TWMS J. App. Eng. Math. V.7, N.1, 2017, pp. 33-50

COMBINING GENETIC ALGORITHM AND SINC-GALERKIN METHOD FOR SOLVING AN INVERSE DIFFUSION PROBLEM

HASSAN DANA MAZRAEH[1], REZA POURGHOLI[1], TAHEREH HOULARI[1], §

Abstract. A numerical approach combining the use of a genetic algorithm with the solution of the Sinc-Galerkin method is proposed for the determination of an unknown time-dependent diffusivity $a(t)$ in an inverse diffusion problem (IDP). At the beginning of the numerical algorithm, Sinc-Galerkin method is employed to solve the direct diffusion problem. The present approach is to rearrange the matrix forms of the governing equations. Then, the genetic algorithm is adopted to find the solution of IDP. The genetic algorithm used in this work is not a classical genetic algorithm. Instead, the application of the genetic algorithm to this discrete-time optimal control problem is called a real-valued genetic algorithm(RVGA). Some numerical experiments confirm the utility of this algorithm as the results are in good agreement with the exact data. Results show that a reasonable estimation can be obtained by combining the genetic algorithm and Sinc-Galerkin method within a CPU with clock speed 2.7 GHz.

Keywords: Inverse diffusion problem, genetic algorithm, Sinc-Galerkin method.

AMS Subject Classification: 65M32, 35K05.

1. Introduction

Solution of an inverse diffusion problem requires to determine an unknown diffusion coefficient from an additional information. These new data are usually given by adding small random errors to the exact values from the solution to the direct problem. Inverse diffusion problems appear in many important scientific and technological fields [1–9]. Hence analysis, design, implementation and testing of inverse algorithms are also great scientific and technological interests. In general, inverse problems are ill-posed, that is, their solution does not satisfy the general requirement of existence, uniqueness, and stability under small changes to the input data. To overcome such difficulties, a variety of techniques for solving inverse diffusion problems have been proposed. Therefore, many researchers have focused on the design of inverse algorithms to solve such problems [10–18].

Existing methods try to find an unknown parameter which solves the relevant diffusion problem. So, we can consider an unknown parameter as a vector that approximates an unknown parameter. When we talk about finding a vector to solve a problem, we can use search methods. One of the most powerful search method is Genetic Algorithm that primarily developed by Holland. Also, the genetic algorithm is a very efficient tool for

---

[1] School of Mathematics and Computer Sciences, Damghan University, P.O.Box 36715-364, Damghan, Iran.
e-mail: dana@du.ac.ir, pourgholi@du.ac.ir, tahereh_houlari@yahoo.com;

§ Manuscript received: January 03, 2016; accepted: June 01, 2016.
TWMS Journal of Applied and Engineering Mathematics, Vol.7, No.1; © Işık University, Department of Mathematics 2017; all rights reserved.

some classical methods that need an initial vector to solve a problem. In this case, the responsibility of a genetic algorithm is to find the best vector for the classic method. The various genetic algorithms are widely used in science and engineering. Fortunately, the parallel implementation of the genetic algorithm is easy and the parallel execution of this algorithm gives the better estimation of the solutions and better execution time [19]. In this work, to solve the IDP by using the genetic algorithm, the unknown function will be guessed and we don't need the regularization. This will improve the execution time. In recent years, some researches have been done to solve IDP by using the genetic algorithm and the Sinc-Galerkin method [20, 21].

In the most of the above papers, the numerical results are given based on noiseless data [6, 8]. This difficulty is overcome in this paper and the results are computed based on noisy data. Furthermore, to solve the IDP by using the genetic algorithm, unknown time-dependent diffusivity $a(t)$ will be guessed and we don't need the regularization. This will improve the execution time.

The plan of this paper is as follows. In section 2, we formulate an inverse diffusion problem. Section 3 contains four subsections and outlines some of the main properties of sinc functions and sinc method that are necessary for the formulation of the direct diffusion problem. Furthermore, in this Section, we solve direct problem with this method. In Sections 4 and 5, the genetic algorithm is proposed for the determination of an unknown time-dependent diffusivity $a(t)$ in IDP . Finally, Some numerical experiments will be given in section 6.

2. Mathematical formulation

In this section, we consider the following an IDP in the dimensionless form

$$T_t(x,t) = a(t)T_{xx}(x,t), \qquad 0 < x < 1, \quad 0 < t < t_M \tag{2.1a}$$

$$T(x,0) = f(x), \qquad 0 \le x \le 1, \tag{2.1b}$$

$$T(0,t) = p(t), \qquad 0 \le t \le t_M, \tag{2.1c}$$

$$T(1,t) = q(t), \qquad 0 \le t \le t_M, \tag{2.1d}$$

and the overspecified condition

$$T(x_a,t) = s(t), \qquad 0 \le t \le t_M, \tag{2.1e}$$

where $f(x)$, $p(t)$, and $q(t)$ are continuous known functions, and t_M represents the final time of interest for the time evolution of the problem, while the function $a(t)$ is unknown which remains to be determined from some interior temperature measurements by using the genetic algorithm.

Problem (2.1) can be solved at least-square sense and a cost function can be defined as a summation of squared differences between measured temperatures and calculated values of T by considering guesses functions for $a(t)$:

$$f(Chromosome) = \sum_{j=1}^{m} (T_j - s_j)^2, \tag{2.2}$$

where T_j, $j = 1, 2, 3, ..., m$, are obtained from solving nonlinear direct parabolic problem by considering guess $a(t)$ and applying the Sinc-Galerkin method, and $s_j = s(t_j)$, $j = 1, 2, 3, ..., m$, are the interior temperatures. To find the optimal solution $a(t)$, equation (2.2) should be minimized.

3. Sinc-Galerkin method for solving the direct diffusion problem

In this section, we will review sinc function properties and the sinc method. A comprehensive review concerning sinc function properties as well as sinc method can be found in [22,23].

Let $\mathbb{C}$ denote the set of all complex numbers. The sinc cardinal or sinc function is defined for each $z \in \mathbb{C}$ as follows:

$$sinc(z) \equiv \begin{cases} \frac{sin(\pi z)}{\pi z}, & z \neq 0, \\ 1, & z = 0. \end{cases} \tag{3.1}$$

For $h > 0$ and any integer k, the translated sinc function with evenly spaced nodes is denoted as $S(j,h)(z)$ and defined by

$$S(j,h)(z) \equiv sinc(\frac{z - jh}{h}),\ j = 0, \pm 1, \pm 2, \ldots . \tag{3.2}$$

The sinc functions are cardinal for the interpolating points $z_k = kh$ in the sense that

$$S(j,h)(kh) = \delta_{jk}^{(0)} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases} \tag{3.3}$$

If f is a function defined on the real line $\mathbb{R}$ then the cardinal function of f, denoted as $C(f,h)(x)$, is as follows:

$$C(f,h)(x) \equiv \sum_{j=-\infty}^{\infty} f(jh) S(j,h)(x). \tag{3.4}$$

Whenever the series in (3.4) converges, the cardinal function interpolates f at the points $\{nh\}_{n=-\infty}^{\infty}$. The series was addressed in [24] and analyzed in details in [25].

The truncated cardinal series, denoted as $C_{M,N}(f,h)(x)$, is defined by

$$C_{M,N}(f,h)(x) \equiv \sum_{j=-M}^{N} f(jh) S(j,h)(x). \tag{3.5}$$

We will now introduce two conformal mappings to transform the eye-shaped and wedge-shaped domains to an infinite strip domain.To do this, we define the function

$$\nu = \Phi(z) = ln(\frac{z}{1-z}).$$

This function Φ provides a conformal transformation of the "eye-shaped" spatial domain in the z-plane

$$D_E = \{z \in \mathbb{C} : |arg(\frac{z}{1-z})| < d\},$$

onto the infinite strip

$$D_s = \{w = u + iv : |v| < d \leq \frac{\pi}{2}\},$$

in the w-plane. This is shown in figure 1

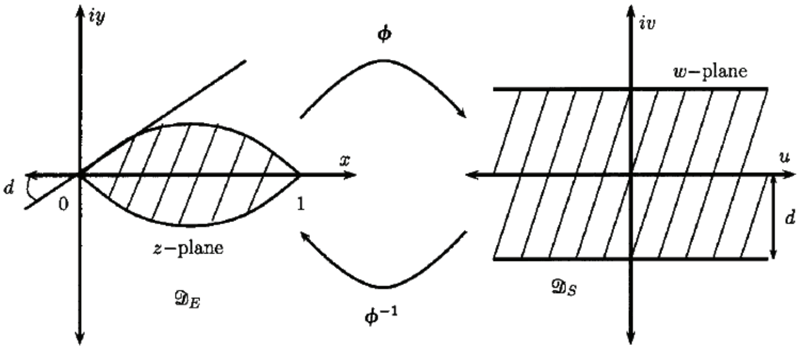

FIGURE 1. Relationship between the domains D_E and D_s

We define the translated Sinc basis functions

$$S_i(z) = S(i,h) \circ \Phi(z) \equiv sinc(\frac{\Phi(z) - ih}{h}). \tag{3.6}$$

For the temporal space, we define the function $\Upsilon(t) = ln(t)$ which is a conformal mapping from D_w the "wedge-shaped" temporal domain onto D_s, the infinite strip, where:

$$D_w = \{t = r + is : |arg(t)| < d \leq \frac{\pi}{2}\},$$

this is shown in figure 2.

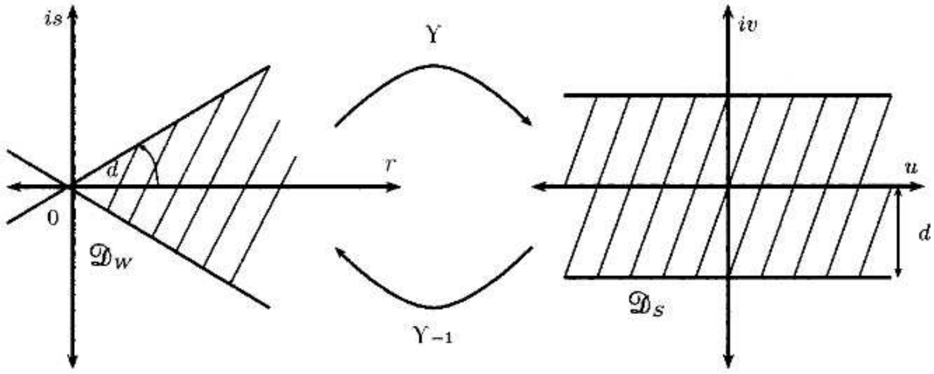

FIGURE 2. Relationship between the domains D_w and D_s

The basic functions are derived from the composite translated Sinc functions,

$$S(j,h) \circ \Upsilon(t) \equiv sinc(\frac{\Upsilon(t) - jh}{h}). \tag{3.7}$$

for $t \in D_w$.
The function $z = \Phi^{-1}(v) = \frac{e^v}{1+e^v}$ is an inverse mapping of $v = \Phi(z)$.
We define the range of Φ^{-1} on the real line as

$$\Gamma = \{\psi(u) = \Phi^{-1}(u) \in D_E : -\infty < u < \infty\}.$$

3.1. **Interpolation and quadrature rules for approximation:**

For problems on a subinterval Γ, of the real line, we employ a conformal map Φ for which $\Phi(\Gamma)=\mathbb{R}$. Suppose $d>0$ and let Φ be a conformal map of the domain D onto D_s. Then over a subinterval $\Gamma=\Phi^{-1}(\mathbb{R})$, we apply the following methods of interpolation [22]

$$f(z)\approx\sum_{k=-\infty}^{\infty}f(kh)S(k,h)\circ\Phi(z). \tag{3.8}$$

and quadrature:

$$\int_{\Gamma}f(z)\approx h\sum_{k=-\infty}^{\infty}f(z_k)/\Phi'(z_k), \tag{3.9}$$

The sinc gride points $z_k\in(0,1)$ in D_E will be denoted by x_i because they are real. For the evenly spaced nodes $\{ih\}_{i=-\infty}^{\infty}$ on the real line, the image which corresponds to these nodes is denoted by

$$x_i=\Phi^{-1}(ih)=\frac{e^{ih}}{1+e^{ih}},\ i=\pm1,\pm2,\ldots$$

and in a same way

$$t_j=\Upsilon^{-1}(jh)=e^{jh},\ j=\pm1,\pm2,\ldots.$$

The Sinc-Galerkin method actually requires the evaluated derivatives of sinc basis functions $S(i,h)\circ\Phi(x)$ at the sinc nodes, $x=x_k$.The rth derivative of $S(i,h)\circ\Phi(x)$, with respect to Φ, evaluated at the nodal point x_k is denoted by

$$\frac{1}{h^r}\delta_{ik}^{(r)}\equiv\frac{d^r}{d\Phi^r}[S(i,h)\circ\Phi(x)]\,|_{x=x_k}. \tag{3.10}$$

Theorem 3.1. *Let Φ be a conformal one-to-one map of the simply connected domain D_E onto D_s then*

$$\delta_{ik}^{(0)}=[S(i,h)\circ\Phi(x)]\,|_{x=x_k}=\begin{cases}1, & k=j;\\ 0, & k\neq j;\end{cases} \tag{3.11}$$

$$\delta_{ik}^{(1)}=h\frac{d}{d\Phi}[S(i,h)\circ\Phi(x)]\,|_{x=x_k}=\begin{cases}0, & k=j;\\ \frac{(-1)^{(k-j)}}{(k-j)}, & k\neq j;\end{cases} \tag{3.12}$$

and

$$\delta_{ik}^{(2)}=h^2\frac{d^2}{d\Phi^2}[S(i,h)\circ\Phi(x)]\,|_{x=x_k}=\begin{cases}\frac{-\pi^2}{3}, & k=j;\\ \frac{-2(-1)^{(k-j)}}{(k-j)^2}, & k\neq j.\end{cases} \tag{3.13}$$

Proof. See [22]. □

The expressions in (3.10) for each i and k can be stored in a matrix $I^{(p)}=[\delta_{ik}^{(p)}]$ for $p=0,1,2$:

$$I^{(0)}=[\delta_{ik}^{(0)}]=\begin{pmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\end{pmatrix}=I, \tag{3.14}$$

$$I^{(1)} = [\delta_{ik}^{(1)}] = \begin{pmatrix} 0 & -1 & \frac{1}{2} & \cdots & \frac{(-1)^{m-1}}{m-1} \\ 1 & & & & \vdots \\ \frac{-1}{2} & & \ddots & & \frac{1}{2} \\ \vdots & & & & -1 \\ \frac{(-1)^m}{m-1} & \cdots & \frac{-1}{2} & 1 & 0 \end{pmatrix}, \tag{3.15}$$

$$I^{(2)} = [\delta_{ik}^{(2)}] = \begin{pmatrix} \frac{-\pi^2}{3} & 2 & \frac{-2}{2^2} & \cdots & \frac{-2(-1)^{m-1}}{(m-1)^2} \\ 2 & & & & \vdots \\ \frac{-2}{2^2} & & \ddots & & \frac{-2}{2^2} \\ \vdots & & & & 2 \\ \frac{-2(-1)^{m-1}}{(m-1)^2} & \cdots & \frac{-2}{2^2} & 2 & \frac{-\pi^2}{3} \end{pmatrix}, \tag{3.16}$$

the above matrices are the $m \times m (m = M + N + 1)$ Toeplitz matrices where

$$-M \leq k \leq N, \qquad -M \leq i \leq N.$$

If function g is evaluated at the sinc nodes $x = x_k$ for $-M_x \leq i \leq N_x$ then the $m_x \times m_x$ square diagonal matrix $D_{m_x}(g)$ is written by

$$D_{m_x}(g) = \begin{pmatrix} g(x_{-M_x}) & & & & \\ & \ddots & & & \\ & & g(x_0) & & \\ & & & \ddots & \\ & & & & g(x_{N_x}) \end{pmatrix}. \tag{3.17}$$

3.2. **Parameter selections for the Sinc-Galerkin method.**

The matrices that comprise the discrete system in the Sinc-Galerkin method are full matrices. More sinc grid points lead to larger matrices and make for an expensive computation. Some cases found in [26] show how to choose an appropriate sinc grid in space and time, and those selections will be used here. If the exact solution satisfies the condition

$$|u(x,t)| \leq C x^{\alpha_s + \frac{1}{2}} (1-x)^{\beta_s + \frac{1}{2}} t^{\gamma_s + \frac{1}{2}} e^{-\delta t}, \tag{3.18}$$

for$(x,t) \in (0,1) \times (0,\infty)$, we should make the following selections

$$N_x = [|\frac{\alpha_s}{\beta_s} M_x + 1|], \; M_t = [|\frac{\alpha_s}{\gamma_s} M_x + 1|], \; N_t = [|\frac{1}{h} ln(\frac{\alpha_s}{\delta} M_x h) + 1|], \tag{3.19}$$

where $[|.|]$ denotes the greatest integer operation, $h \equiv h_x = h_t$ and

$$h = (\frac{\pi d}{\alpha_s M_x})^{\frac{1}{2}}. \tag{3.20}$$

For a given problem with a known real or complex solution, one can determine α, β, γ, and δ using (3.18) where

$$\alpha_s = \alpha - \frac{1}{2} \; and \; \beta_s = \beta - \frac{1}{2}.$$

Then (3.19) and (3.20) provide the computational parameters. In practice, one sets $\alpha = \beta = \gamma = 1$ and $d = \frac{\pi}{2}$. Then from (3.19) and (3.20), $M_x = N_x = N_t$ and $h = \frac{\pi}{2\sqrt{M_x}}$, respectively. Numerical experiments suggest the choice $N_t = \frac{1}{2} M_x$ for the infinite time interval instead of that given in (3.19).To illustrate the performance of the method, we define $\|p_\xi\|$, $\|q_\xi\|$ and $\|E_\xi\|$ for reporting error and convergence results between a true solution $p(x,t)+iq(x,t)$ and a Sinc-Galerkin approximate solution $u_a(x,t)=p_a(x,t)+iq_a(x,t)$ on the sinc grid ξ with $h=h_x=h_t$ as

$$\xi=\{(x_i,t_j): x_i=\frac{e^{ih}}{1+e^{ih}}, t_j=e^{jh}, -M_x-1\le i\le N_x+1, -M_t\le j\le N_t+1\}.$$

3.3. **Direct Problem for the Diffusion Equation.**

The general form of the diffusion equation is as follow:

$$\begin{aligned}
P^{(2)}T(x,t)\equiv T_t(x,t)-a(t)T_{xx}(x,t)&=f(x,t), && 0<x<1,\quad 0<t<\infty, && (3.21a)\\
T(x,0)&=\phi(x), && 0\le x\le 1, && (3.21b)\\
T(0,t)&=p(t), && 0\le t\le\infty, && (3.21c)\\
T(1,t)&=q(t), && 0\le t\le\infty, && (3.21d)
\end{aligned}$$

$\phi(x)$ is a continuous known function, $g(t)$ and $q(t)$ are infinitely differentiable known functions.

We now show the application of the fully Sinc-Galerkin method to solve the direct problem for the diffusion equation. The approximate solution is written as

$$u_{m_x,m_t}(x,t)=\sum_{j=-M_t-1}^{N_t}\sum_{i=-M_x-1}^{N_x+1}u_{ij}\chi_i(x)\theta_j(t), \tag{3.22}$$

where $m_x=M_x+N_x+3$ and $m_t=M_t+N_t+2$. The basis functions $\{s_{ij}(x,t)\}$ for $-M_x-1\le i\le N_x+1, -M_t-1\le j\le N_t$ are given as the product of basis functions for the appropriate one-dimensional problem. They are given by

$$s_{ij}(x,t)\equiv[s(i,h_x)\circ\Phi(x)][s(i,h_t)\circ\Upsilon(t)],$$

where

$$\Phi(x)=ln(\frac{x}{1-x}),\ \Upsilon(t)=ln(t). \tag{3.23}$$

Two linear functions are added to the sinc basis in the spatial dimension

$$\chi_i(x)=\begin{cases}1-x, & i=-M_x-1,\\ s(i,h)\circ\Phi(x), & -M_x\le i\le N_x,\\ x, & i=N_x+1,\end{cases}$$

and one rational function is appended to the temporal base

$$\theta_j(t)=\begin{cases}\frac{t+1}{t^2+1}, & j=-M_t-1,\\ s(j,h)\circ\Upsilon(t), & -M_t\le j\le N_t.\end{cases}$$

Interpolating the boundary and initial conditions in (3.22) dictates that

$$u_{m_x,m_t}(0,t)=\sum_{j=-M_t-1}^{N_t}u_{-M_x-1,j}\theta_j(t)=p(t),$$

$$u_{m_x,m_t}(1,t)=\sum_{j=-M_t-1}^{N_t} u_{N_x+1,j}\theta_j(t)=q(t),$$

$$u_{m_x,m_t}(x,0)=\sum_{i=-M_x-1}^{N_x+1} u_{i,-M_t-1}\chi_i(x)=\phi(x).$$

The sinc approximation to (3.22) is defined by

$$u_{m_x,m_t}(x,t)=\sum_{j=-M_t}^{N_t}\sum_{i=-M_x}^{N_x} u_{ij}s_{ij}(x,t)+g^*(t)\chi_{-M_x-1}(x)+$$

$$q^*(t)\chi_{N_x+1}(x)+\phi(x)\theta_{-M_t-1}(t),$$

where

$$p^*(t)=p(t)-\phi(0)\theta_{-M_t-1}(t),$$

$$q^*(t)=q(t)-\phi(1)\theta_{-M_t-1}(t),$$

and the intervals of i and j confined to $-M_x\le i\le N_x$ and $-M_t\le j\le N_t$, respectively. So $m_x=M_x+N_x+1$ and $m_t=M_t+N_t+1$.

Define the inner product by

$$<\eta,\zeta>\equiv\int_0^{\infty}\int_0^1 \eta(x,t)\zeta(x,t)\nu(x)\omega(t)dxdt,$$

where the product $\nu(x)\omega(t)$ plays the role of a weight function. Assume that the product is given by

$$\nu(x)\omega(t)=\frac{\sqrt{\Upsilon'}}{\Phi'},$$

where

$$\omega(t)=\sqrt{\Upsilon'},\qquad \nu(x)=\frac{1}{\Phi'}.$$

Since $p^*(t)$ and $q^*(t)$ are known functions, the orthogonalization of the residual

$$<P^{(2)}u_{m_x,m_t}-f,s_{kl}>=0,$$

for $-M_x\le k\le N_x$, $-M_t\le l\le N_t$ may be written

$$<P^{(2)}u_h-f^*,s_{kl}>=0, \tag{3.24}$$

where the homogeneous part of the approximate solution is given by

$$u_h(x,t)=\sum_{j=-M_t}^{N_t}\sum_{i=-M_x}^{N_x} u_{ij}s_{ij}(x,t),$$

f^* is also given by

$$f^*(x,t)=f(x,t)-P^{(2)}[p^*(t)\chi_{-M_x-1}(x)+q^*(t)\chi_{N_x+1}(x)+\phi(x)\theta_{-M_t-1}(t)]. \tag{3.25}$$

3.4. **Discrete System Assembly.**

Now we want to discrete the system of (3.24):

$$< u_t, s_{kl} > - < a(t)u_{xx}, s_{kl} > - < f^*, s_{kl} >= 0.$$

The inner product with sinc basis elements is given by

$$< u_t, s_{kl} >= \int_0^\infty \int_0^1 u_t s_{kl} \nu(x)\omega(t)dxdt.$$

This expression contains derivative of u with respect to t. We can remove derivative from the dependent variable u by integrating by parts, once doing this in t. We obtain the following term

$$B_{T_1} - \int_0^\infty \int_0^1 u_h(x,t)[s(k,h_x)\circ\Phi(x)]\nu(x)([s(l,h_t)\circ\Upsilon(t)]\omega(t))'dxdt,$$

where the boundary term

$$B_{T_1} = \int_0^1 [s(k,h_x)\circ\Phi(x)]\nu(x)([s(l,h_t)\circ\Upsilon(t)]\omega(t)u_h(x,t)) \mid_{t=0}^{\infty} dx = 0.$$

If we do the similar calculations for $< a(t)u_{xx}, s_{kl} >$, then we have

$$< a(t)u_{xx}, s_{kl} >= \int_0^\infty \int_0^1 a(t)u_{xx}s_{kl}\nu(x)\omega(t)dxdt.$$

This expression contains the derivatives of the dependant variable u, twice in x. We can similarly remove u_{xx} by integrating by parts, as follows:

$$B_{T_2} - \int_0^\infty \int_0^1 a(t)u_h(x,t)[s(l,h_t)\circ\Upsilon(t)]\omega(t)([s(k,h_x)\circ\Phi(x)]\nu(x))''dxdt,$$

where the boundary term

$$\begin{aligned} B_{T_2} &= \int_0^\infty a(t)[s(l,h_t)\circ\Upsilon(t)]\omega(t)([s(k,h_x)\circ\Phi(x)]\nu(x))'u_h(x,t)) \mid_{x=0}^{1} dt \\ &- \int_0^\infty a(t)[s(l,h_t)\circ\Upsilon(t)]\omega(t)([s(k,h_x)\circ\Phi(x)]\nu(x)u_x(x,t)) \mid_{x=0}^{1} dt \\ &= 0. \end{aligned}$$

Remove the derivatives from the dependent variable u by integrating by parts; twice in x and once in t, to arrive at the identity

$$\int_0^\infty \int_0^1 u_h(x,t)(-\frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2})(s_k(x)s_l(t)\nu(x)\omega(t))dxdt =$$

$$\int_0^\infty \int_0^1 f^* s_k(x)s_l(t)\nu(x)\omega(t)dxdt.$$

We apply the quadrature rule [22] to the iterated integrals and delete the error terms.We also replace $u_h(x,t)$ by u_{ij} and dividing by $h_x h_t$. Hence, we obtain the following discrete sinc system:

$$\frac{a(t_l)\ \omega(t_l)}{\Upsilon'(t_l)} \sum_{-M_x}^{N_x} [-\frac{1}{h_x^2}\delta_{ki}^{(2)}\Phi'(x_i)\nu(x_i) - \frac{1}{h_x}\delta_{ki}^{(1)}(\frac{\Phi''(x_i)\nu(x_i)}{\Phi'(x_i)} + 2\nu'(x_i)) - \delta_{ki}^{(0)}\frac{\nu''(x_i)}{\Phi'(x_i)}]u_{il} \quad (3.26)$$

$$+\frac{\nu(x_k)}{\Phi'(x_k)} \sum_{j=-M_t}^{N_t} [-\frac{1}{h_t}\delta_{lj}^{(1)}\omega(t_j) + \delta_{lj}^{(0)}\frac{\omega'(t_j)}{\Upsilon'(t_j)}]u_{kj} \quad (3.27)$$

$$= \frac{f^*(x_k,t_l)\nu(x_k)\omega(t_l)}{\Phi'(x_k)\Upsilon'(t_l)}. \tag{3.28}$$

This system is identical to the system generated by orthogonalizing the residual via $< p^{(2)}u_h - f^*, s_{kl} >= 0$. We apply the notation of section 2 and obtain the following matrix form

$$\begin{aligned}
&[\frac{-1}{h_x^2} I^{(2)}_{m_x} D(\Phi'\nu) - \frac{1}{h_x} I^{(1)}_{m_x} D(\frac{\Phi''\nu}{\Phi'} + 2\nu') - I^{(0)}_{m_x} D(\frac{\nu''}{\Phi'})] U^{(2)} D(\frac{a\ \omega}{\Upsilon'}) \\
&+ D(\frac{\nu}{\Phi'}) U^{(2)} [\frac{-1}{h_t} I^{(1)}_{m_t} D(\omega) + I^{(0)}_{m_t} D(\frac{\omega'}{\Upsilon'})] \\
&= D(\frac{\nu}{\Phi'}) F^{(2)} D(\frac{\omega}{\Upsilon'}),
\end{aligned}$$

premultiplying by $D(\Phi')$ and postmultiplying by $D(\Upsilon')$ yields the equivalent system

$$\begin{aligned}
&D(\Phi')[\frac{-1}{h_x^2} I^{(2)}_{m_x} D(\Phi'\nu) - \frac{1}{h_x} I^{(1)}_{m_x} D(\frac{\Phi''\nu}{\Phi'} + 2\nu') - I^{(0)}_{m_x} D(\frac{\nu''}{\Phi'})] U^{(2)} D(a\ \omega) \\
&+ D(\nu) U^{(2)} [\frac{-1}{h_t} I^{(1)}_{m_t} D(\omega) + I^{(0)}_{m_t} D(\frac{\omega'}{\Upsilon'})]^T D(\Upsilon') \\
&= D(\nu) F^{(2)} D(\omega).
\end{aligned}$$

It is helpful to single out the portion of the coefficient matrix in this system that corresponds to the second derivative. This is defined by

$$A(v) \equiv \frac{-1}{h_x^2} I^{(2)}_{m_x} - \frac{1}{h_x} I^{(1)}_{m_x} D(\frac{\Phi''}{(\Phi')^2} + \frac{2\nu'}{\Phi'\nu}) - D(\frac{\nu''}{(\Phi')^2\nu}), \tag{3.29}$$

$$B(\sqrt{\Upsilon'}) = \frac{-1}{h_t} I^{(1)}_{m_t} - D(\frac{\omega'}{\omega\Upsilon'}) = \frac{-1}{h_t} I^{(1)}_{m_t} + D(\frac{1}{2}), \tag{3.30}$$

the second equality follows from $\frac{\omega'}{\omega\Upsilon'} = \frac{\Upsilon''}{2(\Upsilon')^2} \equiv \frac{-1}{2}$, where

$$\Upsilon = ln(t).$$

The representation of the system is simplified upon recalling the definition of the matrix $A(\nu)$ and $B(\sqrt{\Upsilon'})$ in (3.29) and (3.30), respectively.

Finally we reach to the form as follow:

$$A_x \nu^{^{(2)}} D(a(t)) + \nu^{^{(2)}} B_t^T = G^{(2)}, \tag{3.31}$$

where

$$\begin{aligned}
A_x &= D(\Phi')A(v)D(\Phi') = D(\Phi')[\frac{-1}{h} I^{(2)} + D(\frac{-1}{(\Phi')^{\frac{3}{2}}}(\frac{1}{\sqrt{\Phi'}})'')]D(\Phi'), \\
B_t &= D((\Upsilon')^{\frac{1}{2}})[-\frac{1}{h} I^{(1)} + D(\frac{1}{2})]D((\Upsilon')^{\frac{1}{2}}), \\
\nu^{^{(2)}} &= D(\nu)U^{(2)} = D((\Phi')^{\frac{-1}{2}})U^{(2)}, \\
G^{(2)} &= D(\nu)F^{(2)} = D((\Phi')^{\frac{-1}{2}})F^{(2)}.
\end{aligned}$$

In the latter, the matrix $F^{(2)}$ is now the matrix of point evaluations of f^* in (3.25). We discretize the system and find it's matrix form and transform obtained equation to follow equation by using theorem A.33 in [22];

$$A\Theta = B,$$

where A is a matrix and B is a vector.

Thus the linear system corresponding to the sinc coefficient u_{ij} can be expressed as

$$A\Theta = B. \tag{3.32}$$

The Matrix A is ill-conditioned. On the other hand, as $g(t)$ is affected by measurement errors, the estimate of Θ by (3.32) will be unstable so that the Tikhonov regularization method must be used to control this measurement errors. The Tikhonov regularized solution ([27], [28], [29], and [30]) to the system of linear algebraic equation (3.32) is given by

$$F_{\alpha}(\Theta) = \|A\Theta - B\|_2^2 + \alpha\|R^{(s)}\Theta\|_2^2.$$

On the case of the $zero_{th}$ -, first-, and second-order Tikhonov regularization method the matrix $R^{(s)}$, for $s = 0, 1, 2$, is given by, see e.g. [31]:

$$R^{(0)} = I_{M_1\times M_1} \in \mathbb{R}^{M_1\times M_1},$$

$$R^{(1)} = \begin{pmatrix} -1 & 1 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(M_1-1)\times M_1},$$

$$R^{(2)} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(M_1-2)\times M_1},$$

where $M_1 = (\gamma + 1) \times (\iota + 1)$.

Therefore, we obtain the Tikhonov regularized solution of the regularized equation as

$$\Theta_\alpha = \left[A^T A + \alpha (R^{(s)})^T R^{(s)}\right]^{-1} A^T B.$$

In our computation, we use the GCV scheme to determine a suitable value of α ([32], [33] and [34]).

Theorem 3.2. *For each fixed t, let $F(x,t) \in B(D_E)$ and $h > 0$. Let Φ and Υ be one-to-one conformal maps of the domains D_E and D_w onto D_s, respectively. Let $x_i = \Phi^{-1}(ih_x), t_j = \Upsilon^{-1}(jh_t)$ and $\Gamma_x = \Phi^{-1}(\mathbb{R}), \Gamma_t = \Upsilon^{-1}(\mathbb{R})$. Assume there are positive constants α_x, β_x and $C_x(t)$ so that*

$$\mid \frac{F(x,t)}{\Phi'(x)} \mid \le C_x(t) \begin{cases} exp(-\alpha_x \mid \Phi(x) \mid, & x \in \Gamma_a^{(x)}, \\ exp(-\beta_x \mid \Phi(x) \mid, & x \in \Gamma_b^{(x)}, \end{cases}$$

where

$$\Gamma_a^{(x)} \equiv \{x \in \Gamma_x : \Phi(x) = u \in (-\infty, 0)\},\ \Gamma_b^{(x)} \equiv \{x \in \Gamma_x : \Phi(x) = u \in (0, \infty)\}.$$

Also for each fixed x, let$F(x,t) \in B(D_w)$and assume there are positive constants α_t, β_t and $C_t(x)$ so that

$$\mid \frac{F(x,t)}{\Upsilon'(x)} \mid \le C_t(x) \begin{cases} exp(-\alpha_t \mid \Upsilon(t) \mid, & x \in \Gamma_a^{(t)}, \\ exp(-\beta_t \mid \Upsilon(t) \mid, & x \in \Gamma_b^{(t)}, \end{cases}$$

where

$$\Gamma_a^{(t)} \equiv \{t \in \Gamma_t : \Upsilon(t) = u \in (-\infty, 0)\},\ \Gamma_b^{(t)} \equiv \{t \in \Gamma_t : \Upsilon(t) = u \in (0, \infty)\}.$$

Then the sinc trapezoidal quadrature rule is

$$\int_{\Gamma_t}\int_{\Gamma_x} F(x,t)dxdt = h_x h_t \sum_{i=-M_x}^{N_x}\sum_{j=-M_t}^{N_t} \frac{F(x,t)}{\Phi'(x_i)\Upsilon'(t_j)} + O(exp(-\alpha_x M_x h_x))$$

$$+O(exp(-\beta_x N_x h_x)) + O(exp(\frac{-2\pi d}{h_x})) + O(exp(-\alpha_t M_t h_t)) + O(exp(-\beta_t N_t h_t))$$

$$+O(exp(\frac{-2\pi d}{h_t})).$$

Hence, make the selections

$$N_x = [| \frac{\alpha_x}{\beta_x}M_x + 1 |], \ M_t = [| \frac{\alpha_x}{\alpha_t}M_x + 1 |], \ N_t = [| \frac{\alpha_x}{\beta_t}M_x + 1 |]$$

where $h \equiv h_x = h_t$ *and*

$$h = \sqrt{\frac{2\pi d}{\alpha_x M_x}}$$

and the exponential order of the sinc trapezoidal quadrature rule is $O(e^{-(\sqrt{2\pi d\alpha_x M_x})^{\frac{1}{2}}})$.

Corollary.*An important special case housed in the previous theorem occurs when the double integrand has the form* $G(x,t)S(p,h_x)\circ\Phi(x)S(q,h_t)\circ\Upsilon(t)$. *Due to the interpolation*

$$S(p,h_x)\circ\Phi(x) = S(p,h_x)(ih_x) = \delta_{ip}^{(0)} \text{ and } S(q,h_t)\circ\Upsilon(t) = S(q,h_t)(jh_t) = \delta_{jq}^{(0)},$$

the sinc quadrature rule is a weighted point evaluation to the order of the method

$$\int_{\Gamma_t}\int_{\Gamma_x} G(x,t)S(p,h_x)\circ\Phi(x)S(q,h_t)\circ\Upsilon(t)dxdt = h_x h_t \frac{G(x_p,t_q)}{\Phi'(x_p)\Upsilon'(t_q)}$$

$$+O(exp(\tfrac{-2\pi d}{h_x})) + O(exp(\tfrac{-2\pi d}{h_t})).$$

Proof. See [26]. □

4. GENETIC ALGORITHM

Genetic algorithms, primarily developed by Holland [35], have been successfully applied to various optimization problems. It is essentially a searching method based on the Darwinian principles of biological evolution. Genetic algorithm is a stochastic optimization algorithm which employs a population of chromosomes; each of them represents a possible solution. By applying genetic operators, each successive incremental improvement in a chromosome becomes the basis for the next generation. The process continues until the desired number of generations has been completed or the predefined fitness value has been reached.

Typically binary coding is used in classic genetic algorithm, where each solution is encoded as a chromosome of binary digits. Each member of the population represents an encoded solution in the classic genetic algorithm. For many problems, this kind of coding is not natural. The genetic algorithm used in this work is not a classic genetic algorithm. Instead, the application of genetic algorithm to this discrete-time optimal control problem is called a real-valued genetic algorithm(RVGA). The continuous function is discrete for numerical computation and simulated by a chromosome. The value of each gene is a real number and indicates the heat generation at each time step [36].

The procedure of a RVGA is as follows:

Step 1. Generate at random an initial population of chromosomes.

Step 2. Evaluate the fitness of each chromosome in the population.
Step 3. Select chromosomes, based on the fitness function, for recombination.
Step 4. Recombine pairs of parents to generate new chromosomes.
Step 5. Mutate the resulting new chromosomes.
Step 6. Evaluate the fitnesses of new chromosomes.
Step 7. Update population.
Step 8. Repeat Step 3 to Step 7, until the fitness function is convergent or less than a predefined value.

5. A MODIFIED RVGA TO DETERMINE $a(t)$

In this paper, we have used a modified RVGA to determine $a(t)$. We guess $a(t)$ by $a(t) = a \times (\frac{b + c \times x + d \times x^2}{e + f \times x + g \times x^2})$ equation. Where a, b, c, d, e, f and g are coefficients, that the modified RVGA must find them. In the modified RVGA, chromosomes are encoded as real-valued vectors. J-th element of each chromosome is j-th coefficient. We consider each element of chromosomes as a gene. That $g_{p,j}$ is j-th gene of chromosome of p. For finding optimal solution of $a(t)$, the Equation (2.2) should be minimum. For this purpose, we consider Equation (2.2) as fitness function and calculate simulated s by solving nonlinear direct heat parabolic problem by sinc-galerkin method for each chromosome. At the end of algorithm, the chromosome by lowest fitness is the best solution of $a(t)$. To improve the performance of RVGA, we added a new step to algorithm after "mutation" operator, for modifying new chromosomes at each iteration.
The procedure of a modified RVGA is as follows:

Step 1. Generate at random an initial population of chromosomes.
Step 2. Evaluate the fitness of each chromosome in the population.
Step 3. Select some chromosomes as parents by tournament selection.
Step 4. For generating pair of new chromosomes, pair of parents crossover together as follow:

$$g_{ch1,j} = \alpha \times g_{p1,j} + (1 - \alpha) \times g_{p2,j}, \qquad j = 1, 2, 3, ..., M,$$

$$g_{ch2,j} = \beta \times g_{p1,j} + (1 - \beta) \times g_{p2,j}, \qquad j = 1, 2, 3, ..., M.$$

Where $p1$ illustrates first parent, $p2$ illustrates second parent, $ch1$ illustrates first new chromosome, $ch2$ illustrates second new chromosome, α and β are random numbers in $[-0.25, 1.25]$.

Step 5. For applying "Mutation" operation on new chromosomes, selecting a gene of each new chromosome randomly and each element of genes adding by random number.
Step 6. Finding the first best gene between new chromosomes and copy that gene to first gene of all chromosomes. Then finding the second best gene between new chromosomes and copy that gene to second gene of all chromosomes. Continue this procedure for all genes. Now all new chromosomes are same. For generating new hopeful chromosomes, genes of second to end chromosomes replace by genes of first chromosome adding by random small values.
Step 7. Evaluate the fitness of new chromosomes.
Step 8. Update the population.
Step 9. Repeat Step 3 to Step 8, until the fitness function is convergent or less than a predefined value.

6. Numerical results

We are going to demonstrate numerically, some of results for the unknown function $a(t)$ in the IDP (2.1). The aim of this section is to illustrate the applicability of the present method described in Section 5 for solving IDP. As expected the IDP is ill-posed and therefore it is necessary to investigate the stability of the present method by giving some test problems.
In this section, we have two examples, for $0 < x < 1, 0 < t < 1$.

Our first example is

Example 6.1.

$$T_t(x,t) = a(t)T_{xx}(x,t), \qquad 0 < x < 1, \quad 0 < t < t_M \tag{6.1a}$$

$$T(x,0) = \sin(\pi x), \qquad 0 \le x \le 1, \tag{6.1b}$$

$$T(0,t) = 0, \qquad 0 \le t \le t_M, \tag{6.1c}$$

$$T(1,t) = e^{-\pi^2 t^2}(\sin(\pi)), \qquad 0 \le t \le t_M, \tag{6.1d}$$

and the overspecified condition

$$s(t_j) = U(0.9, t_j) + \sigma R, \quad j = 1, 2, 3, \cdots, 9, \tag{6.1e}$$

where t_j's are the sinc times nodes, the exact $a(t)$ is $2t$ and the exact $T(x,t)$ is $e^{-\pi^2 t^2}(\sin(\pi x))$.

The second example is

Example 6.2.

$$T_t(x,t) = a(t)T_{xx}(x,t), \qquad 0 < x < 1, \quad 0 < t < t_M \tag{6.2a}$$

$$T(x,0) = \frac{1}{3}e^{-x}, \qquad 0 \le x \le 1, \tag{6.2b}$$

$$T(0,t) = \frac{t^2+1}{3}, \qquad 0 \le t \le t_M, \tag{6.2c}$$

$$T(1,t) = e^{-1}(\frac{t^2+1}{3}), \qquad 0 \le t \le t_M, \tag{6.2d}$$

and the overspecified condition

$$s(t_j) = U(0.9, t_j) + \sigma R, \quad j = 1, 2, 3, \cdots, 9, \tag{6.2e}$$

where, t_j's are the sinc times nodes, the exact value of $a(t)$ is $\frac{2t}{t^2+1}$ and the exact value of $T(x,t)$ is $e^{-x}(\frac{t^2+1}{3})$.

The experimental data $s(t_j)$ (measured temperatures) are obtained from the exact solution of the direct problem by adding a random perturbation error to the exact solution of the direct diffusion problem in order to generate noisy data, where $\sigma = 0.001$ and R is a random value in $(0,1)$.

Remark 6.1. *In an inverse parabolic problem, there are two sources of error in the estimation. The first source is the unavoidable bias deviation (or deterministic error). The* *second source of error is the variance due to the amplification of measurement errors (stochastic error). The global effect of deterministic and stochastic errors is considered in the mean squared error or total error, [12].*

$$S = \left[\frac{1}{(N-1)} \sum_{i=1}^{N} (\widehat{a_i} - a_i)^2\right]^{\frac{1}{2}}, \tag{6.3}$$

where N *is the total number of estimated values,* $\widehat{a_i}$ *is calculated values from guessed* $a(t)$ *and* a_i *is exact values of* $a(t)$.

In our examples, here, a population of 20 chromosomes of 7 genes(a, b, c, $\cdots$, g) is used as the initial guess to obtain for numerical results of modified RVGA. Table 1 presents the results for 1 to 1000 generations for the first example and Table 2 presents the results for 1 to 1000 generations for the second example. Note that S calculated by 10 total number of points.

Gen.	Best fitness	Time(s)	S
1	$4.1270e-02$	14.8147	1.2058
100	$8.1576e-04$	244.9134	0.9575
200	$5.9278e-04$	458.4853	0.8002
300	$4.2176e-04$	702.1418	0.6557
400	$6.0035e-05$	928.8491	0.1548
500	$7.5774e-05$	1149.7431	0.2227
600	$5.8481e-06$	1382.1371	0.0482
700	$7.6551e-06$	1614.9502	0.0646
800	$1.7093e-06$	1817.3170	0.0175
900	$3.1832e-07$	2039.2769	0.0045
1000	$2.4076e-07$	2260.1846	0.0018

Table 1. The results of modified RVGA for a population of 20 chromosomes of 7 genes for 1 to 1000 generations for the first example.

Gen.	Best fitness	Time(s)	S
1	$1.6261e-01$	27.0602	1.6193
100	$7.5126e-03$	364.2550	0.5059
200	$6.9907e-04$	595.8909	0.4445
300	$5.4721e-04$	828.1386	0.4294
400	$2.5750e-04$	1055.8407	0.1789
500	$2.4352e-04$	1288.3499	0.1466
600	$4.7328e-05$	1526.3516	0.0830
700	$5.8526e-06$	1753.5497	0.0757
800	$3.8044e-06$	1986.5678	0.0539
900	$1.2990e-06$	2201.5307	0.0162
1000	$2.0151e-07$	2371.3372	0.0098

Table 2. The results of modified RVGA for a population of 20 chromosomes of 7 genes for 1 to 1000 generations for the second example.

Figure 3 and 4 show the exact and numeric $a(t)$ for the first and the second examples respectively by implementing modified RVGA for 1000 generation.

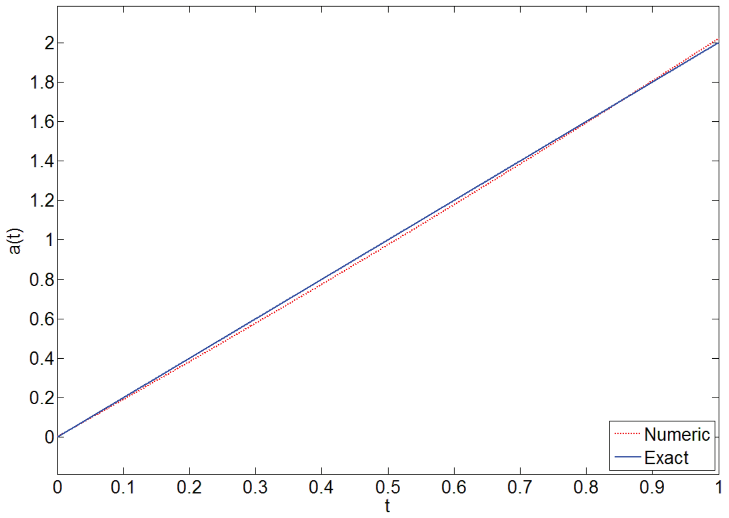

FIGURE 3. The exact and numeric $a(t)$ for the first example by implementing modified RVGA for 1000 generation.

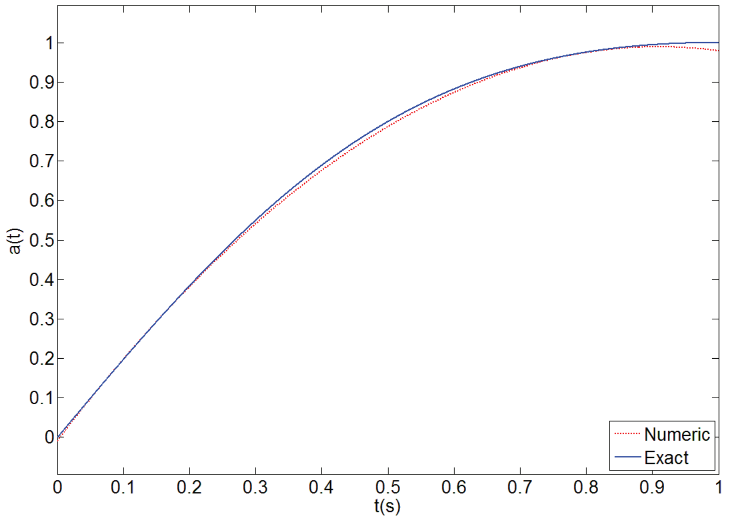

FIGURE 4. The exact and numeric $a(t)$ for the second example by implementing modified RVGA for 1000 generation.

7. CONCLUSION

The present study successfully applies a numerical method to IDP (2.1) and the following results are obtained:

(1) The present study successfully applies the numerical method to the inverse diffusion problem (IDP).
(2) To solve the IDP by using the genetic algorithm, the unknown function will be guessed and we don't need the regularization. This will improve the execution time.
(3) To solve the direct diffusion problem, we used the Sinc-Galerkin method. Therefore, we obtain the solution of direct problem in a more extensive time range. In fact, the solutions are obtained in the whole domain. Furthermore, this method improves the execution time for solving direct diffusion problem.
(4) Results show that a good estimation can be obtained by combining the genetic algorithm and the Sinc-Galerkin method within a CPU with clock speed 2.7 GHz.

References

[1] Cannon,J.R. and Van de Hoek,J., (1982), The one phase stefan problem subject to energy, J. Math. Anal. Appl., 86, pp.281-292.

[2] Cannon,J.R., Eteva,S.P., and Van de Hoek,J., (1987), A Galerkin procedure for the diffusion equation subject to the specification of mass, SIAM J. Numer. Anal., 24, pp.499-515.

[3] Cannon,J.R., (1963), The solution of the heat equation subject to the specification of energy, Quart. Appl. Math., 21, pp.155-160.

[4] Capasso,V. and Kunisch,K., (1988), A reaction-diffusion system arising in modeling manevironment diseases, Quart. Appl. Math., 46, pp.431-450.

[5] Shidfar,A., Pourgholi,R., and Ebrahimi,M., (2006), A numerical method for solving of a nonlinear inverse diffusion problem, Comput. Math. Appl., 52, pp.1021-1030.

[6] Dehghan,M., (2001), An inverse problem of finding a source parameter in a semilinear parabolic equation, Appl. Math. Model., 25, pp.743-754.

[7] Dehghan,M., (2002), Numerical techniques for a parabolic equation subject to an overspecified boundary condition, Appl. Math. Comput. 132, pp.299-313.

[8] Dehghan,M., (2003), Numerical solution of one-dimensional parabolic inverse problem, Appl. Math. Comput., 136, pp.333-344.

[9] Tatari,M. and Dehghan,M., (2007), Identifying a control function in parabolic partial differential equations from overspecified boundary data, Computers and Mathematics with Applications, 53, pp.1933-1942.

[10] Beck,J.V., Blackwell,B., and St.Clair,C.R., (1985), Inverse Heat Conduction: IllPosed Problems, Wiley-Interscience, NewYork.

[11] Beck,J.V. and Murio,D.C., (1986), Combined function specification-regularization procedure for solution of inverse heat condition problem, AIAA J., 24, pp.180-185.

[12] Cabeza,J.M.G., Garcia,J.A.M., and Rodriguez,A.C., (2005), A Sequential Algorithm of Inverse Heat Conduction Problems Using Singular Value Decomposition, International Journal of Thermal Sciences, 44, pp.235-244.

[13] Molhem,H. and Pourgholi,R., (2008), A numerical algorithm for solving a one-dimensional inverse heat conduction problem, Journal of Mathematics and Statistics, 4(1), pp.60-63.

[14] Pourgholi,R., Azizi,N., Gasimov,Y.S., Aliev,F., and Khalafi,H.K., (2009), Removal of Numerical Instability in the Solution of an Inverse Heat Conduction Problem, Communications in Nonlinear Science and Numerical Simulation, 14(6), pp.2664-2669.

[15] Pourgholi,R. and Rostamian,M., (2010), A numerical technique for solving IHCPs using Tikhonov regularization method, Applied Mathematical Modelling, 34(8), pp.2102-2110.

[16] Pourgholi,R., Rostamian,M., and Emamjome,M.,(2010), A numerical method for solving a nonlinear inverse parabolic problem, Inverse Problems in Science and Engineering, 18(8), pp.1151-1164.

[17] Tadi,M., (1997), Inverse Heat Conduction Based on Boundary Measurement, Inverse Problems, 13, pp.1585-1605.

[18] Shidfar,A., Zolfaghari,R., and Damirchi,J., (2009), Application of Sinc-collocation method for solving an inverse problem,Journal of computational and Applied Mathematics, 233, pp.545-554.

[19] Cant-Paz,E., (1995), A summary of research on parallel genetic algorithms, IlliGAL Report No 95007, University of Illinois.

[20] Pourgholi,R., Dana,H., and Tabasi,S.H., (2014), Solving an inverse heat conduction problem using genetic algorithm: Sequential and multi-core parallelization approach, Applied Mathematical Modelling, Volume 38, Issues 78.
[21] Pourgholi,R., Molai,A.A., and Houlari,T., (2013), Resolution of an inverse parabolic problem using Sinc-Galerkin method, TWMS J. App. Eng. Math.,V2(3), pp.160-181.
[22] Lund,J. and Bowers,K., (1991), Sinc Methods for Quadrature and Differential Equations, Siam, Philadelphia, PA.
[23] Stenger,F., (1979), A Sinc-Galerkin method of solution of boundary-value problems, Math. Comp., 33, pp.85-109.
[24] Whittaker,E.T., (1915), On the functions which are represented by the expansions of the interpolation theory, Proc. Roy. Soc. Edinburg, 35, pp.181-194.
[25] Whittaker,J.M., (1935), Interpolation Function Theory, in: Cambridge Tracts in Mathematics and Mathematical Physics, Vol.33, Cambridge University Press, London.
[26] Koonprasert,S. and Bowers,K., (2004), The Fully Sinc-Galerkin Method for Time-Dependent Boundary Conditions, Numerical Method for Partial Differential Equations, 20(4), pp.494-526.
[27] Hansen,P.C., (1992), Analysis of discrete ill-posed problems by means of the L-curve, SIAM Rev., 34, pp.561-80.
[28] Lawson,C.L. and Hanson,R.J., (1995), Solving Least Squares Problems, Philadelphia, PA:SIAM.
[29] Tikhonov,A.N. and Arsenin,V.Y., (1977), On the solution of Ill-posed problems, New York, Wiley.
[30] Tikhonov,A.N., and Arsenin,V.Y., (1977), Solution of Ill-Posed Problems, V.H.Winston and Sons, Washington, DC.
[31] Martin,L., Elliott,L., Heggs,P.J., Ingham,D.B., Lesnic,D., and Wen,X., (2006), Dual Reciprocity Boundary Element Method Solution of the Cauchy Problem for Helmholtz-type Equations with Variable Coefficients, Journal of sound and vibration, 297, pp.89-105.
[32] Elden,L., (1984), A Note on the Computation of the Generalized Cross-validation Function for Ill-conditioned Least Squares Problems, BIT, 24, pp.467-472
[33] Golub,G.H., Heath,M., and Wahba,G., (1979), Generalized Cross-validation as a Method for Choosing a Good Ridge Parameter, Technometrics, 21(2), pp.215-223.
[34] Wahba,G., (1990), Spline Models for Observational Data, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol.59, SIAM, Philadelphia.
[35] Holland,J.H., (1975), Adaptation in Natural and Artificial System, University of Michigan Press, Ann Arbor.
[36] Liu,F-B., (2008), A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source, International Journal of Heat and Mass Transfer, 51, pp.3745-3752.

Reza Pourgholi, for the photograph and short biography, see TWMS J. Appl. and Eng. Math., V.2, No.2, 2012.

Hassan Dana Mazraeh is a lecturer at Damghan University, School of Mathematics and Computer Science, Iran. His areas of research are evolutionary computing, numerical analysis, and scientific computing. Web address: http://faculty.du.ac.ir/dana/.

Tahereh Houlari for the photograph and short biography, see TWMS J. Appl. and Eng. Math., V.3, No.2, 2013.