Available online at www.sciencedirect.com

ScienceDirect

International Journal of Heat and Mass Transfer 49 (2006) 4732–4750

International Journal of HEAT and MASS TRANSFER

www.elsevier.com/locate/ijhmt

Applying neural networks to the solution of forward and inverse heat conduction problems

S. Deng, Y. Hwang *

Department of Weapon System Engineering, Chung Cheng Institute of Technology, National Defense University, No. 190, Sanyuan 1st St., Dashi Jen, Taoyuan 33509, Taiwan, ROC

Received 27 October 2005; received in revised form 10 June 2006
Available online 14 August 2006

Abstract

This paper employs the continuous-time analogue Hopfield neural network to compute the temperature distribution in forward heat conduction problems and solves inverse heat conduction problems by using a back propagation neural (BPN) network to identify the unknown boundary conditions. The weak generalization capacity of BPN networks is improved by employing the Bayesian regularization algorithm. The feasibility of the proposed method is examined in a series of numerical simulations. The results show that the proposed neural network analysis method successfully solves forward heat conduction problems and is capable of predicting the unknown parameters in inverse problems with an acceptable error.

Keywords: Neural network; Inverse heat conduction problems (IHCPs)

1. Introduction

In forward heat conduction problems the heating characteristics, the boundary conditions and the initial conditions of a body are known and are used to establish the internal temperature field. Conversely, in inverse heat conduction problems (IHCPs), experimental temperature measurements are taken at various points in the interior of a body and are used to estimate the unknown boundary conditions existing at the external surface. IHCPs are mathematically ill-posed in the sense that the existence, uniqueness and stability of their solutions cannot be assured [1]. IHCPs are generally solved using some form of numerical technique. Classical approaches include space marching [2,3] the single future time step method [1, pp. 115–119], the function specification method [4,1, pp. 119–134], the regularization method [1, pp. 134–145] and the trial function method [1, pp. 145–148]. Since the 1970s, computer science and technology have advanced rapidly and hence contemporary researchers generally solve IHCPs using numerical methods such as the finite element method [5–7], the finite different method (FDM) [8], the boundary element method [9,10], sequential method [11], Kalman filter method [12,13] and Genetic algorithm [14].

In examining forward heat conduction problems, this study commences by developing the governing equations for various one- and two-dimensional heat conduction cases and then describes the continuous-time analogue Hopfield neural network (CHNN) scheme. The differential equations of the CHNN are derived and correlated with the governing equations of the conduction problems. The framed Hopfield-type neural network is then applied to solve a number of conventional, and rather more complicated, one- and two-dimensional heat conduction problems. The accuracy of the CHNN solutions is verified via comparison with the exact solutions and the FDM results.

The rapid development of artificial neural network technology in recent years has led to an entirely new approach for the solution of IHCPs [15–18,36]. Neural networks are artificial intelligence systems which mimic the biological

* Corresponding author.
E-mail addresses: sgdeng@ccit.edu.tw (S. Deng), g960405@ccit.edu.tw (Y. Hwang).

0017-9310/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2006.06.009

Nomenclature

BPN	back propagation neural
C_i	amplifier input capacitance of *i*th neuron
CHNN	Continuous-time analogue Hopfield neural network
d_j	desired output of *j*th neuron
I_i	external input to *i*th neuron
L	thermal layer thickness
M, N	total number of spatial nodes
q	heat flux
R_i	resistance of *i*th neuron
t	time (continuous)
t_{f}	final time
T	temperature
T_0	uniform initial temperature
$\widehat{T}_i$	difference between temperature of ith neuron and its neighbor
u_i	internal state of *i*th neuron
$\bar{\mathbf{w}}_1$, $\bar{\mathbf{w}}_2$, $\bar{\mathbf{w}}_3$	sub-matrices of **W**
w_{ij}	connection strength between neurons *j* and *i*
W	coefficient matrix
$\tilde{x}$, $\tilde{y}$	dimensionless axial coordinate
$\tilde{x}_u$, $\tilde{y}_u$	measurement location
y_j	output of *j*th neuron
α	thermal diffusivity
β	steepness of sigmoid function
δt	sampling interval time
ξ_1, ξ_2	object function parameters
θ_k	bias of neural network
κ	thermal conductivity
ρ	total shunt capacitance
$\varphi(\cdot)$	neuron activation function
Ψ	coefficient matrix
Ψ_j	continuous functions of one variable
$\overline{\Psi}_1$, $\overline{\Psi}_2$	sub-matrices of Ψ
o	null matrix

Subscripts

i, j	indices

Superscripts

T	transpose of matrix
$\sim$	dimensionless value

processes of a human brain by using non-linear processing units to simulate the functions of biological neurons. The processing units (nodes) are fully interconnected by joints of invariable strength which mimic the synaptic behavior of the human brain. The units are multi-dimensional, self-organizing, fuzzy and self-learning capabilities. Neural networks can be applied to the solution of IHCPs by training the network with multiple samples of temperature distribution data obtained from forward heat conduction problems and adjusting the weights of the individual nodes such that the actual network outputs closely approximate the target values. The fully trained neural network can then predict an unknown output for any arbitrary input by applying the network weights established during the training stage.

The following sections perform a forward analysis using a neural network. Two inverse heat conduction problems are then considered to confirm the validity of the proposed method. The first problem concerns a one-dimensional cylindrical coordinate system, while the second involves a two-dimensional rectangular coordinate system. In both cases, the CHNN scheme is used to perform the forward heat conduction analysis. The results of the forward analysis are used as training data for a three-layered back propagation neural (BPN) network designed to solve IHCPs with different heat profiles. The BPN network is trained using eight different algorithms and the relative performance of each algorithm is examined in terms of its convergence rate and the accuracy of the final solutions. These algorithms include: Conjugate gradient back propagation with resilient back propagation (CRB) [19], gradient descent with momentum and adaptive learning rate back propagation (GMB) [20], conjugate gradient back propagation with Fletcher–Reeves updates (CBF) [21], scaled conjugate gradient back propagation (SCB) [22], quasi-Newton back propagation (QNB) [21, p. 242], one-step secant back propagation (OSB) [23], conjugate gradient back propagation with Powell–Beale restarts (CBP) [24] and Levenberg–Marquardt back propagation (LMB) [20]. To overcome the weak generalization capacity of general back propagation algorithms when applied to non-linear function approximations, and to take account of the uncertain noise inherent in the current IHCPs, the network is also trained using the Bayesian regularization scheme. The performance of the network trained using the best training algorithm is then compared with that trained using the Bayesian regularization approach.

2. Formulation of forward heat conduction problems

This study applies the CHNN model to solve the temperature distribution field of various forward heat conduction problems expressed in either a one-dimensional Cartesian coordinate system, a cylindrical coordinate system or a two-dimensional rectangular coordinate system.

2.1. One-dimensional Cartesian coordinate system

Initially, this study constructs the homogeneous differential equation for the one-dimensional heat conduction case. It is assumed that the one-dimensional bar (in Fig. 5a top left) or finite plate (in Fig. 5b bottom right) has a thermally insulated surface at $x = L$ and that a time-varying heat flux, $q(t)$, acts at $x = 0$. The boundary and initial conditions are as shown. The corresponding system state equations can be expressed in dimensionless form as

$$\frac{1}{\alpha}\frac{\partial \widetilde{T}}{\partial \tilde{t}} = \frac{\partial^2 \widetilde{T}}{\partial \tilde{x}^2}, \quad 0 \leqslant \tilde{x} \leqslant L,\ \tilde{t} > 0, \tag{1a}$$

$$\widetilde{T}(\tilde{x}, 0) = T_0, \quad 0 \leqslant \tilde{x} \leqslant L, \tag{1b}$$

$$\frac{\partial \widetilde{T}}{\partial \tilde{x}} = -\tilde{q}(\tilde{t}), \quad \tilde{x} = 0,\ \tilde{t} > 0, \tag{1c}$$

$$\partial \widetilde{T}/\partial \tilde{x} = 0, \quad \tilde{x} = L, \tag{1d}$$

where

$$\widetilde{T} = (T - T_0)/(q_0 L/\kappa), \quad \tilde{q} = q/q_0, \quad \tilde{x} = x/L,$$
$$\tilde{t} = \alpha t/L^2, \quad L = 1. \tag{1e}$$

2.2. One-dimensional cylindrical coordinate system

In Fig. 1a, it is assumed that the one-dimensional hollow cylinder has a thermally insulated surface at $r = b$ and that the time-varying heat flux, $q(t)$, acts at the inner surface, $r = a$. The boundary and initial conditions are as shown. The corresponding system state equations can be expressed in dimensionless form as [25]

$$\frac{1}{\alpha}\frac{\partial \widetilde{T}}{\partial \tilde{t}} = \frac{\partial^2 \widetilde{T}}{\partial \tilde{r}^2} + \frac{1}{\tilde{r}}\frac{\partial \widetilde{T}}{\partial \tilde{r}}, \quad a \leqslant \tilde{r} \leqslant b,\ \tilde{t} > 0, \tag{2a}$$

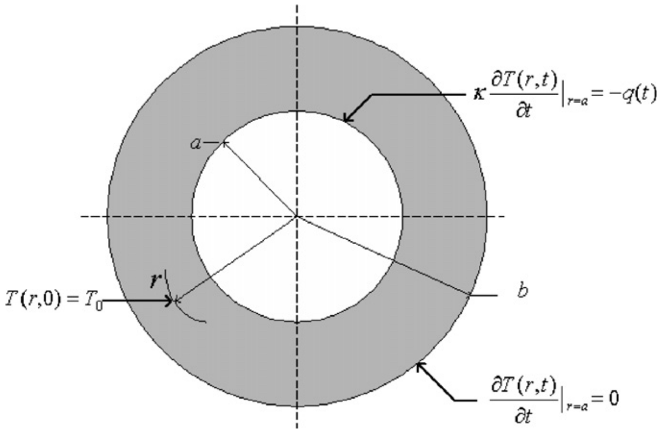
$$\widetilde{T}(\tilde{r}, 0) = T_0, \quad a \leqslant \tilde{r} \leqslant b, \tag{2b}$$

$$\frac{\partial \widetilde{T}}{\partial \tilde{r}} = -\tilde{q}(\tilde{t}), \quad \tilde{r} = a,\ \tilde{t} > 0, \tag{2c}$$

$$\frac{\partial \widetilde{T}}{\partial \tilde{r}} = 0, \quad \tilde{r} = b, \tag{2d}$$

where

$$\widetilde{T} = (T - T_0)/(q_0 L/\kappa), \quad \tilde{q} = q/q_0, \quad \tilde{r} = r/L,$$
$$\tilde{t} = \alpha t/L^2, \quad L = b - a = 1. \tag{2e}$$



$\kappa \frac{\partial T(r,t)}{\partial t}\Big|_{r=a} = -q(t)$
a
r
b
$T(r,0) = T_0$
$\frac{\partial T(r,t)}{\partial t}\Big|_{r=a} = 0$

Fig. 1a. 1-D boundary and initial conditions for hollow cylinder.

In Eqs. (1e) and (2e), T_0 is the uniform initial temperature, α the thermal diffusivity, κ the thermal conductivity, q_0 the nominal value of the surface heat flux and $\tilde{q}(\tilde{t})$ the dimensionless heat flux. Note that for convenience, the superscript ($\sim$) is omitted throughout the remainder of this paper.

2.3. Two-dimensional rectangular coordinate system

This study also establishes the homogeneous differential equation of heat conduction for a two-dimensional rectangular system. In this case, the temperature distribution involves two heat flux inputs, $q_1(t)$ and $q_2(t)$, at the boundaries. Fig. 1b illustrates the boundary and initial conditions of the internal heat conduction for this two-dimensional case. In dimensionless form, the corresponding system of equations is given by [26]

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}, \quad 0 \leqslant x \leqslant x_u,\ 0 \leqslant y \leqslant y_u,\ t > 0, \tag{3a}$$

$$T(x, y, 0) = T_0, \quad 0 \leqslant x \leqslant x_u,\ 0 \leqslant y \leqslant y_u, \tag{3b}$$

$$\frac{\partial T}{\partial x} = -q_1(t), \quad x = 0, \quad 0 \leqslant y \leqslant y_u,\ t > 0, \tag{3c}$$

$$\frac{\partial T}{\partial y} = 0, \quad 0 \leqslant x \leqslant x_u,\ y = 0,\ t > 0, \tag{3d}$$

$$\frac{\partial T}{\partial x} = 0, \quad x = x_u,\ 0 \leqslant y \leqslant y_u,\ t > 0, \tag{3e}$$

$$\frac{\partial T}{\partial y} = -q_2(t), \quad 0 \leqslant x \leqslant x_u,\ y = y_u,\ t > 0, \tag{3f}$$

where T_0 is the uniform initial temperature and $q_1(t)$ and $q_2(t)$ are the two heat flux inputs.

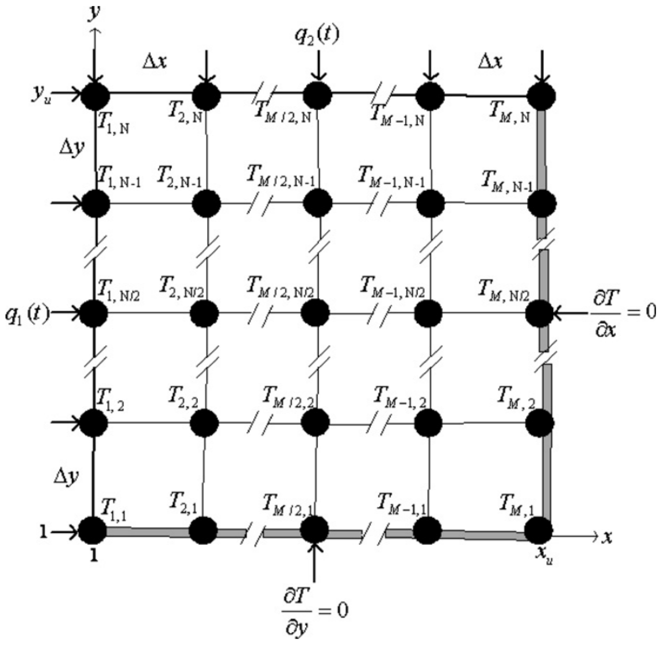


Fig. 1b. 2-D boundary and initial conditions.

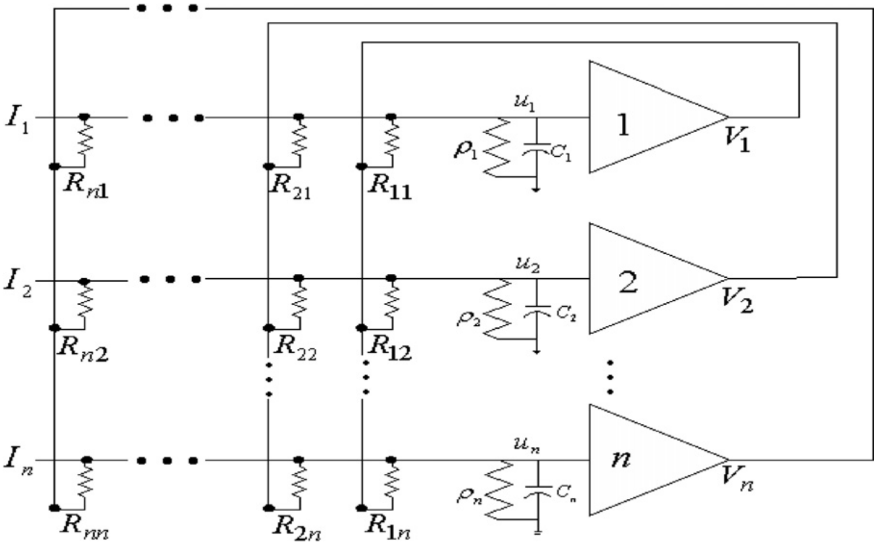


Fig. 2a. Basic Hopfield neural network connectivity circuit diagram.

2.4. Hopfield neural network

The Hopfield continuous-time dynamic neural network constructed from n dynamic neural units can be described by the following non-linear differential equations [27,28]:

$$C_i\frac{\mathrm{d}u_i(t)}{\mathrm{d}t}=\sum_{\substack{j=1\\ i\neq j}}^{n}w_{ij}V_j(t)-\frac{u_i(t)}{R_i}+I_i(t),\quad i=1,2,\ldots,n, \tag{4a}$$

$$y_i(t)=\varphi(u_i(t)),\quad i=1,2,\ldots,n, \tag{4b}$$

where u_i is the internal state of the ith neuron, $1/R_i=\sum_{j=1}^{n}w_{ij}+1/\rho_i$, $1/R_{ij}=w_{ij}$, w_{ij} is the strength of the connection between neurons j and i, y_j is the output of the jth neuron, $\varphi(\cdot)$ is the neuron activation function, I_i is the external input to neuron i, and y_i is the output signal from each neuron, including the ith neuron.

The neuron's output, i.e. $y_i=\varphi(u_i)$, is a non-decreasing function of the activation level. Hopfield network implementations generally employ a sigmoid activation function such as the tanh function or a piecewise linear approximation to a sigmoid. The hyperbolic tangent sigmoid function has the form:

$$\varphi(u_i)=\tanh(\beta u_i)=(\mathrm{e}^{\beta u_i}-\mathrm{e}^{-\beta u_i})(\mathrm{e}^{\beta u_i}+\mathrm{e}^{-\beta u_i})^{-1}. \tag{5}$$

This sigmoid function is rotationally symmetric about the origin of the coordinate axes, and asymptotically approaches limiting points of $(\infty,1)$ and $(-\infty,-1)$. The parameter β in the sigmoid function is adjustable according to the non-linearity of the problem. In this paper, a value of $\beta=1/2$ is used in the simulations [29].

In this study, the numerical solutions to the two-dimensional forward heat conduction problems are obtained using the CHNN approach. The detailed derivations of this technique are presented in Appendix A. It can be shown (see Appendix A) that the weights and resistances of the CHNN are equal to the coefficients in a spatially discretized heat conduction equation. In fact, is the values of w are similar to W when $I=0$ and $R=\infty$.

2.5. Connectivity structure

Fig. 2a shows the basic CHNN connectivity circuit diagram when all of the neuron weights are known. In practice, this study considers the temperature distributions of heat conduction models with boundary conditions and with some of the connective weight strengths equal to zero. Therefore, the basic CHNN circuit diagram shown in Fig. 2a can be modified to the form presented in Fig. 2b. In this figure, the function $f(t)$ represents the conversion between the temperature and the corresponding voltage, and the neuron weight strengths, W, are the inverse of the resistance values of the CHNN. In practice, the temperature distribution weight strengths are variable. Therefore, during the simulation process, a voltage converter is used to control the current source in order to introduce non-linear weight strengths throughout the circuit.

3. Formulation for inverse problems

3.1. Mapping neural network architecture

A feed forward network with n-input units and m-output units can perform mapping from an n-dimensional cube R^n to an m-dimensional cube R^m. According to the original theorem proposed by Kolmogorov [30]: *For all* $n\geqslant 2$, *and for any continuous real function g of n variables in the domain* $[0,1]$, g: $[0,1]^n\to R$, *there exist* $2n+1$ *continuous, monotonously increasing one-variable functions in* $[0,1]$, *by which g can be reconstructed according to the following equation*:

$$g(u)=\sum_{j=1}^{2n+1}\Psi_j\left(\sum_{i=1}^{n}\phi_{ij}(u_i)\right), \tag{6}$$

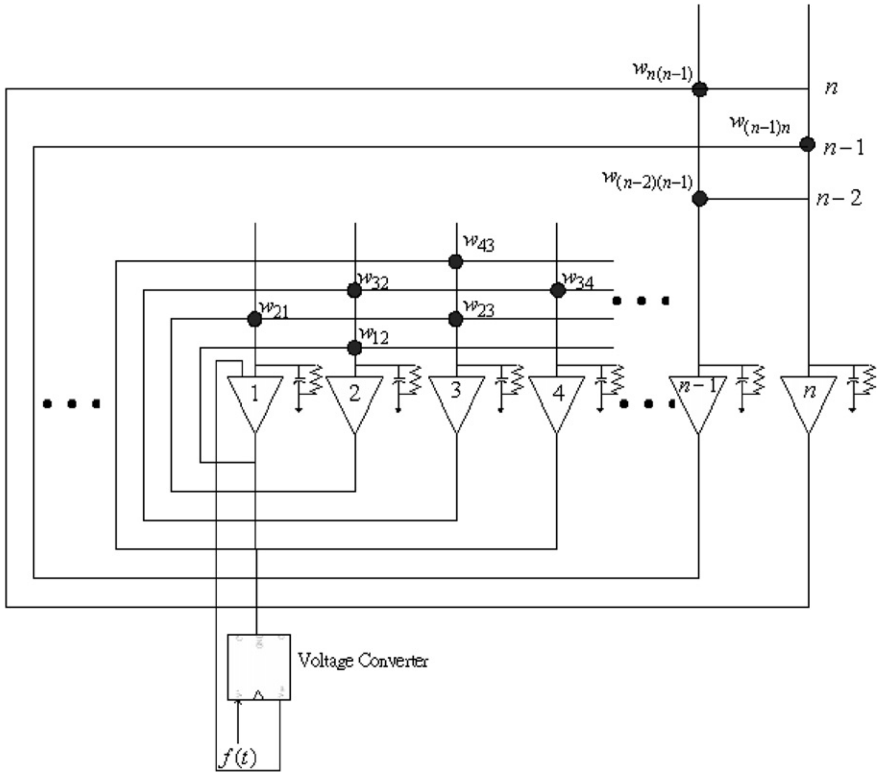


Fig. 2b. Modified Hopfield neural network connectivity circuit diagram for heat conduction temperature profiles.

where Ψ_j are continuous functions with one variable. Many authors have attempted to improve this theorem since its original formulation, most commonly by replacing ϕ_{ij} with $b_i\varphi_j$, where b_i is a constant. Therefore, Eq. (6) can be rewritten as

$$g(u) = \sum_{j=1}^{2n+1} \Psi_j\left(\sum_{i=1}^{n} b_i\varphi_j(u_i)\right). \tag{7}$$

Eq. (7) corresponds to a three-layered feed forward network architecture with a single output. Based on the theoretical foundation above, and using a more complex proof process [31], the following theorem can be obtained:

Let $\varphi(u)$ be a non-constant, bounded and monotonously increasing continuous function. There exist an integer k and sets of real constants c_i, θ_i and w_{ij}, where $(i = 1,2,\ldots,k)$, $(j = 1,2,\ldots,n)$ such that the expression:

$$\hat{g}(u_1,u_2,\ldots,u_n) = \sum_{i=1}^{k} c_i\varphi\left(\sum_{j=1}^{n} w_{ij}u_j - \theta_i\right) \tag{8}$$

can be defined to meet

$$\max|g(u_1,u_2,\ldots,u_n) - \hat{g}(u_1,u_2,\ldots,u_n)| < \varepsilon, \tag{9}$$

where w_{ij} corresponds to the neuron weight. From the theoretical description above, for the condition $\varepsilon > 0$, there exists a three-layered feedforward network in which the activation function of the hidden layer is $\varphi(u)$. The activation function of the input layer is non-linear, while that of the output layer is linear, and the three-layered feed forward network given in Eq. (8) satisfies Eq. (9). The theorem presented above provides the foundation for applying the BPN network to the solution of IHCPs.

3.2. Back propagation neural network

Fig. 3 illustrates a general BPN network. As shown, this network is a feed forward, fully connected hierarchical M-layered network consisting of an input layer, M-2 hidden layers and an output layer. If the kth unit in the Mth layer is denoted by (M,k), the state variable u_k^M for this unit and its output signal y_k^n to the units in the next layer $(M + 1,\ k)$ can be written as follows:

$$u_k^M = \sum_k (w_{k,j}^{M,M-1}u_j^{M-1} + \theta_k^M), \tag{10}$$

$$y_k^M = \varphi_k^M(u_k^M). \tag{11}$$

Here, $w_{k,j}^{M,M-1}$ is the connection strength between units (M,k) and $(M-1,j)$, and θ_k^M and $\varphi_k^M(\bullet)$ are the bias and activation functions of unit (M,k), respectively. The output signal y_k^M is transmitted to all units in the next $(M + 1)$th layer.

In the current study, the BPN network is used to solve various IHCPs. The input parameters, $u^0 = (u_1^0, u_2^0, \ldots)$, to the neural network are the temperature data at specified points in the interior of the object of interest, while the outputs of the network, $y^M = (y_1^M, y_2^M, \ldots)$, are parameters relating to the boundary conditions, e.g. the heat flux.

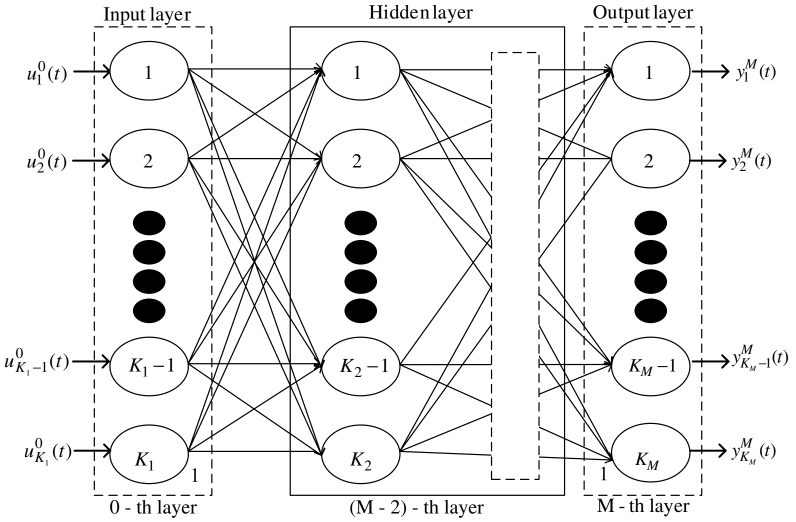


Fig. 3. Structure of multilayered neural network.

3.3. Formulation modes of training

Using a BPN network to solve IHCPs involves two basic stages, a learning stage and a recalling stage. The learning process is a supervised learning process which uses a training model and a set of target output values from the problem domain. If the training data set fully covers the problem space and training is successfully completed, in the recalling stage, the neural network can provide an appropriate output for any arbitrary unknown input. The goal of the training process is to modify the connection strengths $w_{k,j}^{M,M-1}$ and the biases θ_k^M which characterize the BPN network such that the actual output vector $y^M = (y_1^M, y_2^M, \ldots)$ approximates the target output vector, $d^M = (d_1^M, d_2^M, \ldots)$ as closely as possible.

Conventionally, a BPN algorithm iteratively adjusts the link-weights using the steepest descent technique. However, the resulting convergence is inherently slow and the solution may become trapped at local minima. Since the introduction of the original BPN learning algorithm, extensive research has been performed aimed at developing methods to accelerate the convergence rate. This research can be classified into two basic categories. The first category involves the development of ad hoc techniques [19], and includes such approaches as varying the learning rate or using momentum and rescaling variables, while the second category comprises standard numerical optimization techniques [21–24]. All of these techniques successfully improve the performance of the traditional BPN, and the choice of an appropriate algorithm depends on the target problem's complexity, the volume of training data available, the network size and the permissible error tolerance.

3.4. Improving generalization

When the volume of the available training data is limited or fixed, the network generalization capability is governed by the network structure. If the neural network structure is smaller than the available training data, the opportunity for over-training is small. When employing networks to solve a certain class of problem, including IHCPs, establishing a suitable neural network structure (in particular, establishing the number of neurons in the hidden layer) is challenging. A regularization method can be used to improve the neural network generalization capability. In regularization methods, a regularization term is added to the sum of the squared errors term of the objective function, i.e. the network objective function is modified to the following form:

$$F(w) = \xi_1 E_d + \xi_2 E_w, \tag{12}$$

where w is the network weight, $E_d = \sum_j (y_j - d_j)^2$ is the sum of the squared errors of the output, d_j and y_j are the desired output and the actual output, respectively, $E_w = \sum_j w_j^2$ is the regularization term, and ξ_1 and ξ_2 are the objective function parameters. Adding the regularization term to the objective function enables the smallest connection weights to approach a value of zero. This improves the network's fit and precision and reduces the network complexity, thereby obtaining an improved generalization.

The regularization coefficients ξ_1 and ξ_2 in Eq. (12) affect the training result. If ξ_2 is oversized, network overfitting can result. Conversely, if ξ_1 is oversized, underfitting may occur. The Bayesian regularization method selects appropriate values of ξ_1 and ξ_2 values in the network training stage, regards the weight value as a random variable, and assumes that the prior probability of the training data set d, and the weight collection set w obey the Gaussian distribution. Based on the Bayesian framework [32], the posterior probability is then maximized to obtain the minimized regularized objective function $F(w)$ at points ξ_1 and ξ_2, i.e.

$$\xi_1^{\mathrm{MP}} = \frac{\gamma}{2E_w(w^{\mathrm{MP}})} \quad \text{and} \quad \xi_2^{\mathrm{MP}} = \frac{N-\gamma}{2E_d(w^{\mathrm{MP}})}, \tag{13}$$

where $\gamma = k - \xi_1^{\mathrm{MP}} Trace(H^{\mathrm{MP}})^{-1}$, in which k is the total number of weights in the network, $H = \xi_2 \nabla^2 E_d(w^{\mathrm{MP}}) +$

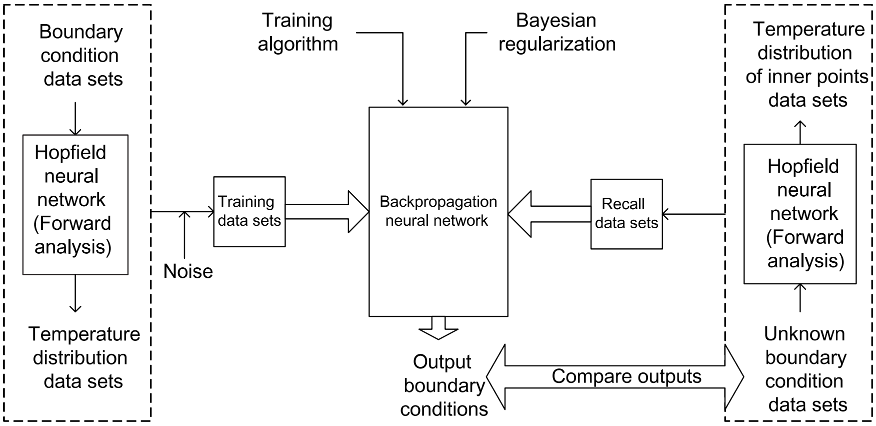

Boundary condition data sets
Hopfield neural network (Forward analysis)
Temperature distribution data sets
Noise
Training data sets
Training algorithm
Bayesian regularization
Backpropagation neural network
Recall data sets
Output boundary conditions
Compare outputs
Temperature distribution of inner points data sets
Hopfield neural network (Forward analysis)
Unknown boundary condition data sets

Fig. 4. Flow chart of forward and inverse neural network analysis.

$\xi_1 \nabla^2 E_w(w^{\mathrm{MP}})$, N is the product of the number of output neurons and the training data number, and γ is the number of parameters used in the neural network. If γ approaches k, the network scale is insufficiently large and the number of hidden layers must be increased. If the network is over a certain scale, the change value of γ becomes very small.

Bayesian regularization network training is a recursive process. Initially, with values of ξ_1 and ξ_2, network training is performed using the Levenberg–Marquardt (LM) algorithm to obtain the minimized $F(w)$ point, i.e. w^{MP}. Eq. (13) is then used to update the values of ξ_1 and ξ_2, and the LM algorithm is applied once again to train the network. This process is repeated recursively until the prescribed tolerance is obtained [33].

4. Framework of forward and inverse techniques

This study considers both one- and two-dimensional transient heat conduction problems. The test cases consider boundary heat fluxes with a variety of simple and complex waveforms. The solutions of the forward heat conduction problems obtained using the CHNN method are used to estimate the temperature distribution. Initially this study considers three standard heat conduction problems, namely a “triangular” initial temperature profile in a bar, a heat flux varying over time with a triangular profile, and a surface subjected to a constant flux. In the first two cases, the CHNN results are compared to the exact solutions in order to verify the accuracy of the proposed CHNN scheme, while in the third case, the CHNN solutions are compared with the results of the finite difference method. Having confirmed the accuracy of the CHNN method in solving conventional forward heat conduction problems, it is then applied to the solution of three rather more complicated time-varying boundary heat flux problems. When using the BPN to solve the current IHCPs, the three-layered BPN applied to the one-dimensional IHCP has six nodes in the input layer, four nodes in the hidden layer and one node in the output layer. Meanwhile, the BPN used to solve the two-dimensional IHCP has six nodes in the input layer, 15 nodes in the hidden layer and two nodes in the output layer. Regarding the activation functions of the neural network, the sigmoid function is chosen for the hidden layer and a linear function is used for the output layer.

Solving the forward heat conduction problem yields temperature data for a number of specified points inside the domain. To reflect actual engineering problems, a measurement noise of 10E−4 is added to the results of the temperature data to use as inputs to the BPN. Fig. 4 illustrates the forward and inverse neural network analysis procedure conducted in this study. Respective pairs of the boundary conditions and the temperature data calculated at specified points in the domain by the CHNN method were used to train the neural network using various training methods. After the training process had been completed, the forward heat conduction problems were solved for arbitrarily assigned boundary conditions to obtain temperature distributions in the domain. The boundary conditions for a given set of calculated temperature data were then predicted using the trained neural network. The predicted boundary conditions were then compared with the arbitrarily assigned boundary conditions in the previous step to confirm the validity of the proposed neural network analysis.

In this study, the accuracy of the network outputs obtained using the different training methods was evaluated using the root mean square error (MSE) indicator, defined as

$$\mathrm{MSE} = (1/n)\sum_{i=1}^{n}(y_i^{\mathrm{mea}} - d_i^{\mathrm{exa}})^2, \tag{14}$$

where y^{mea} is the neural network solution at time t, d^{exa} is the analytical solution or desired output for the same input data at time t, and n is the number of measurements.

5. Results and discussion

5.1. Forward problems

In the first two cases, the CHNN results were compared to the exact solutions to verify the accuracy of the proposed CHNN scheme. In the third case, the solutions of the CHNN scheme were compared to those of the FDM method. Having confirmed its performance in solving standard heat conduction problems, the CHNN method was then applied to three rather more complicated time-varying boundary heat flux problems. Again, the solutions of the CHNN scheme and the FDM method were compared.

The six heat flux profiles considered in the current simulations were as follows:

Case 1: A triangular temperature profile within a laterally insulated bar of length L, whose ends are maintained at zero temperature. The initial temperature is given by

$$f(x) = \begin{cases} x, & 0 < x < \frac{L}{2}, \\ L - x, & \frac{2}{L} < x < L. \end{cases} \tag{15a}$$

The exact solution is given by [34]

$$T(x,t) = \frac{4L}{\pi^2}\left\{\sin\frac{\pi x}{L}\exp\left[-\left(\frac{c\pi}{L}\right)^2 t\right] - \frac{1}{9}\sin\frac{3\pi x}{L}\exp\left[-\left(\frac{3c\pi}{L}\right)^2 t\right] - \cdots\right\}. \tag{15b}$$

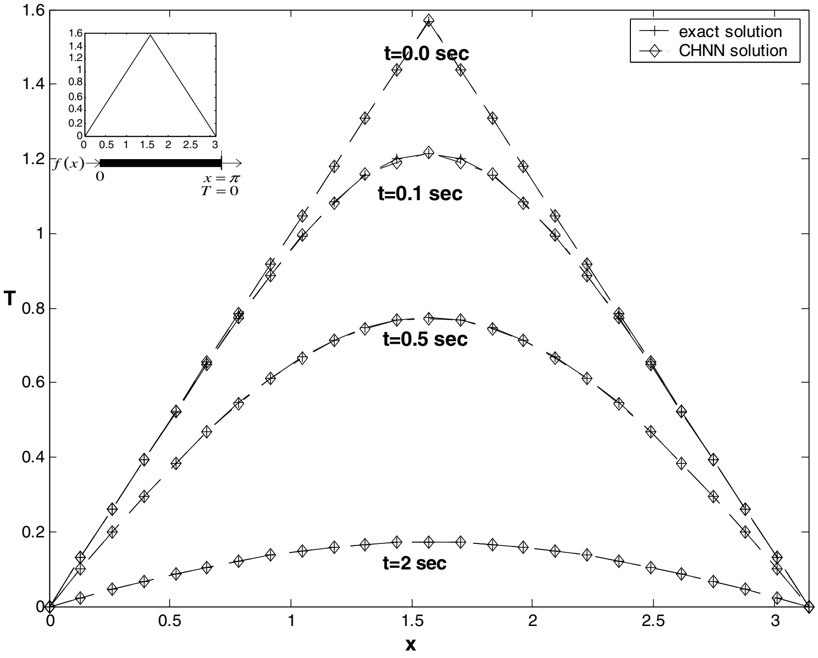


Fig. 5a. Comparison of exact and CHNN solutions for initial triangular temperature distribution in laterally insulated bar of length π and $c = 1$.

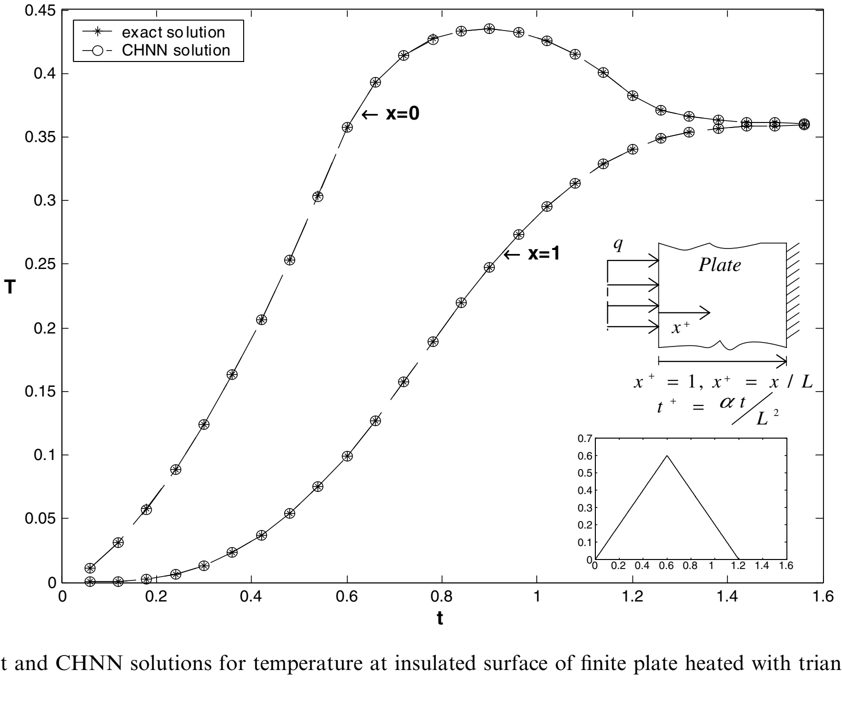


Fig. 5b. Comparison of exact and CHNN solutions for temperature at insulated surface of finite plate heated with triangular time-varying heat flux.

Case 2: Prior to $t^+ = 0$, the heat flux is zero. When t^+ lies between zero and 0.6, the surface flux, q, increases linearly with time. For $t^+ > 0.6$, the flux decreases linearly to zero at $t^+ = 1.2$ and remains at zero thereafter. The exact solutions for the temperatures at $x^+ = 0$ and $x^+ = 1$ for the linear heat flux are given by [1, pp. 169–329]:

$$\phi^+(0,t^+) = \frac{1}{2}(t^+)^2 + \frac{1}{3}t^+ - \frac{1}{45} + \frac{2}{\pi^4}\sum_{n=1}^{\infty}\frac{1}{n^4}\exp(-\pi^2 n^2 t^+), \tag{16a}$$

$$\phi^+(1,t^+) = \frac{1}{2}(t^+)^2 - \frac{1}{6}t^+ + \frac{7}{360} + \frac{2}{\pi^4}\sum_{n=1}^{\infty}\frac{(-1)^n}{n^4}\exp(-\pi^2 n^2 t^+), \tag{16b}$$

$$T^+(x,t^+) = \begin{cases} \phi^+(x^+,t^+), \quad 0 < t^+ \leqslant 0.6, \\ \phi^+(x^+,t^+) - 2\phi^+(x^+,t^+ - 0.6), \quad 0.6 < t^+ \leqslant 1.2, \\ \phi^+(x^+,t^+) - 2\phi^+(x^+,t^+ - 0.6) + \phi^+(x^+,t^+ - 1.2), \\ \quad t^+ > 1.2, \end{cases} \tag{16c}$$

where

$$T^+ \equiv \frac{T - T_0}{qL/k}, \quad t^+ \equiv \frac{\alpha t}{L^2}, \quad x^+ \equiv \frac{x}{L}. \tag{16d}$$

Table 1
Exact solution of temperatures at insulated surface $x^+ = 0$ of finite plate heated with triangular time-varying heat flux shown in Fig. 5b

$\Delta t^+ = 0.06$					
t^+	$T(0,t^+)$	t^+	$T(0,t^+)$	t^+	$T(0,t^+)$
0.06	0.011056	0.66	0.393497	1.26	0.371417
0.12	0.031271	0.72	0.414453	1.32	0.366259
0.18	0.057453	0.78	0.427080	1.38	0.363457
0.24	0.088499	0.84	0.433584	1.44	0.361912
0.30	0.123841	0.90	0.435099	1.50	0.361057
0.36	0.163166	0.96	0.432248	1.56	0.360585
0.42	0.206303	1.02	0.425373		
0.48	0.253158	1.08	0.414663		
0.54	0.303677	1.14	0.400223		
0.60	0.357828	1.20	0.382112		

Case 3: The initial temperature of a finite solid cylinder is zero and its surface is subjected to a constant flux. The temperature distribution in the cylinder is given by [35,1, pp. 17–18]

$$T_a^+(r^+,t_a^+) = 2t_a^+ + \frac{1}{2}(r^+)^2 - \frac{1}{4} - 2\sum_{n=1}^{\infty}\mathrm{e}^{-\beta_n^2 t_a^+}\frac{J_0(r^+\beta_n)}{\beta_n^2 J_0(\beta_n)}, \tag{17a}$$

where β_n, $n = 1, 2, \ldots$, are the positive roots of the Bessel function,

$$J_1(\beta_n) = 0, \tag{17b}$$

a is the cylinder radius, and

$$T_a^+(r^+,t_a^+) \equiv \frac{[T(r,t) - T_0]k}{q_c a}, \quad t_a^+ \equiv \frac{\alpha t}{a^2}, \quad r^+ \equiv \frac{r}{a}. \tag{17c}$$

Case 4: The time-varying flux has a step profile followed by a ramp profile (step-ramp profile) from $t = 0$ to $t_{\mathrm{f}} = 6$ with a time interval of 0.01, i.e.

$$q(t) = \begin{cases} 0, & 0 < t \leqslant 1, \\ 1, & 1 < t \leqslant 3, \\ \frac{1}{750}t - 0.2, & 3 < t \leqslant 6. \end{cases} \tag{18}$$

Case 5: The time-varying flux has a triangular profile followed by a sine profile (triangular–sine profile) from $t = 0$ to $t_{\mathrm{f}} = 6$ with a time interval of 0.01, i.e.

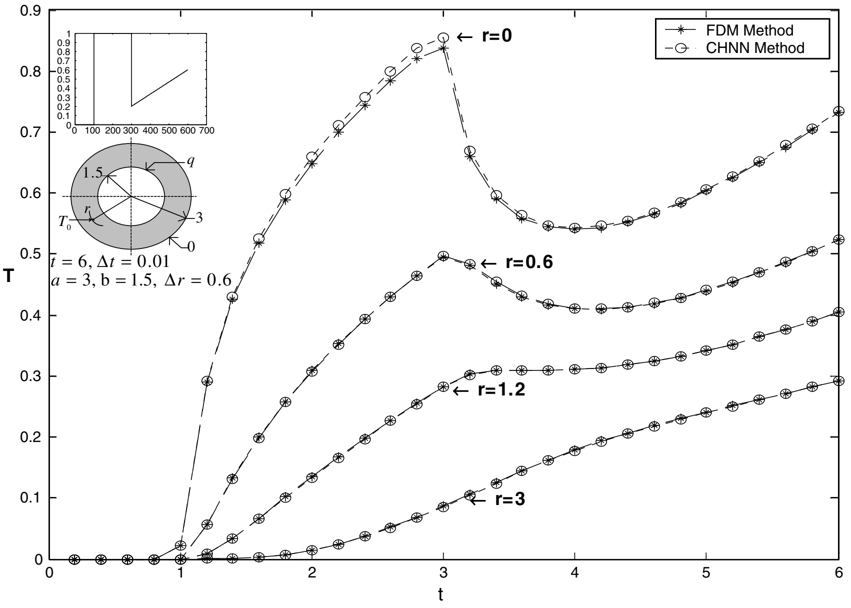


Fig. 6a. Comparison of CHNN and FDM results for temperature with step-ramp heat flux.

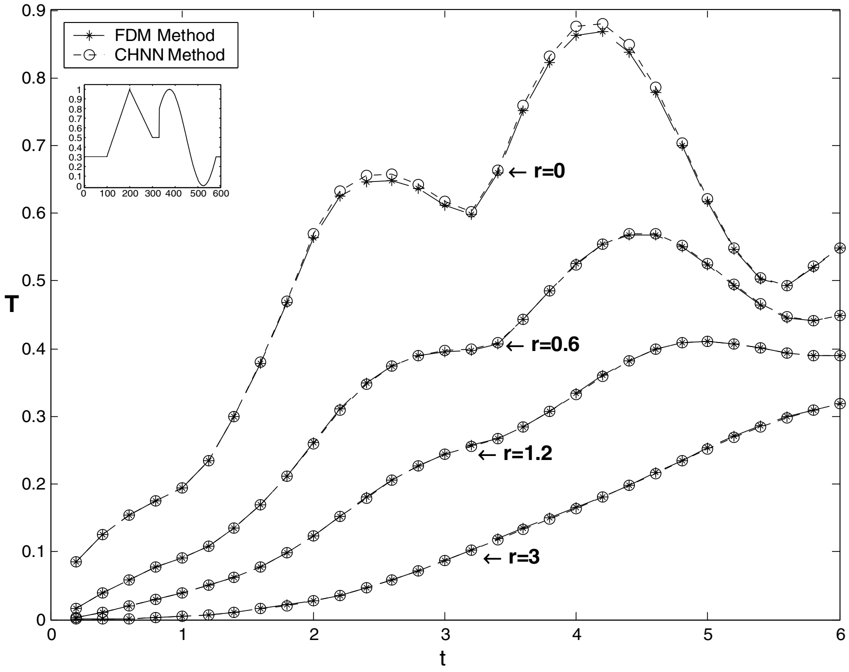


Fig. 6b. Comparison of CHNN and FDM results for temperature with triangular–sine heat flux.

$$q(t)=\begin{cases}0.3, & 0.0<t\leqslant 1.0,\\ 0.007t-0.4, & 1.0<t\leqslant 2.0,\\ -0.005t+2, & 2.0<t\leqslant 3.0,\\ 0.5, & 3.0<t\leqslant 3.3,\\ 0.5\left(1+\sin\left(\frac{\pi}{150}t\right)\right), & 3.3<t\leqslant 5.8,\\ 0.3, & 5.8<t\leqslant 6.0.\end{cases}\tag{19}$$

Case 6: The time-varying flux has a sine profile followed by a second sine profile (sine–sine profile) from $t=0$ to $t_f=6$ with a time interval of 0.01, i.e.

$$q(t)=\begin{cases}\sin(2\pi t/250), & 0<t\leqslant 3,\\ 0.5\sin(2\pi t/250)+0.5\sin(2\pi t/25), & 3<t\leqslant 6.\end{cases}\tag{20}$$

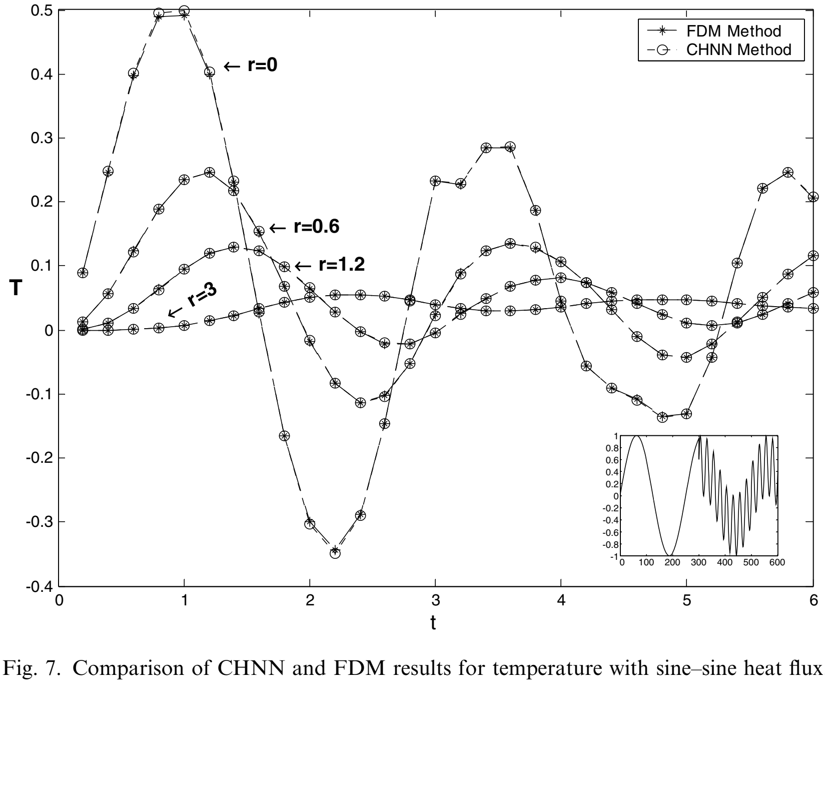


Fig. 7. Comparison of CHNN and FDM results for temperature with sine–sine heat flux.

5.1.1. One-dimensional forward heat conduction problem

Figs. 5a and 5b compare the exact solutions and the CHNN solutions for Cases 1 and 2, respectively. The MSE for the CHNN results of Case 1 is 1.61E−06. For Case 2, the MSE at $x = 0$ is 1.32E−08 while at $x = 1$, the MSE is 1E−09. Eq. (16c) provides the numerical values of $T(0, t^+)$ for the case of a triangular time-varying heat flux. The corresponding results are presented in Table 1 for $\delta t^+ = 0.06$.

In Case 3, the initial temperature of the finite solid cylinder is zero and the surface is subjected to a constant flux. In this study, the resulting temperature distribution was solved using both the CHNN scheme and the FDM numerical technique. The MSE results for the two methods at $r = 0$ were found to be 5.23E−08 for the CHNN method and 4.27E−07 for the FDM method.

From these results, it is clear that the CHNN method provides accurate solutions for the temperature distribution in one-dimensional heat conduction problems, irrespective of the initial temperature or the variation of the heat flux at the boundary over time. The results of Cases 1 and 2 show that the solutions of the CHNN scheme are in good agreement with the exact results. Furthermore, the results of Case 3 indicate that the performance of the CHNN method is slightly better than that of the FDM approach. Having confirmed the capability of the pro-

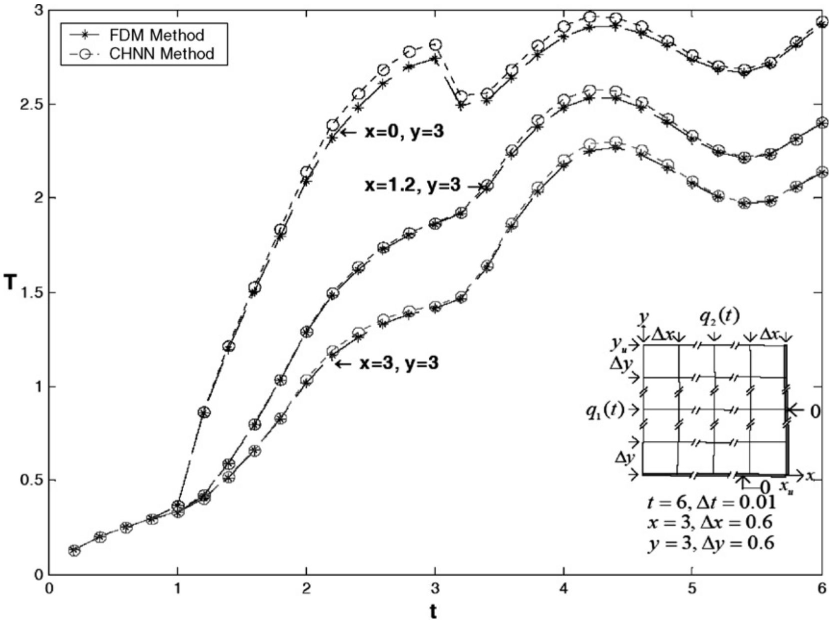


Fig. 8. Comparison of CHNN and FDM results for temperature for step-ramp heat flux at $x = 0$ and triangular–sine at $y = 0$.

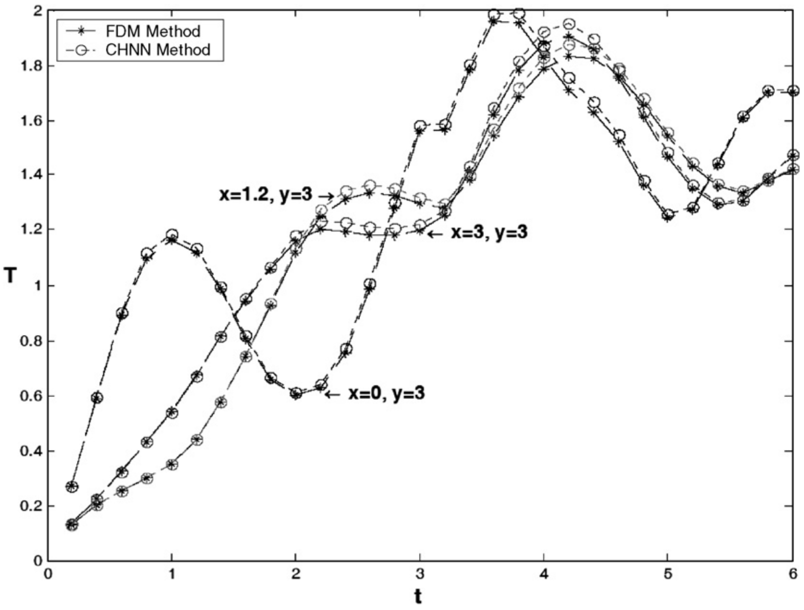


Fig. 9. Comparison of CHNN and FDM results for temperature for sine–sine heat flux at $x = 0$ and triangular–sine heat flux at $y = 0$.

posed CHNN method in solving the standard heat conduction problems of Cases 1–3, the CHNN model was then applied to a series of one- and two-dimensional heat conduction problems with more complicated heat flux profiles.

Eqs. (18)–(20) describe the three time-varying boundary heat fluxes considered in the second set of simulations. The one-dimensional cylindrical temperature history results for the insulated surface and at specified inner points within the cylinder are presented in Figs. 6a, 6b and 7 (Cases 4–6 above).

5.1.2. Two-dimensional forward heat conduction problems

For the two-dimensional cases (Figs. 8–10), the input heat flux profiles were specified as follows:

(1) A step-ramp heat flux at $x = 0$ (described by Eq. (18)) and a triangular–sine heat flux at $y = 0$ (described by Eq. (19)). The corresponding results are illustrated in Fig. 8.

(2) A sine–sine heat flux at $x = 0$ (described by Eq. (20)) and a triangular–sine heat flux at $y = 0$ (described by Eq. (19)). The corresponding results are illustrated in Fig. 9.

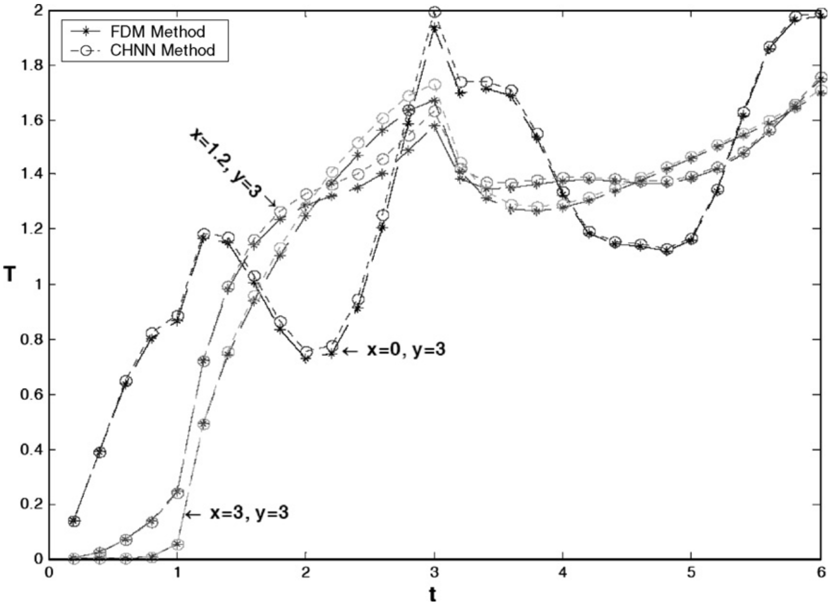


Fig. 10. Comparison of CHNN and FDM results for temperature for sine–sine heat flux at $x = 0$ and step-ramp heat flux at $y = 0$.

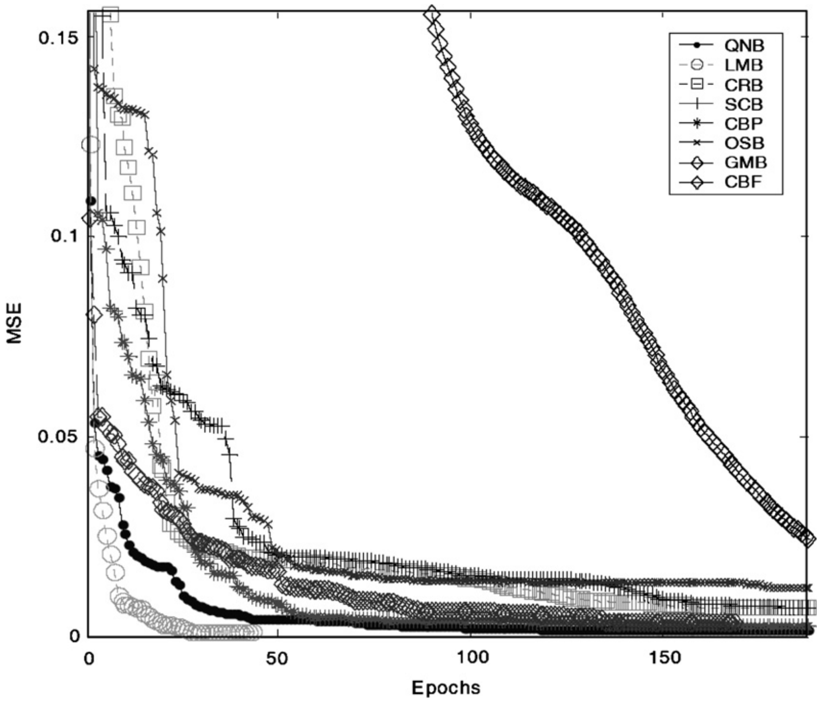


Fig. 11a. Comparison of convergence performance of eight training algorithms for Case 4.

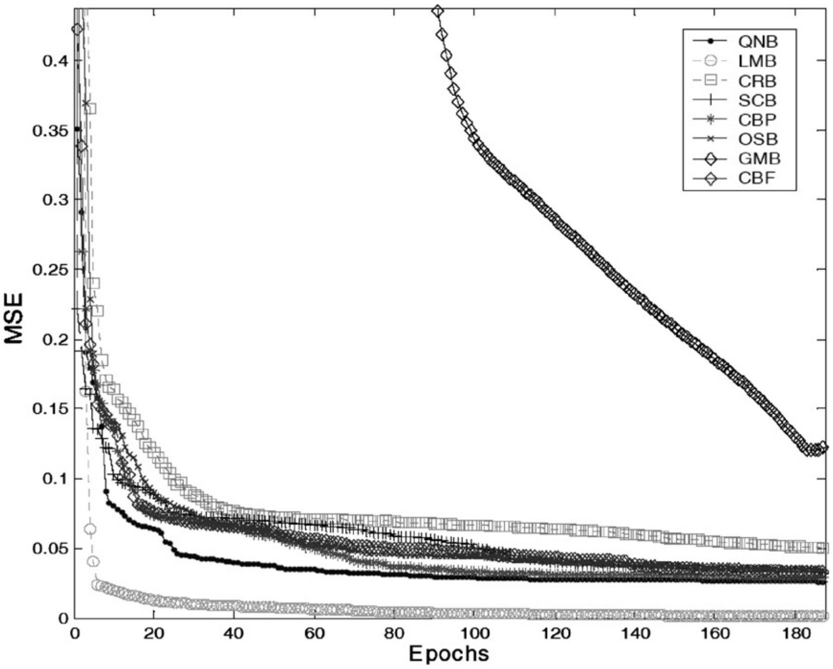


Fig. 11b. Comparison convergence performance of eight training algorithms for Type III.

(3) A sine–sine heat flux at $x = 0$ (described by Eq. (20) and a step-ramp heat flux at $y = 0$ (described by Eq. (18)). The corresponding results are illustrated in Fig. 10.

Similarly, the temperature data calculated by the CHNN method and the FDM scheme at $x = 3$ and $y = 3$ for the two-dimensional cases are plotted in Figs. 8–10, respectively. In general, the simulation results obtained by the two methods are in good agreement. In the second set of simulations (i.e. Figs. 6a, 6b and 7–10), the MSE value discrepancy between the CHNN and FDM solutions varies from 0.1% (Figs. 6a, 6b and 7) to 0.3% Figs. 8–10. This compares to an MSE value of E−06 between the CHNN results and the exact solutions in Cases 1 and 2 and an MSE value of E−07 between the CHNN/FDM solutions and the exact solution in Case 1.

On the evidence of the results obtained from the first three cases (i.e. the standard heat conduction cases), in which the results (Figs. 5a, 5b and Table 1) of the CHNN method are compared with the exact solutions and those of the FDM scheme, it can be inferred that the proposed method can efficiently compute the temperature distribution both at a heated surface and at an insulated surface.

5.2. Inverse heat conduction problems

5.2.1. One-dimensional

In this study, training data for the BPN network were prepared by solving the forward problems using the CHNN method under the boundary conditions of an arbitrarily assigned heat flux, q. The CHNN Eqs. (1a)–(1d), (2b)–(2d) were discretized. In the discretization procedure, the calculation domain was divided into six nodal points, values of the heat flux, q, and the temperature distribution, T, were prepared, and the value of the time constant was generated in the range 0–6 s. A total of 601 sets of time-history temperature data were obtained for each of the heat fluxes of Cases

Table 2
Performance of eight training algorithms in terms of iterations and MSE for one-dimensional cases

Algorithm	Type					
	Case 4		Case 5		Case 6	
	Iterations	MSE	Iterations	MSE	Iterations	MSE
QNB	211	0.000981	216	0.001000	3000	0.003661
LMB	56	0.000816	35	0.000972	3000	0.002882
CRB	3000	0.001020	3000	0.001880	3000	0.034049
SCB	964	0.000997	3000	0.000998	3000	0.024123
CBP	500	0.000999	3000	0.001119	355	0.038943
OSB	3000	0.001283	2105	0.000999	3000	0.035751
GMB	3000	0.008931	3000	0.005648	3000	0.059163
CBF	3000	0.001798	442	0.000999	472	0.040512

4–6 shown in Figs. 6a and 6b (top left) and Fig. 7 (bottom right), respectively. These data were then used as input to the inverse problems. A total of 2404 sets of temperature history data were then calculated by the CHNN method for four different measurement positions, as shown in Figs. 6a, 6b, 7 and 8 by the circular symbol data. This temperature data were then used to train the BPN network designed to solve the current IHCPs. In the IHCP (recalling) stage for

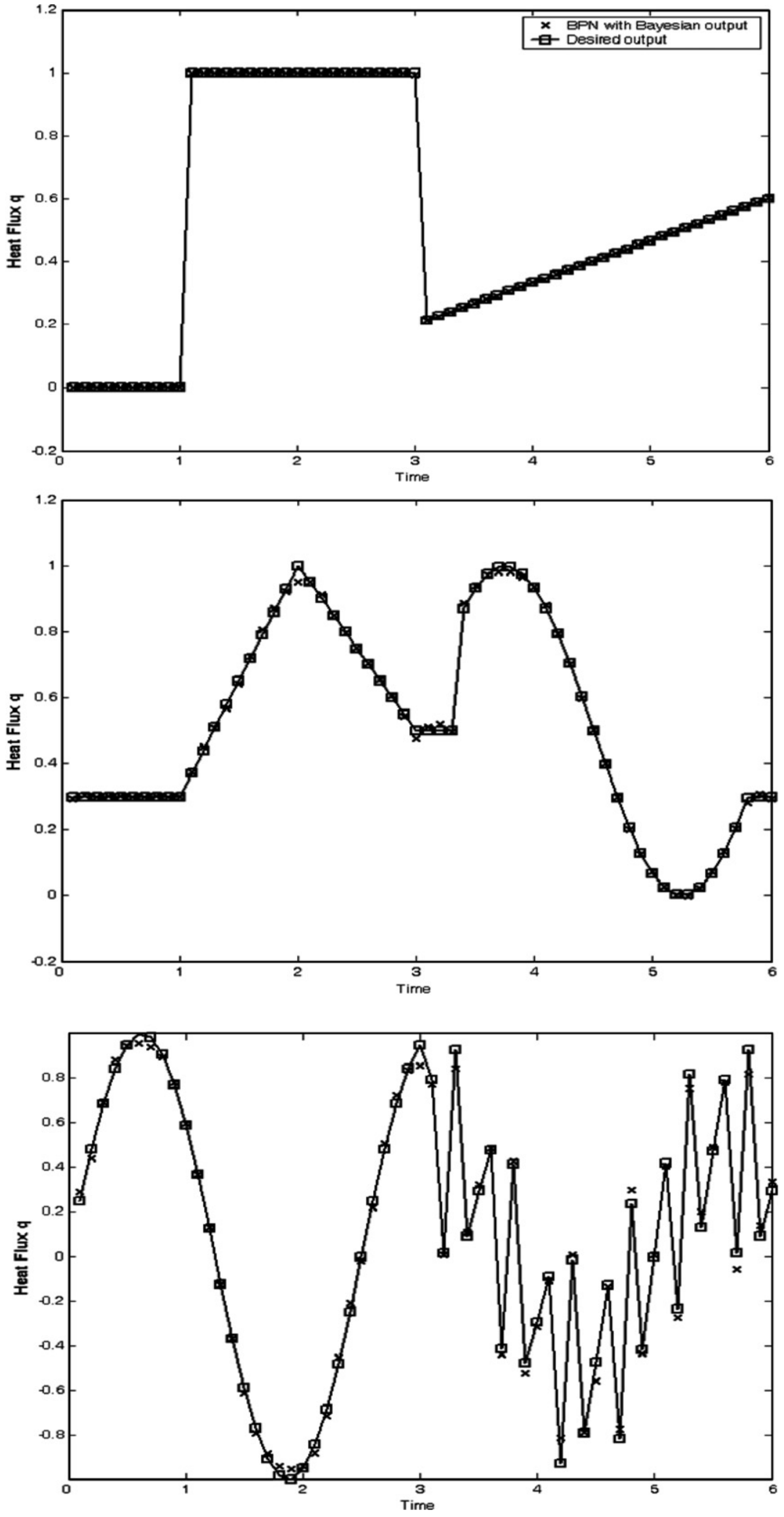


Fig. 12. Estimated heat flux of one-dimensional IHCP, Top: Case 4, Middle: Case 5 and Bottom: Case 6.

these three cases, a total of 200 sets of temperature history data (corresponding to four different measurement positions) were calculated by the CHNN method.

The neural network assumed the following parameters: four units in the hidden layer, a learning rate of 0.0001, a momentum rate of 0.09, 3000 convergence iterations, and a target error of 0.0001. Network training was performed using eight different algorithms, i.e. QNB, LMB, CRB, SCB, CBP, OSB, GMB, CBF, and Bayesian regularization. Fig. 11a and Table 2 compare the convergence characteristics of the eight training algorithms when applied to Case 4. Note that the results for Cases 5 and 6 are not shown in Fig. 11a because the convergence rates are virtually identical to those shown in Case 4. From Table 2, it is clear that the QNB and LMB algorithms yield a superior convergence performance and that the LMB scheme yields the minimum MSE for Cases 4 and 5. Furthermore, the LMB algorithm also provides the best convergence performance for Case 6. However, it nevertheless fails to achieve the target error of 0.0001, and the number of iterations required is higher.

Having evaluated the performance of each of the eight training algorithms, the best method, i.e. the LMB algorithm, was employed to train the network and the results compared with those obtained from a network trained using the Bayesian regularization scheme. The MSE values of the outputs obtained from the Bayesian regularization-trained network for Cases 4–6 were found to be 0.0018%, 0.01%, and 0.17%, respectively, while the MSE values obtained from the LMB-trained network for the same cases were 0.08%, 0.09%, and 0.29%, respectively. Therefore, although the Bayesian network requires more iteration in its solution procedure, its results are more accurate. Fig. 12 shows the heat flux profiles of Cases 4–6, denoted by cross symbols, which were used as the boundary conditions for the forward problem in preparing the input data for the neural network. The calculated temperature data were input to a neural network fully trained by the Bayesian generalization method. The output from the trained neural network, denoted by the circular symbols, was then compared with the desired output, as shown in Fig. 12.

Table 3
Performance of eight training algorithms in terms of iterations and MSE for two-dimensional cases

Algorithm	Type					
	Case 4 ($x = 0$)		Case 6 ($x = 0$)		Case 6 ($x = 0$)	
	Case 5 ($y = 0$)		Case 4 ($y = 0$)		Case 5 ($y = 0$)	
	Iteration	MSE	Iteration	MSE	Iteration	MSE
QNB	231	0.000999	3000	0.001294	3000	0.001174
LMB	56	0.000999	1610	0.000999	409	0.000997
CRB	1252	0.001000	3000	0.020345	3000	0.022617
SCB	3000	0.006254	3000	0.004101	3000	0.005028
CBP	1266	0.001000	1805	0.006292	1606	0.007411
OSB	3000	0.001041	2500	0.012028	3000	0.019309
GMB	3000	0.006352	3000	0.032036	3000	0.040580
CBF	1577	0.000999	2193	0.007269	1510	0.014962

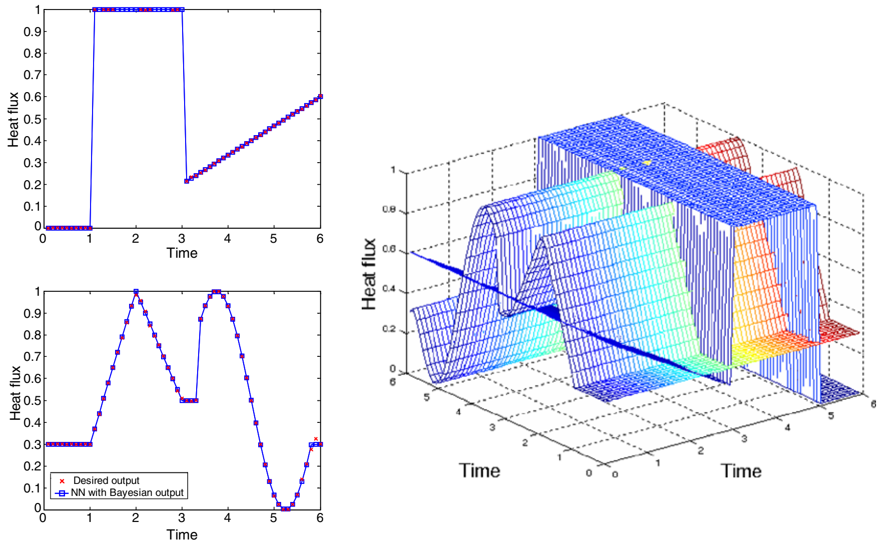


Fig. 13. Estimated heat flux of two-dimensional IHCP, Case 4 ($x = 0$) and Case 5 ($y = 0$).

5.2.2. Two-dimensional

The training data for the neural network were prepared by solving the forward problems using the CHNN under the boundary conditions of arbitrarily assigned heat fluxes $q_1(x = 0)$ and $q_2(y = 0)$. The CHNN Eqs. ((3a)–(3f)) were discretized. For the two-dimensional cases, the input heat flux profiles were as follows: Type I: a step-ramp heat flux at $x = 0$ (described by Eq. (18)) and a triangular–sine heat flux at $y = 0$ (described by Eq. (19)), Type II: a sine–sine heat flux at $x = 0$ (described by Eq. (20)) and a triangular–sine heat flux at $y = 0$ (described by Eq. (19)), and Type III: a sine–sine heat flux at $x = 0$ (described by Eq. (20)) and a step-ramp heat flux at $y = 0$ (described by Eq. (18)). As before, the calculation domain was divided into six nodal points at $x = 0$ and $y = 0$, respectively, temperature distributions, T, were prepared, and the time constant was generated in the range 0–6 s. After solving the forward heat conduction problem with these boundary conditions, a total of 2404 sets of temperature data were obtained at four different measurement points. The training data were then prepared by combining the temperature data at these four points with the given boundary conditions of q_1 and q_2.

The neural network was trained using the training data prepared above and the same eight training algorithms as used in the one-dimensional case. However, in this case, the neural network had a 6–15–2 framework. The learning rate parameter was 0.0001, the momentum rate was 0.09, the number of iterations in the solution procedure was 3000, and the target error was 0.0001. Fig. 11b demonstrates the convergence characteristics of these methods for the Type III input heat flux profile. Note that the results for Types I and II are not presented in this figure because

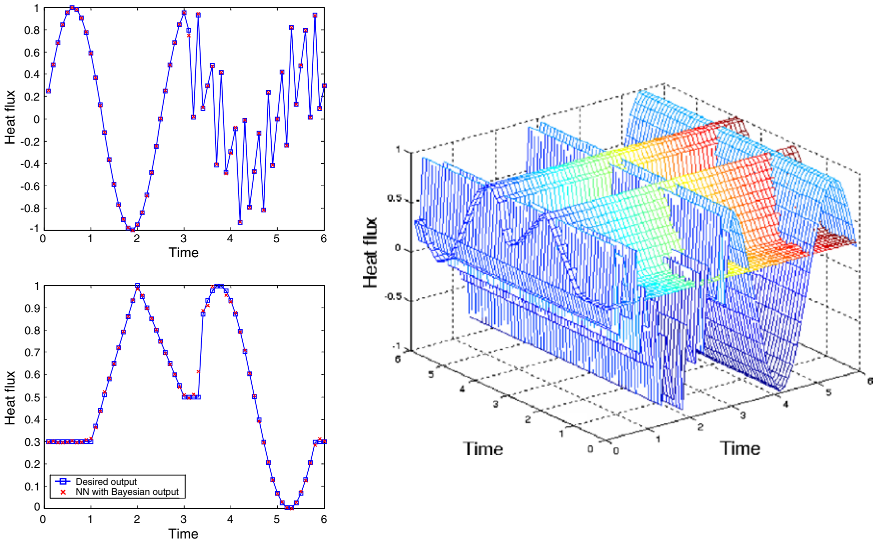


Fig. 14. Estimated heat flux of two-dimensional IHCP, Case 6 ($x = 0$) and Case 5 ($y = 0$).

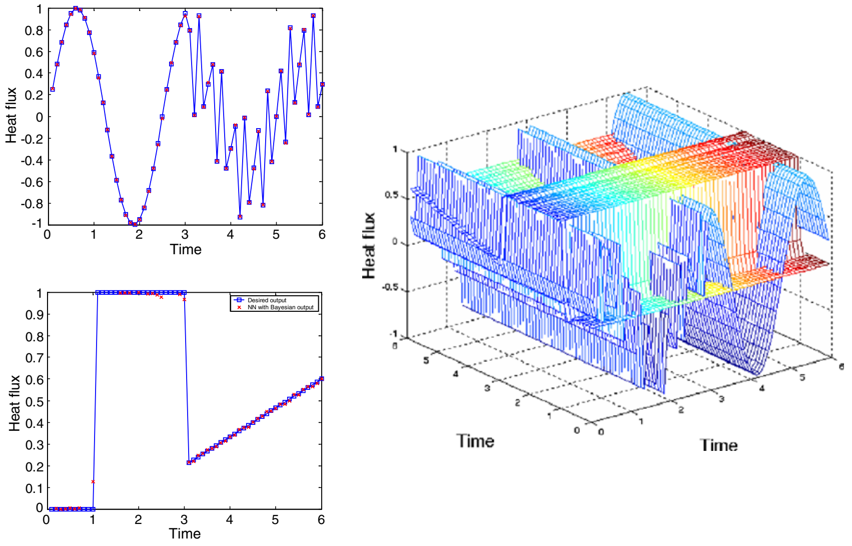


Fig. 15. Estimated heat flux of two-dimensional IHCP, Case 6 ($x = 0$) and Case 4 ($y = 0$).

their convergence speeds are virtually the same as those of Case 6 (“Type III”). Table 3 lists the convergence characteristics of the eight training algorithms. It is apparent that the LMB algorithm consistently achieves the best performance, both in terms of the convergence speed and the MSE error.

Accordingly, simulations for the Types I, II, and III heat influx profiles were performed using a network trained using the LMB algorithm. The results were then compared with those obtained from a network trained using the Bayesian regularization scheme. The MSE values obtained from the BPN trained using the Bayesian regularization scheme were found to be 0.00091%, 0.0079%, and 0.026% for the Type I, Type II, and Type III profiles, respectively. The Bayesian method yields a better convergence than the LMB, but requires more iteration. Figs. 13–15 shows the Types I–III heat flux profiles, denoted by cross symbols, used as the boundary conditions for the forward problem in preparing the input data for the neural network. The calculated temperature data were input to the neural network trained by the Bayesian generalization method. The output from the trained neural network, denoted by circular symbols, was then compared with the desired output, as shown in Figs. 13–15.

6. Conclusion

This paper has applied the continuous-time analogue Hopfield neural network (CHNN) to solve time-varying forward heat conduction problems. The relationship between the CHNN synaptic connection weights and the governing equations of the one- and two-dimensional heat conduction problems has been established. A neural network algorithm for inverse heat conduction problems (IHCPs) has been proposed to predict the unknown parameters in the boundary conditions. The forward simulation results have confirmed that the proposed CHNN scheme can successfully solve the temperature fields of both one- and two-dimensional heat conduction problems. Two inverse heat conduction problems have been considered to test the validity of the proposed method using a back-propagation neural network. The results have shown that the Bayesian method provides the best training performance. To obtain reasonable results, the neural network should be fully trained with a set of training data which covers the complete anticipated range of the unknown inputs. In conclusion, the method proposed in this study provides an accurate and convenient technique for the real-time determination of heat conduction transient state temperature distributions. The test results have shown that the proposed method is capable of predicting the unknown parameters in IHCPs with an acceptance error tolerance.

Appendix A

This study employs the central-difference approximation for the space derivative to formulate the relationship between the temperature and the boundary conditions. Using this approach, Eq. (3a) is obtained as

$$\frac{\partial T_{i,j}}{\partial t}=\frac{T_{i+1\cdot j}(t)-2T_{i,j}(t)+T_{i-1,j}(t)}{\Delta x^2}+\frac{T_{i\cdot j+1}(t)-2T_{i,j}(t)+T_{i,j-1}(t)}{\Delta y^2}$$
$$\text{for } i=1,2,\ldots,M,\quad j=1,2,\ldots,N,\text{ and }\ M,N>1, \tag{A1}$$

where M and N are the total number of spatial nodes in the x- and y-directions, respectively, and $\Delta x = x_u/M$ and $\Delta y = y_u/N$ are the space intervals in the x- and y-directions, respectively.

Rearranging Eq. (A1), it can be shown that:

$$\frac{\partial T_{i,j}}{\partial t}=\sum_{j=1}^{N}W_{ij}(T_j-T_i)+\sum_{i=1}^{N}W_{ji}(T_i-T_j). \tag{A2}$$

According to Eq. (A2), the difference between the temperature of the ith neuron and its neighbor is multiplied by $W_{i(i+1)} = 1/\Delta x^2$ and $W_{i(i-1)} = 1/\Delta x^2$, while the other elements in the **W** matrix are equal to zero. Similarly, the difference between the temperature of the jth neuron and its neighbor is multiplied by $W_{j(j+1)} = 1/\Delta y^2$ and $W_{j(j-1)} = 1/\Delta y^2$, while the other elements in the **W** matrix are equal to zero.

Preceding sections have derived heat conduction models (Eq. (A1)) and the CHNN differential equations (Eqs. (4a) and (4b)). In Eq. (4a), C is assumed to be constant and the CHNN weights, w, are similar to W, when $I = 0$ and $R = \infty$. In other words, Eq. (4a) is similar to Eq. (A1). The current numeric simulations use discrete time steps, and Eq. (4a) can be rewritten with a neuron internal state T_i as follows:

$$T_i^{t+1}=T_i^t+\sum_{\substack{j=1\\ i\neq j}}^{n}[W_{ij}\widehat{T}_i^t]\delta t\quad\text{for } i=1,2,\ldots,n, \tag{A3}$$

where $\widehat{T}_i$ is the difference between the temperature of the ith neuron and its neighbour and δt is the interval time.

At the boundary of $x = 1$, $T_{0,j}(t)$ is solved from Eq. (3c) as

$$\frac{\partial T_{1,j}(t)}{\partial r}=\frac{T_{2,j}(t)-T_{0,j}(t)}{2\Delta x}=-q_1(t).$$

This yields:

$$T_{0,j}(t)=T_{2,j}(t)+2\Delta x q_1(t). \tag{A4}$$

Similarly, the temperature at the boundaries of $y = 1$, $x = x_u$ and $y = y_u$ can be solved using Eqs. ((3d)–(3f)), respectively, i.e.

$$T_{i,0}(t)=T_{i,2}(t), \tag{A5}$$

$$T_{M+1,j}(t)=T_{M-1,j}(t), \tag{A6}$$

$$T_{i,N+1}(t)=T_{i,N-1}(t)+2\Delta y q_2(t). \tag{A7}$$

From Eqs. (A2) and (A4)–(A7), the continuous-time state can be expressed in the following matrix form:

$$\dot{\mathbf{T}}(t)=\mathbf{W}\mathbf{T}(t)+\Psi q(t), \tag{A8}$$

where $\mathbf{W}$ is the neural network connectivity strength matrix. This matrix is of the form $(M \times N) \times (M \times N)$, and is given by

$$\mathbf{W} = \begin{bmatrix} \bar{\mathbf{w}}_1 & \bar{\mathbf{w}}_3 & o & o & o & \cdots & o \\ \bar{\mathbf{w}}_2 & \bar{\mathbf{w}}_1 & \bar{\mathbf{w}}_2 & o & o & \cdots & o \\ o & \bar{\mathbf{w}}_2 & \bar{\mathbf{w}}_1 & \bar{\mathbf{w}}_2 & o & \cdots & o \\ \vdots & & & \ddots & & & \vdots \\ o & \cdots & o & \bar{\mathbf{w}}_2 & \bar{\mathbf{w}}_1 & \bar{\mathbf{w}}_2 & o \\ o & \cdots & o & o & \bar{\mathbf{w}}_2 & \bar{\mathbf{w}}_1 & \bar{\mathbf{w}}_2 \\ o & \cdots & o & o & o & \bar{\mathbf{w}}_3 & \bar{\mathbf{w}}_1 \end{bmatrix}, \tag{A9}$$

in which

$$\bar{\mathbf{w}}_1 = \begin{bmatrix} 0 & 2/\Delta y^2 & 0 & 0 & 0 & \cdots & 0 \\ 1/\Delta y^2 & 0 & 1/\Delta y^2 & 0 & 0 & \cdots & 0 \\ 0 & 1/\Delta y^2 & 0 & 1/\Delta y^2 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & 0 & 1/\Delta y^2 & 0 & 1/\Delta y^2 & 0 \\ 0 & \cdots & 0 & 0 & 1/\Delta y^2 & 0 & 1/\Delta y^2 \\ 0 & \cdots & 0 & 0 & 0 & 2/\Delta y^2 & 0 \end{bmatrix} \tag{A10}$$

and

$$\bar{\mathbf{w}}_2 = \begin{bmatrix} 1/\Delta x^2 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1/\Delta x^2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1/\Delta x^2 & 0 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \cdots & 0 & 0 & 1/\Delta x^2 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1/\Delta x^2 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1/\Delta x^2 \end{bmatrix} \tag{A11}$$

and $\bar{\mathbf{w}}_3 = 2\bar{\mathbf{w}}_2$. The sub-matrices $\bar{\mathbf{w}}_1$, $\bar{\mathbf{w}}_2$, $\bar{\mathbf{w}}_3$, and $\mathbf{o}$ are $M \times N$ matrices. Sub-matrix o is a null sub-matrix. The state matrix $T(t)$ is given by

$$T(t) = [\mathbf{T}_1(t) \quad \mathbf{T}_2(t) \quad \cdots \quad \mathbf{T}_{M-1}(t) \quad \mathbf{T}_M(t)]^{\mathrm{T}} \tag{A12}$$

where:

$$\mathbf{T}_1(t) = \begin{bmatrix} 0 & T_{1,1} - T_{1,2} & 0 & \cdots & 0 & 0 & 0 \\ T_{1,2} - T_{1,1} & 0 & T_{1,2} - T_{1,3} & & \vdots & \vdots & \vdots \\ 0 & T_{1,3} - T_{1,2} & 0 & & 0 & 0 & 0 \\ 0 & 0 & T_{1,4} - T_{1,3} & \ddots & T_{1,N-3} - T_{1,N-2} & 0 & 0 \\ 0 & 0 & 0 & & 0 & T_{1,N-2} - T_{1,N-1} & 0 \\ \vdots & \vdots & \vdots & & T_{1,N-1} - T_{1,N-2} & 0 & T_{1,N-1} - T_{1,N} \\ 0 & 0 & 0 & \cdots & 0 & T_{1,N} - T_{1,N-1} & 0 \end{bmatrix} \tag{A13}$$

$$\vdots$$

$$\mathbf{T}_M(t) = \begin{bmatrix} 0 & T_{M,1} - T_{M,2} & 0 & \cdots & 0 & 0 & 0 \\ T_{M,2} - T_{M,1} & 0 & T_{M,2} - T_{M,3} & & \vdots & \vdots & \vdots \\ 0 & T_{M,3} - T_{M,2} & 0 & & 0 & 0 & 0 \\ 0 & 0 & T_{M,4} - T_{M,3} & \ddots & T_{M,N-3} - T_{M,N-2} & 0 & 0 \\ 0 & 0 & 0 & & 0 & T_{M,N-2} - T_{M,N-1} & 0 \\ \vdots & \vdots & \vdots & & T_{M,N-1} - T_{M,N-2} & 0 & T_{M,N-1} - T_{M,N} \\ 0 & 0 & 0 & \cdots & 0 & T_{M,N} - T_{M,N-1} & 0 \end{bmatrix}. \tag{A14}$$

The input matrix Ψ is of the form $(M \times N) \times 2$, and is given by

$$\Psi = [\,\overline{\Psi}_{\mathbf{1}} \quad \overline{\Psi}_{\mathbf{2}} \quad \cdots \quad \overline{\Psi}_2\,]^{\mathrm{T}}, \tag{A15}$$

where the sub-matrices Ψ_1 and Ψ_2 are both of the form $N \times 2$ and are given by

$$\overline{\Psi}_{\mathbf{1}} = \begin{bmatrix} 2/\Delta x & 0 \\ 2/\Delta x & 0 \\ \vdots & \vdots \\ 2/\Delta x & 0 \\ 2/\Delta x & 2/\Delta y \end{bmatrix} \tag{A16}$$

and

$$\overline{\Psi}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 2/\Delta y \end{bmatrix}. \tag{A17}$$

The input matrix $\mathbf{q}(t)$ is of the form 2×1, and is given by

$$\mathbf{q(t)} = [q_1(t) \quad q_2(t)]^{\mathrm{T}}. \tag{A18}$$

References

[1] J.V. Beck, B. Blackwell, C.R.ST. Clair, Inverse Heat Conduction Ill-posed Problems, Wiley, New York, 1985, pp. 1–3.

[2] N. Al-Khalidy, Analysis of boundary inverse heat conduction problems using space marching with Savitzky–Gollay digital filter, Int. Commun. Heat Mass Transfer 26 (2) (1999) 199–208.

[3] M.S. Shin, J.W. Lee, Prediction of the inner wall shape of an eroded furnace by the nonlinear inverse heat conduction technique, JSME Int. J. B 43 (4) (2000) 544–549.

[4] D. Maillet, A. Degiovanni, R. Pasquetti, Inverse heat conduction applied to the measurement of heat transfer coefficient on a cylinder. Comparison between an analytical and a boundary element technique, J. Heat Transfer Trans. ASME 113 (3) (1991) 549–557.

[5] T.C. Chen, P.C. Tuan, Input estimation method including finite-element scheme for solving inverse heat conduction problems, Numer. Heat Transfer, Part B: Fundam. 47 (3) (2005) 277–290.

[6] R.A. Khachfe, Y. Jarny, Numerical solution of 2-d nonlinear inverse heat conduction problems using finite-element techniques, Numer. Heat Transfer: Part B: Fundam. 37 (1) (2000) 45–67.

[7] A.A. Tseng, F.Z. Zhao, Multidimensional inverse transient heat conduction problems by direct sensitivity coefficient method using a finite-element scheme, Numer. Heat Transfer: Part B: Fundam. 29 (3) (1996) 365–380.

[8] K.J. Dowding, J.V. Beck, A Sequential gradient method for the inverse heat conduction problem (IHCP), J. Heat Transfer Trans. ASME (121) (1999) 300–306.

[9] D. Lesnic, L. Elliott, D.B. Ingham, Application of the boundary element method to inverse heat conduction problems, Int. J. Heat Mass Transfer 39 (7) (1996) 1503–1517.

[10] C.H. Hung, C.W. Chen, A boundary-element-based inverse problem of estimating boundary conditions in an irregular domain with statistical analysis, Numer. Heat Transfer: Part B: Fundam. 33 (2) (1998) 251–268.

[11] S.K. Kim, I.M. Daniel, Solution to inverse heat conduction problem in nanoscale using sequential method, Numer. Heat Transfer: Part B: Fundam. 44 (5) (2003) 439–456.

[12] P.C. Tuan, M.C. Ju, The validation of the robust input estimation approach to two-dimensional inverse heat conduction problems, Numer. Heat Transfer: Part B: Fundam. 37 (2) (2000) 247–265.

[13] F. Scarpa, G. Milano, Kalman smoothing technique applied to the inverse heat conduction problem, Numer. Heat Transfer: Part B: Fundam. 27 (3) (1995) 291–307.

[14] M. Raudensky, K.A. Woodbury, J. Kral, T. Brezina, Genetic algorithm in solution of inverse heat conduction problems, Numer. Heat Transfer: Part B: Fundam. 28 (3) (1995) 293–306.

[15] M. Raudensky, J. Horsky, J. Krejsa, Usage of neural network for coupled parameter and function specification inverse heat conduction problem, Int. Commun. Heat Mass Transfer 22 (5) (1995) 661–670.

[16] J. Krejsa, K.A. Woodbury, J.D. Ratliff, M. Raudensky, Assessment of strategies and potential for neural networks in the inverse heat conduction problem, Inverse Problems Eng. 7 (1999) 197–213.

[17] H. Shiguemori Elcio, S. Da Silva José DemÍSio, F. De Campos Velho Haroldo, Estimation of initial condition in heat conduction by neural network, Inverse Problems Sci. Eng. 12 (3) (2004) 317–328.

[18] H. Deng, S. Guessasma, G. Montavon, H. Liao, C. Coddet, D. Benkrid, S. Abouddi, Combination of inverse and neural network methods to estimate heat flux, Numer. Heat Transfer: Part A: Appl. 47 (6) (2005) 593–607.

[19] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proceedings of the IEEE International Conference on Neural Networks, San Francisco, 1993, pp. 586–591.

[20] M.T. Hagan, B. Howard, M.H. Demuth, M.H. Beale, Neural Network Design, PWS Publishing Company, Boston, 2002, pp. 12–19.

[21] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey, 1998, pp. 239.

[22] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.

[23] R. Battiti, First and second order methods for learning: between steepest descent and Newton's method, Neural Comput. 4 (2) (1992) 141–166.

[24] M.J.D. Powell, Restart procedures for the conjugate gradient method, Math. Program. 12 (1977) 241–254.

[25] P.C. Tuan, C.C. Ji, L.W. Fong, W.T. Huang, Application of Kalman filtering with input estimation technique to on-line cylindrical inverse heat conduction problems, Int. J. Jpn. Soc. Mech. Eng. Ser. B 40 (1) (1997) 126–133.

[26] P.C. Tuan, C.C. Ji, L.W. Fong, W.T. Huang, An input estimation approach to on-line two dimensional inverse heat conduction problems, Numer. Heat Transfer: Part B: Fundam. 29 (1996) 345–363.

[27] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. USA 81 (10) (1984) 3088–3092.

[28] J.J. Hopfield, D.W. Tank, Computing with neural circuits: a model, Science 233 (4764) (1986) 625–633.

[29] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the backpropagation method, Biol. Cybernet. 59 (1988) 257–263.

[30] D.W. Patterson, Artificial Neural Networks, Prentice Hall, New Jersey, 1998, pp. 182.

[31] P. Antognetti, V. Milutonovic (Eds.), Neural Networks: Concepts, Applications and Implementations, vol. I, Prentice Hall, New Jersey, 1991, pp. 121–122.

[32] D.J.C. MacKay, Bayesian interpolation, Neural Comput. 4 (1992) 415–447.

[33] F.D. Foresee, M.T. Hagan, Gauss–Newton approximation to Bayesian learning, in: Proceedings of the 1997 International Joint Conference on Neural Networks, Houston, 1997, pp. 1930–1935.

[34] E. Kreyszig, Advanced Engineering Mathematics, seventh ed., Wiley, New York, 1993, pp. 648–649.

[35] H.S. Carslaw, J.C. Jaeger, Conduction of Heat in Solids, second ed., Oxford University Press, New York, 1959, pp. 203, 328–329.

[36] J.B. Gao, J.Z. Zhang, M.Q. Xu, Measuring one dimensional temperature field of rotor using Hopfield neural network, Thermal Turbine 32 (2) (2003) 78–81.