An Intelligent Approach Based on Artificial Neural Network (ANN) to Solve Sequencing Batch Reactor

Abd El-sameea M.[1], Zaki S.[2] , Abd El-wahed W.F.[3] and Ibrahim G.[4**]
1,2,4 Basic Engineering Sciences Dept -Faculty of Engineering - Menoufia Univ.
3- Faculty of Computer Sciences - Menoufia Univ.
** gamal_i_m_a@yahoo.com

Abstract

This paper presents an intelligent approach based on artificial neural networks (ANN) to obtain numerical solutions of the mathematical models of dynamical system, that are represented by ordinary differential equations (ODEs) with initial value representing a sequencing batch reactor. The intelligent approach consists of two phases. The first phase focuses on developing a simulation model of the given ODEs for obtaining an approximate solution. The second phase concentrated on linking the simulation model with a feedforward ANN containing adjustable parameters (weights), such that the output of the first phase is used as a target of the ANN. Hence, the applicability of this approach ranges from single ordinary differential equations (ODEs), to system of coupled ODEs. The intelligent approach is applied to solve a chemical application problem.

Keywords: Initial value problem; Artificial neural networks; Simulation; Backpropagation algorithm; Sequencing batch reactor.

1. Introduction

Most problems in science and engineering are represented by a set of differential equations (DEs) through the process of mathematical modeling. Analytical solutions of the developed mathematical models based on physical laws may not be always possible. In addition, analytical methods are generally inadequate to obtain closed form solutions of the considered problems. The most used numerical methods developed for solving DEs include finite difference method, finite element method (FEM), finite volume method and boundary element method (BEM) [1-12].

Enormous progress in the field of numerical solutions of ordinary differential equations (ODEs) has been developed in the last three decades. However, these methods are still based on some discretization of the domain of analysis into a number of finite elements.

Artificial neural networks are considered as approximation schemes where the input data for the design of a network consists of only a set of unstructured discrete data points. Different strategies were developed for solving ordinary differential equations (ODEs) using feedforward artificial neural networks (ANNs) [13–18].

The remainder of this paper is organized as follows: In sections 2,3, we introduce the proposed feedforward ANN structure and backpropagation training algorithm scheme. In sections 4,5, we introduce our problem definition and solution methodology. In section 6, the obtained results are also graphically presented and some conclusive remarks are given. All computer programs developed in this paper have been performed by using MATLAB. Finally, section 7, concludes the paper.

2. The feedforward artificial neural network

A typical feedforward neural network (FFNN) consists of a number of layers of neural units. Every neural unit in a layer is interconnected to every unit in the adjacent layers.The strength of every interconnection is characterized by its weight. Information propagates from the input layer (first layer) to the output layer (last layer). Each neural unit weights the input it receives from the units in the previous layer using the appropriate interconnection weight. Subsequently, the sum of the inputs weighted is filtered through a transfer function to produce an output from the unit . The outputs from the output layer represent the final prediction of the neural network Figure (1). Training of the neural network is done by systemically adjusting the interconnection weights to minimize the error such that the predicated output from the network is as close as possible to the desired output. Several methods are used to minimize the error; for example, it is possible to use either steepest descent (i.e. the back propagation algorithm or any of its variants), or conjugate gradient method .

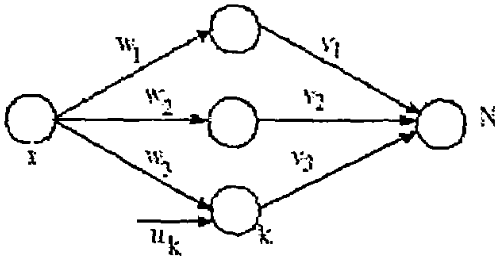

Fig.1. Neural network with one hidden layer

3. Learning of artificial neural networks

Learning of ANN is an algorithm for adjusting the network weights w_{ij} to Minimize the difference between the actual outputs O_j and the desired output T_j -We can define an error function to quantify this difference

$$E(W_{ij}) = \frac{1}{2}\sum_{p}\sum_{j}(T_j - O_j)^2 \qquad (1)$$

This is known as the total squared error summed over all output unit j and all training patterns p.

The aim of learning is to minimize this error by adjusting the weight W_{ij}.Typically we make a series of small adjustments to the weights $W_{ij} \rightarrow \mathrm{W_{ij}} + \Delta W_{ij}$ until the error $E(W_{ij})$ is 'small enough'

To minimize the error function (1) we use a series of gradient decent weight updates

$$\Delta W_{kl} = -\eta \frac{\partial E(W_{ij})}{\partial W_{kl}} \qquad (2)$$

If the transfer function for the output neuron is $f(x)$, and the activations of the previous layer of neurons are z_i, then the output are $O_j = f\left(\sum_i z_i w_{ij}\right)$ and

$$\Delta W_{kl} = -\eta \frac{\partial}{\partial W_{kl}}\left[\frac{1}{2}\sum_{p}\sum_{j}\left(T_j - f\left(\sum_i z_i w_{ij}\right)\right)^2\right] \qquad (3)$$

After repeated application of the chain rule, and some tidying up, we end up with a very simple weight update equation:

$$\Delta W_{kl} = \eta \sum_{p} (T_l - O_l) f'\left(\sum_{n} z_n w_{nl} \right) z_k \tag{4}$$

We use a differentiable activation function such as sigmoid function in the hidden layer $\sigma(z) = \frac{1}{1+e^{-z}}$.Due to the properties of the sigmoid derivative, the general weight update in equation (4) simplifies so that it only contain neuron activations and no derivative

$$\Delta W_{kl} = \eta \sum_{p} (T_l - O_l) O_l (1 - O_l).z_k \tag{5}$$

Equation (5) is known as the generalized Delta Rule for training for feedforward ANN with sigmoid activation function.

4. Sequencing batch reactor

Sequencing batch reactors (SBR) are basically fill-and-draw systems: wastewater is added and simultaneously treated during the fill period, it is treated during the react period , allowed to settle during the settle period and at the end of the cycle, the effluent is withdrawn during the withdraw period . A great deal of research effort has been spent on modeling the suspended-growth processes. Models, which are specifically oriented to SBR, are available in the literature [19], [20]. These models were able to predict the system performance with various degrees of success.

A model was developed to describe the performance of SBR is a system of ordinary differential equations.

The model presented here is structured upon four processes: substrate-associated growth process, product-associated growth process, hydrolysis process, and decay process. Three soluble components are considered in the model: soluble substrate,S_s, soluble intermediate product, P_s (both materials are considered degradable, however the rate of $P_s$degradation is slower than the rate of degradation ofS_s), and soluble inert material. Three particulate materials are presented : particulate organics, X_s, active biomass, X_a, and inert particulate organics, X_i. This model is developed for completely aerobic systems and does not include the nitrification and denitrification processes.

In the model all biomass (X_s, X_a, X_i) concentrations are based on total volume while two substrates and intermediate product (S_s, S_i, P_s) concentrations are based on bulk liquid volume. The distinction between bulk liquid volume and total volume is very essential as the concentration of solids in such systems is significant, thus the difference between the two volumes. This distinction is often neglected when modeling such systems and therefore, erroneous results may occur. The relationship between the bulk volume (V_b) and the total volume (V_t) is given by the following equation:

$$\frac{dV_b}{dt} = Q.\left(1.0 - \frac{X_{to}}{\rho_w}\right)$$

The model is presented in the following system of ODEs

Particulate organics, X_s:

$$\frac{dX_s}{dt} = A_1.(X_{sf} - X_s) + \left(1 - f_p\right)R_4 - R_3$$

Active biomass (X_a):

$$\frac{dX_a}{dt} = A_1.(X_{af} - X_a) + R_1 + R_2 - R_4$$

Inert particulate organics (X_i):

$$\frac{dX_i}{dt} = A_1.(X_{if} - X_i) + f_p.R_4$$

Soluble substrate (S_s):

$$\frac{dS_s}{dt} = A_5.S_{sf} - \frac{R_1}{(y_h.A_2)} - \beta.S_s$$

Soluble intermediate product (P_s):

$$\frac{dP_s}{dt} = A_5 P_{sf} + \frac{1}{A_2}\left(\frac{y_p}{y_h}R_1 - \frac{R_2}{y_h} + (1-\alpha)R_3\right) - \beta.P_s$$

Soluble inert substrate (S_i):

$$\frac{dS_i}{dt} = A_5.S_{if} + \frac{\alpha}{A_2}R_3 - \beta.S_i$$

Where

$$R_1 = \frac{\mu_m.(S_s - S^*)}{(S_s - S^*) + k_s} X_a \quad , \quad R_2 = \frac{\mu_m.P_s}{P_s + k_p} X_a$$

$$R_3 = \frac{k_h.(X_s / X_a)}{(X_s / X_a) + k_x} X_a \quad , \quad R_4 = \beta_h.X_a$$

$$\mu_m = \bar{\mu}_m \frac{\alpha_1}{\alpha_1 + P_s}$$

$$X_{to} = X_s + X_a + X_i \quad , \quad COD = S_s + P_s + S_i$$

$$\beta = \left(\frac{dV_b}{dt} \right) . \frac{1}{(V_t A_2)}$$

$$A_1 = \frac{Q}{V_t}; \quad \mathrm{A}_2 = 1.0 - \frac{X_{to}}{\rho_w}; \quad \mathrm{A}_3 = 1.0 - \frac{X_{tof}}{\rho_w}; \quad \mathrm{A}_4 = \frac{A_3}{A_2}; \quad \mathrm{A}_5 = A_1.A_4$$

The values of the various parameters used in this model are obtained from the literature and are listed in Table (1).

Table 1. Parameters used in the model.

Parameter	Value	Parameter	Value
β_h	0.052	f_p	0.08
K_s	20	K_p	500
K_h	0.0916	K_x	0.15
y_p	0.25	y_h	0.5
$\bar{\mu}_m$	0.4	α	0.025
α_1	60		

5. Solution methodology

In this paper, an intelligent methodology is presented to solve ODEs that relies on the function approximation capabilities of feedforward ANNs. The intelligent methodology consists of two phases. The first phase focuses on developing a simulation model of the given ODEs for obtaining an approximate solution. The second phase concentrated on linking the simulation model with a feedforward ANN with adjustable parameters (weights) such that the first phase is used as a target of the ANN to train it. Hence, we obtain ANN representing the solution of ODE. This feedforward ANN is with one hidden layer varying the neuron number in the hidden layer according to complexity of the considered problem. The ANN having an appropriate architecture that has been trained with backpropagation algorithm. The method uses feedforward neural networks with one hidden layer in which varies the neuron number as a basic approximation element, whose weights and biases are adjusted to minimize an error function. Optimization techniques requiring the calculation of the gradient of the selected error function w.r.t. the network parameter are used to train the network.

6. Computational analysis:

The initial value problem given by the equations (2)-(7) is sloved by Runge-Kutta method for multiple ordinary differential equations, then solved by proposed approach and compare them at different cases of fill and reaction time.

In the first phase of propoed approch we construct a simulation model of the given system of ODEs as shown in Figure(2) .

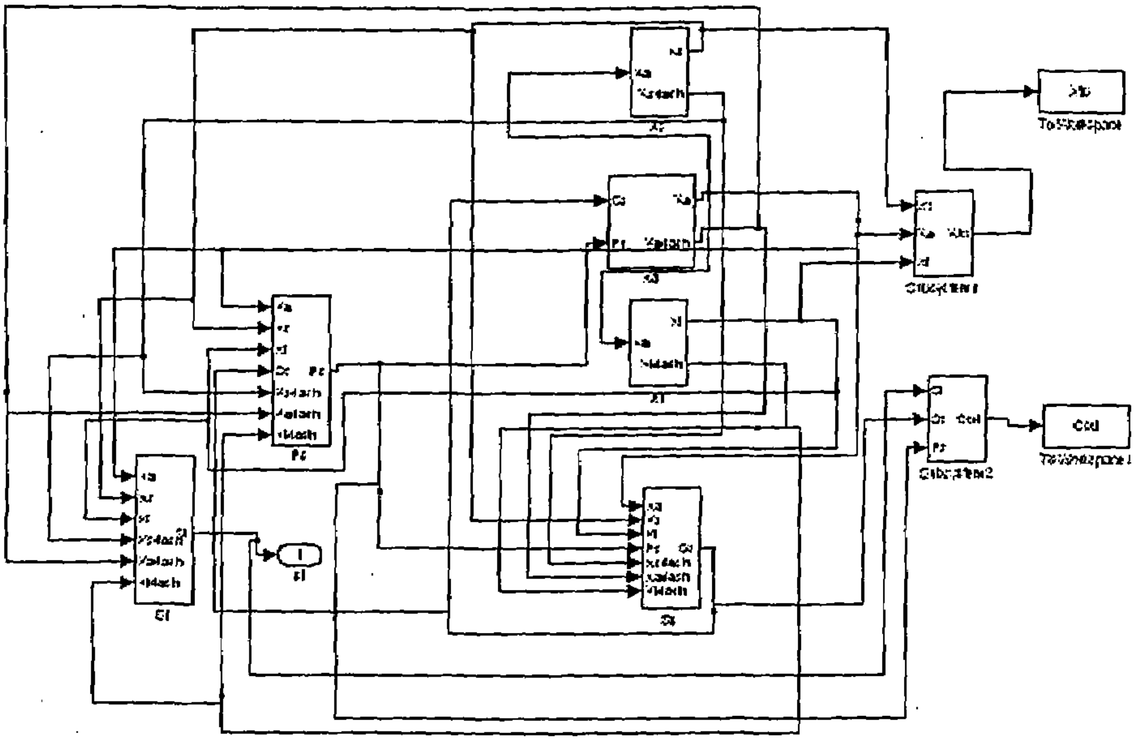

Fig. 2. Simulation of the sequencing batch reactor

Then we design a feedforward ANN that represent the solution of the system for different fill and reaction time ,all cases with COD loading 1580 mg/l .

Figures (3, 4, 5, 6, 7, 8) represent the solution comparison of the system by using Runge-Kutta method for multiple ODEs and neural solution, and we show that they coincide with each other.

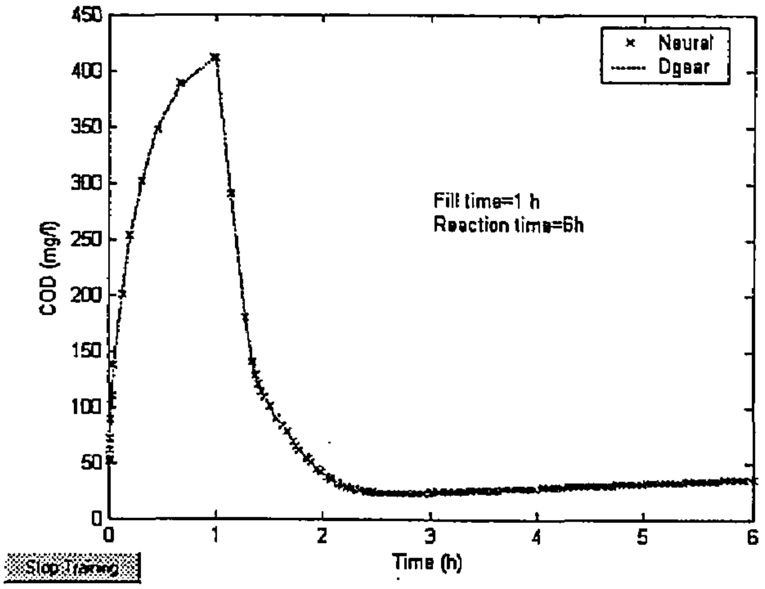

Fig. 3. Comparison of model solution with Dgear and ANN for COD concentration vs time(Fill time=1 h and react time =6 h)

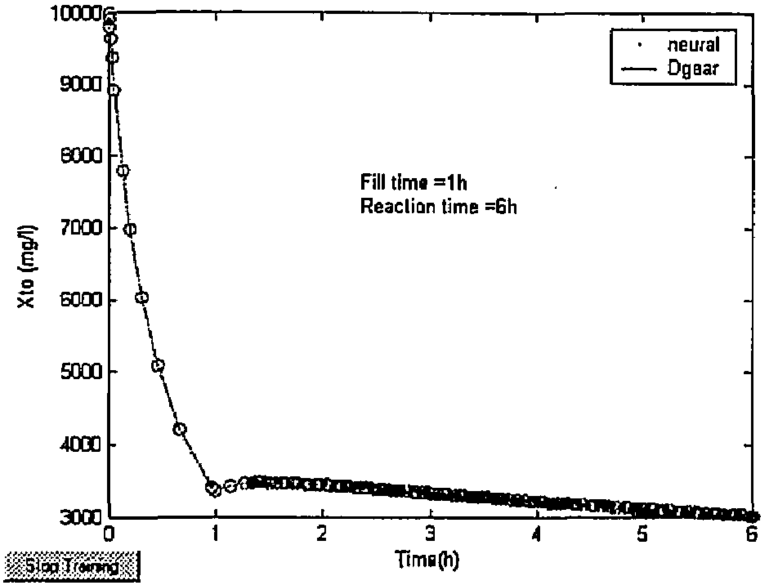

Fig. 4. Comparison of model solution with Dgear and ANN for total organic biomass X_{to} vs time(fill time =1h and react time=6h)

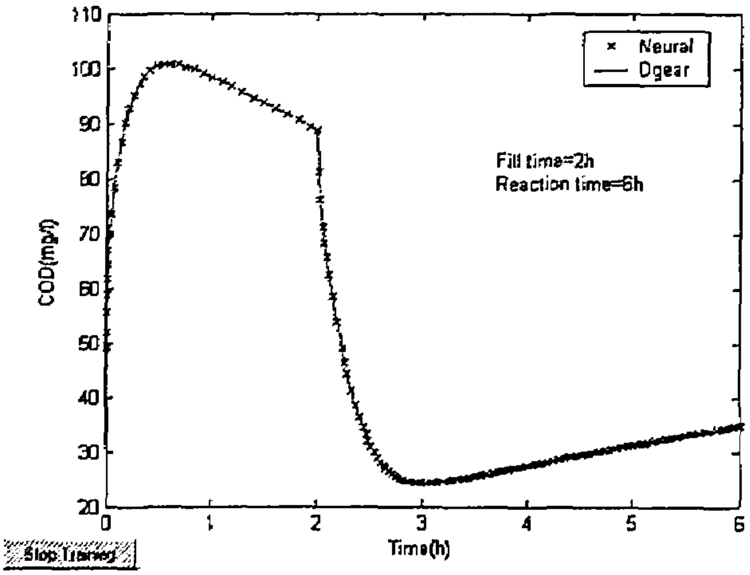

Fig. 5. Comparison of model solution with Dgear and ANN for COD concentration vs. time (fill time=2 h and react time = 6 h)

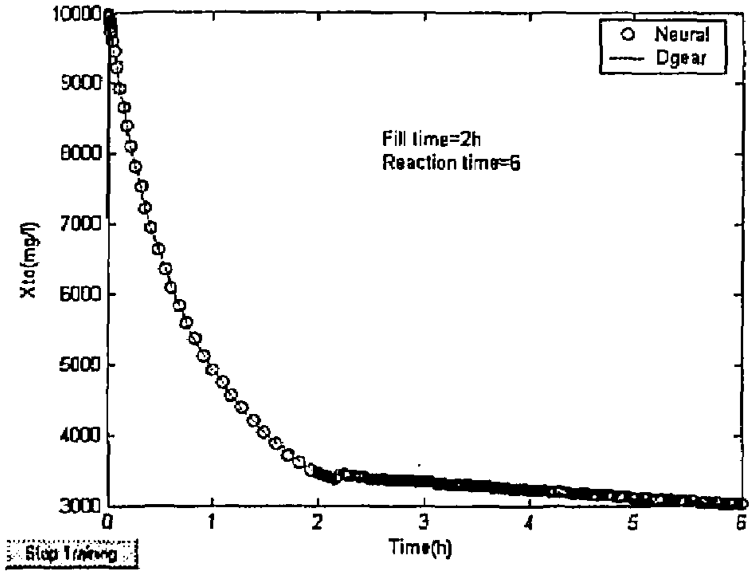

Fig. 6. Comparison of model solution with Dgear and ANN for total organic bimassX_{to} vs. time(fill time =2 h and react time=6 h)

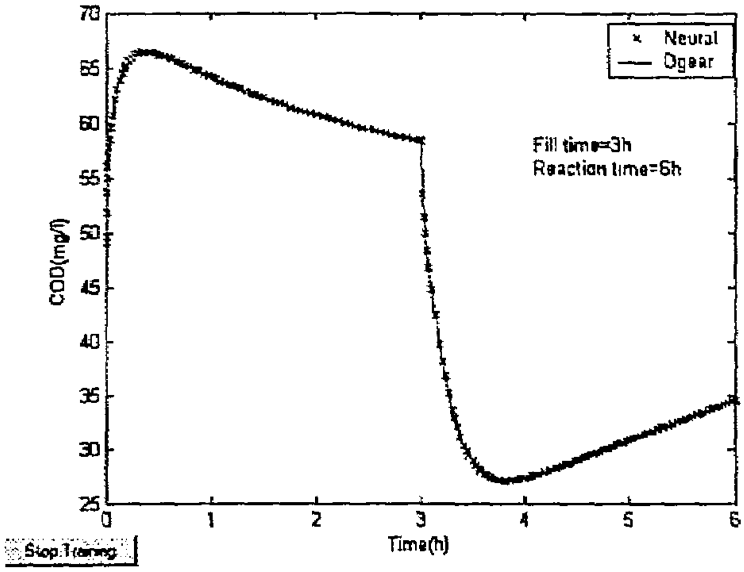

Fig. 7. Comparison of model solution with Dgear and ANN for COD concentration vs time(fill time =3 h and react time=6 h)

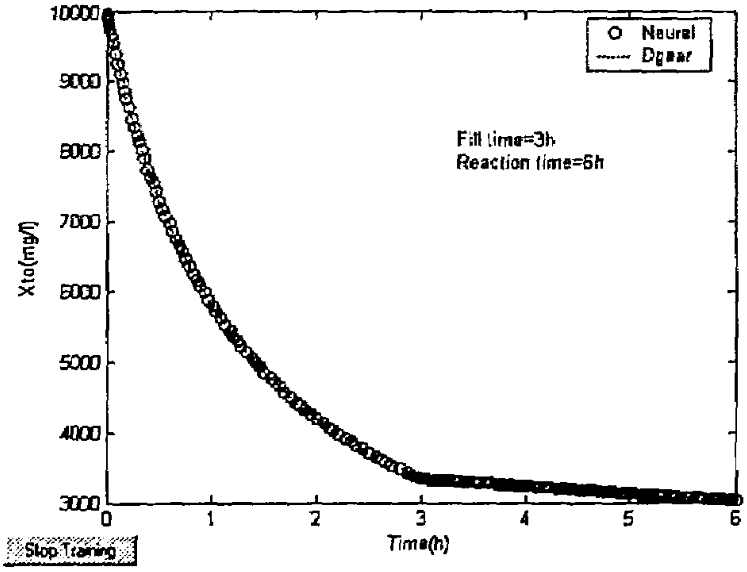

Fig. 8. Comparison of model solution with Dgear and ANN for total organic bimass X_{to} vs time(fill time =3 h and react time=6 h)

7. Conclusion

The dynamic systems are generally represented by either ODEs or PDEs.That is why, we propose an alternative method using feedforward ANNs to solve a system of ordinary differential equations (sequencing batch reactor). To test accuracy of this method, the problems are also solved by either Runge-Kutta or analytical methods.Then, the obtained results are graphically presented and compared with each other. Figures show that the results are in very close agreement. Further more,we test the method for training point and outside the training points to see approximate capability of the method for ODEs.

The architecture of the proposed by ANN consists of one hidden layer varying its neuron number to deal with highly non-linear problems. We successfully apply this method to problems whose dynamics are represented by ODEs (sequencing batch reactor).

Consequently, this method can be used for a wide class of linear and non-linear ODEs.Therefore, it is general and easy to apply for numerical solutions of dynamic problems.

References

[1] P.M. Lima, M.P. Carpentier, Iterative methods for a singular boundary-value problem, J. Comput. Appl. Math. 111 (1999) 173–186.

[2] H. Guoqiang, W. Jiong, K. Hayami, X. Yuesheng, Correction method and extrapolation method for singular two-point boundary value problems, J. Comput. Appl. Math. 126 (2000) 145–157.

[3] D. Kincaid, W. Cheney, Numerical Analysis, Brooks/Cole, Monterey, CA, 1991.

[4] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media, IEEE Trans. Antennas Propagation AP-14 (1996) 302–307.

[5] P. Hunter, A. Pullan, FEM/BEM Notes, Department of Engineering Science, The University of Auckland, New Zealand, 1997.

[6] M.A. Kolbehdari, M.S. Nakhla, M.N.O. Sadiku, Hybrid model of scatteringfrom eccentrically nested dielectric cylinders, J. Franklin Inst. 225B (1999) 43–51.

[7] G. Jacobsohn, A discrete Taylor series method for the solution of two-point boundary-value problems, J. Franklin Inst. 338 (2001) 61–68.

[8] Y. Guoyou, W.J. Mansur, J.A.M. Carrer, L. Gong, Stability of Galerkin and collocation time domain boundary element methods as applied to the scalar wave equation, Comput. Struct. 74 (2000) 495–506.

[9] L. Meirovitch, T.J. Stemple, A new approach to the modeling of distributed structures for control, J. Franklin Instit. 338 (2001) 241–254.

[10] W-S. Lee, Y-H. Ko, C-C. Ji, A study of an inverse method for the estimation of impulsive heat flux, J.Franklin Inst. 337 (2000) 661–671.

[11] J. Kouatchou, Parallel implementation of a high-order implicit collocation method for the heat equation, Math. Comput. Simul. 54 (2001) 509–519.

[12] B. Bialecki, G. Fairweather, Orthogonal spline collocation methods for partial differential equations , J. Comput. Appl. Math. 128 (2001) 55–82.

[13] T. Nguyen-Thien, T. Tran-Cong, Approximation of functions and their derivatives: a neural network implementation with applications, Appl. Math. Modell. 23 (1999) 687–704.

[14] S. He, K. Reif, R. Unbehauen, Multilayer neural networks for solving a class of partial differential equations, Neural Networks 13 (2000) 385–396.

[15] C.L. Karr, I. Yakushin, K. Nicolosi, Solving inverse initial-value, boundary-value problems via genetic algorithm, Eng. Appl. Artif. Intell. 13 (2000) 625–633.

[16] N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, Neural Networks 14 (2001) 185–199.

[17] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks 9 (5) (1998) 987–1000.

[18] I.E. Lagaris, A. Likas, D.G. Papageorgio, Neural-network methods for boundary value problems with irregular boundaries, IEEE Trans. Neural Networks 11 (5) (2000) 1041–1049.

[19] Artan N., Orhan D. and Beler Baykal B. (1990) Implication of the task group model-I. The effect of initial substrate concentration. Wat. Res. 24, 1251-1258.

[20] Ibrahim G. (1990) Modeling and simulation of SBR in wastewater treatment. Trans. Egyptian Soc. Chem. Engrs16, 136-150.