGOSZ A.: *Evolutionary optimization in thermoelastic problems using the boundary element method*, Comput. Mech. **28** (2002), 317–324.
- [16] DIVO E., KASSAB A., RODRIGUEZ F.: *Characterization of space dependent thermal conductivity with a BEM-based genetic algorithm*, Numer. Heat Transf. A **37** (2000), 845–875.
- [17] KARR CH.L., YAKUSHIN I., NICOLOSI K.: *Solving inverse initial-value, boundary-value problems via genetic algorithm*, Eng. Appl. Artif. Intel. **13** (2000), 625–633.
- [18] MERA N.S., ELLIOTT L., INGHAM D.B.: *A multi-population genetic algorithm approach for solving ill-posed problems*, Comput. Mech. **33** (2004), 254–262.
- [19] WROBEL L.C., MILTIADOU P.: *Genetic algorithms for inverse cathodic protection problems*, Eng. Anal. Bound. Elem. **28** (2004), 267–277.
- [20] SŁOTA D.: *Using genetic algorithms for the determination of an heat transfer coefficient in three-phase inverse Stefan problem*, Int. Comm. Heat & Mass Transf. **35** (2008), 149–156.
- [21] SŁOTA D.: *Solving the inverse Stefan design problem using genetic algorithms*, Inverse Probl. Sci. Eng. **16** (2008), 829–846.
- [22] MAJCHRZAK E., MOCHNACKI B.: *Application of the BEM in the thermal theory of foundry*, Eng. Anal. Bound. Elem. **16** (1995), 99–121.
- [23] ROGERS J. C. W., BERGER A. E., CIMENT M.: *The alternating phase truncation method for numerical solution of a Stefan problem*, SIAM J. Numer. Anal. **16** (1979), 563–587.
- [24] BECK J.V., BLACKWELL B., CLAIR C.R.ST.: *Inverse Heat Conduction. Ill Posed Problems*, Wiley Interscience, New York 1985.
- [25] KURPISZ K., NOWAK A.J.: *Inverse Thermal Problems*, Computational Mech. Publ., Southampton 1995.

-
- [26] TIKHONOV A.N., ARSEIN V.Y.: *Solution of Ill-Posed Problems*, Wiley & Sons, New York 1977.
- [27] LAIT E.: *Mathematical modeling of heat flow in the continuous casting of steel*, Ironmaking and Steelmaking **44** (1973), 589–594.
- [28] MOCHNACKI B, SUCHY J.S.: *Numerical Methods in Computations of Foundry Processes*, PFTA, Kraków 1995.
- [29] SŁOTA D.: *Influence of choice of the crossover operator on the solution of an inverse Stefan problem by genetic algorithms*. In: Artificial Intelligence and Soft Computing, A. Cader, L. Rutkowski, R. Tadeusiewicz, J. Zurada (eds.), EXIT, Warszawa 2006, 217–223.
- [30] SŁOTA D.: *Influence of the mutation operator on the solution of an inverse Stefan problem by genetic algorithms*, w: Computational Science – ICCS 2006, part I, V.N. Alexandrov, G. D. van Albada, P.M.A. Slood, J. Dongarra (eds.), LNCS, **3991**, Springer, Berlin 2006, 786–789.
- [31] SŁOTA D.: *Influence of the selection method on the solution of an inverse Stefan problem using genetic algorithms*. In: Proc. of the 10th IASTED Int. Conf. on Artificial Intelligence and Soft Computing, A.P. del Pobil (ed.), ACTA Press, Anaheim 2006, 285–290.