Neuro-Fuzzy Finite Element Method and a Posteriori Error Estimation

Hidetomo ICHIHASHI †, Tetsuya MIYOSHI † Kazunori NAGASAKA †
and
Yoshihiro SATOH †,

† Department of Industrial Engineering
College of Engineering, University of Osaka Prefecture
1-1 Gakuencho, Sakai, Osaka 593, Japan

Abstract

The finite element method(FEM) is a useful tool to solve partial differential equations with various boundary conditions in many engineering problems. In this paper, we propose a Neuro-Fuzzy finite element method with an iterative learning scheme, as an application of Gaussian radial basis function(RBF) networks, in which the finite elements are fuzzy subsets defined by Gaussian membership functions.

For almost all problems in practical engineering where FEM is applied, analytical solutions are not obtained. Since the finite element concept is based on the discretization of a continuum, the subject of developing reliable a posteriori error estimates is central to the effective use of FEM. In the FEM some error estimators which are reasonably accurate and useful in the existing problem have been proposed. The approximate solutions of Neuro-Fuzzy FEM also need some tools for error estimation. Hence, we propose a method of error estimation based on the same iterative learning scheme. Illustrative examples of solving Laplace equations are shown with the computational results of approximate solutions and estimated errors by using the proposed methods.

Key words: finite element method, radial basis function, Gaussian function, fuzzy set, error estimatior

I. Introduction

FOR approximating nonlinear mappings, especially interpolating the data points in a high dimensional space, multi-layers networks with adaptive learning algorithms are widely applied. Among others, the most popular networks consist of sigmoidal basis functions [13]. J.Moody and C.Darken [5] have proposed radial basis function(RBF) networks, a technique for interpolating in a high dimensional space, and reported that RBF networks are potentially 1000 times faster than the sigmoidal basis function networks with backpropagation for comparable error rates. T.Poggio and F.Girosi showed that regularization principles lead to approximation schemes which are equivalent to networks with one hidden layer [10, 11]. Gaussian RBFs have recently been applied to the various kinds of problems such as handwritten character recognition, speech recognition, view based object recognition[12], time series prediction and system optimization. The network can be regarded as a three-layered network (input, output and hidden layer) with Gaussian hidden units. Although closely related, these paradigms have been given such diverse names as "Gaussian potential functions"[4], "localized receptive fields"[5], "regularization networks"[11], and "locally tuned processing units"[6]. Since the output of the network is a linear combination of Gaussian functions, the network can be reinterpreted as fuzzy if-then rules [1–3, 7, 14–16]. Learning algorithms are relatively simple compared with sigmoidal neural networks with back propagation. Hence, the network can be applicable to more complicated and difficult problems with supervised or unsupervised learning.

In solving of a partial differential equation or a system of differential equations, the finite element method (FEM) is commonly used as a powerful and effective tool of numerical procedure. Though FEM has been improved through the development of digital computers, accuracy of the solutions using FEM largely depends on the selection of knots, and it is not easy to change the knots adaptively. In this paper, we propose a method for the approximate solution of partial differential equations as an application of Gaussian RBF networks. The finite elements in the FEM are replaced by fuzzy subsets defined by Gaussian membership functions, and an adaptive learning method is employed. We call this method "Neuro-Fuzzy FEM".

For almost all problems in practical engineering, in which FEM is applied, analytical solutions of the problems are not obtained. Since the accuracy of the approximate solution depends on the selection of the knots, an improper finite element grid produces significant errors in its solution. Hence, the approximate solution of FEM needs some tools for error estimation. For FEM, many error estimators have been presented which are not only reasonably accurate, but whose evaluation is computationally so simple that it can be readily implemented in existing finite element codes[8, 9, 17, 18].

In this paper, we also propose a method of error estimation for Neuro-Fuzzy FEM which is based on the same adaptive learning scheme. As an illustrative example, we show computational results of the approximate solution of Laplace differential equations and some estimated solution errors.

II. Fuzzy Finite Element Defined by Gaussian Membership Function

Let $A_{i,k}$ denote the membership function of the k-th fuzzy rule in the domain of the i-th input variable x_i. The k-th rule is written as:

"If x_1 is $A_{1,k}$ and x_2 is $A_{2,k}$ ... and x_n is $A_{n,k}$ then y is w_k."

The conclusion part of the fuzzy reasoning rule which infers the output y is simplified as real number w_k. The final output(y) is written as:

$$y = \sum_k w_k \cdot \mu_k = \sum_k w_k \cdot (\prod_i A_{i,k}(x_i)) \qquad (1)$$

where μ_k is the compatibility degree of the premise part of the k-th fuzzy rule, which is computed with the algebraic product operator. This model is a simplified fuzzy model[1–3, 7]. In the case of Gaussian membership function, $A_{i,k}$ is defined as:

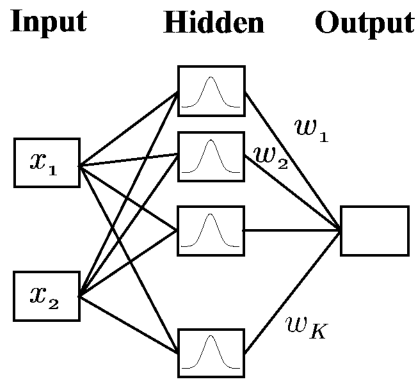

Fig. 1. A neural network with Gaussian hidden units

$$A_{i,k}(x_i) = \exp\left(-\frac{(x_i - a_{i,k})^2}{b_{i,k}}\right) \qquad (2)$$

where the parameters $a_{i,k}$ and $b_{i,k}(i = 1, \cdots, n)$ are given for each k and are changed in the training procedure. μ_k has the form of Gaussian function $F(\| x - a \|)$ and y in Eq.(1) is its linear combination, *i.e.* the network of Gaussian radial basis functions[5, 10–12]. The model has three layers which are fully interconnected with the adjacent layers, but there is no interconnection within each layer (See Figure 1).

The triangular finite elements in the FEM are replaced by fuzzy subsets whose membership functions are Gaussian functions. In other words, the shape function (the basis function) is Gaussian. We call this method "Neuro-Fuzzy FEM". No finite element grids are required in this method. Since the approximate function consists of Gaussian functions, it is infinitely differentiable. Hence, any partial differential equations and boundary conditions can be approximated using y in Eq.(1). Let us consider a potential problem governed by the Laplace equation:

$$\nabla^2 f(x_1, x_2) = 0 \qquad on \quad R \qquad (3)$$

where R is a domain subject to the boundary conditions:

$$f(x_1, x_2) = g(x_1, x_2) \qquad on \quad B_1 \qquad (4)$$

$$\frac{\partial f(x_1,x_2)}{\partial \boldsymbol{n}} = h(x_1,x_2) \qquad on \qquad B_2 \qquad (5)$$

where $B = B_1 \bigcup B_2$ is the boundary of R and $\boldsymbol{n}$ is the unit vector of the outward normal to B at the boundary point. The function $g(x_1,x_2)$ and $h(x_1,x_2)$ are prescribed. The Laplace equation is approximated using y in Eq.(1) as:

$$\nabla^2 y(x_1,x_2) = 0 \qquad on \qquad R \qquad (6)$$

$$y(x_1,x_2) = g(x_1,x_2) \qquad on \qquad B_1 \qquad (7)$$

$$\frac{\partial y(x_1,x_2)}{\partial \boldsymbol{n}} = h(x_1,x_2) \qquad on \qquad B_2 \qquad (8)$$

$\partial^2 y/\partial x_i^2$ in Eq.(6) can be written as:

$$\frac{\partial^2 y}{\partial x_i^2} = \sum_k \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}}{b_{i,k}^2} \cdot w_k \cdot \mu_k \right) \qquad (\text{for} \quad i = 1,2) \quad (9)$$

Let the left-hand sides of the Eqs.(6)$\sim$(8) be $F(\boldsymbol{x})$, $G(\boldsymbol{x})$ and $H(\boldsymbol{x})$ respectively. Then, the cost function to estimate the unknown functions y (i.e. the unknown parameters w_k, $a_{i,k}$ and $b_{i,k}$ of Eq.(1)), which is the squared error function, can be described as:

$$\begin{aligned} E(\boldsymbol{x}) &= \frac{1}{2} \cdot \{F(\boldsymbol{x})^2 + K_1 \cdot (G(\boldsymbol{x}) - g(\boldsymbol{x}))^2 \\ &\quad + K_2 \cdot (H(\boldsymbol{x}) - h(\boldsymbol{x}))^2\} \qquad (10) \end{aligned}$$

where K_1 and K_2 are positive constants. We formulated a nonlinear mathematical programming problem to find the values of parameters $a_{i,k}$, $b_{i,k}$ and w_k in the Neuro-Fuzzy model by minimizing the cost function of Eq.(10). The learning rule based on the Least Mean Square(LMS) is

$$\begin{aligned} w_k^{NEW} &= w_k^{OLD} - \alpha \cdot \delta_{w_k} \qquad (11) \\ a_{i,k}^{NEW} &= a_{i,k}^{OLD} - \beta \cdot \delta_{a_{i,k}} \qquad (12) \\ b_{i,k}^{NEW} &= b_{i,k}^{OLD} - \gamma \cdot \delta_{b_{i,k}} \qquad (13) \end{aligned}$$

where α, β and γ denote the positive learning constants corresponding to w_k, $a_{i,k}$ and $b_{i,k}$ respectively. In this training procedure, we calculate the cost function of Eq.(10) and δ_w, δ_a and δ_b at all data points chosen uniformly in the region R and the boundary B. These δ_w, δ_a and δ_b are represented as:

$$\begin{aligned} \delta_C &= F(\boldsymbol{x}) \cdot \frac{\partial F(\boldsymbol{x})}{\partial C} \\ &\quad + K_1 \cdot (G(\boldsymbol{x}) - g(\boldsymbol{x})) \cdot \frac{\partial G(\boldsymbol{x})}{\partial C} \\ &\quad + K_2 \cdot (H(\boldsymbol{x}) - h(\boldsymbol{x})) \cdot \frac{\partial H(\boldsymbol{x})}{\partial C} \\ &\quad C \in \{w_k, a_{i,k}, b_{i,k}\} \qquad (14) \end{aligned}$$

For instance, $\partial F(\boldsymbol{x})/\partial C$ in Eq.(14) are

$$\begin{aligned} \frac{\partial F(\boldsymbol{x})}{\partial w_k} &= \sum_{i=1}^{2} \left(\frac{4(x_i - a_{i,k})^2 - 2b_{i,k}}{b_{i,k}^2} \right) \\ &\quad \cdot \mu_k \qquad (15) \\ \frac{\partial F(\boldsymbol{x})}{\partial a_{i,k}} &= \left(\frac{8(x_i - a_{i,k})^3 - 12(x_i - a_{i,k})b_{i,k}}{b_{i,k}^3} \right. \\ &\quad + \frac{8(x_i - a_{i,k})(x_j - a_{j,k})^2}{b_{i,k} b_{j,k}^2} \\ &\quad \left. - \frac{4(x_i - a_{i,k})}{b_{i,k} b_{j,k}} \right) \cdot \mu_k \cdot w_k \qquad (16) \\ \frac{\partial F(\boldsymbol{x})}{\partial b_{i,k}} &= \left(\frac{4(x_i - a_{i,k})^4 - 10(x_i - a_{i,k})^2 b_{i,k}}{b_{i,k}^4} \right. \\ &\quad + \frac{2}{b_{i,k}^2} + \frac{4(x_i - a_{i,k})^2 (x_j - a_{j,k})^2}{b_{i,k}^2 b_{j,k}} \\ &\quad \left. - \frac{2(x_i - a_{i,k}^2)}{b_{i,k}} \right) \cdot \mu_k \cdot w_k \qquad (17) \end{aligned}$$

where the subscript j is 2 when $i = 1$ and $j = 1$ when $i = 2$.

III. A Posteriori Error Estimator with Neuro-Fuzzy Approach

For almost all problems in practical engineering, FEM is widely applied because the analytical solutions are not obtained. The FEM is based on the concept that the continuous region is partitioned with finite elements, so it is necessary to estimate how accurate the approximate solution of the FEM is, and some tools for the error estimation are required.

At each data point (i.e. collocation) chosen uniformly in the region R and the boundary B, the errors of the approximate function y in Eqs.(6)–(8) are represented by the following equations respectively.

$$\begin{aligned} E_1^{after} &= \nabla^2 f(x_1,x_2) - \nabla^2 y(x_1,x_2) \\ &= \nabla^2 \{f(x_1,x_2) - y(x_1,x_2)\} \\ &= \nabla^2 e(x_1,x_2) \qquad on \quad R \qquad (18) \end{aligned}$$

$$
\begin{aligned}
E_2^{after} &= f(x_1,x_2) - y(x_1,x_2) \\
&= e(x_1,x_2) \qquad on \quad B_1 \qquad (19)
\end{aligned}
$$

$$
\begin{aligned}
E_3^{after} &= \frac{\partial f(x_1,x_2)}{\partial \boldsymbol{n}} - \frac{\partial y(x_1,x_2)}{\partial \boldsymbol{n}} \\
&= \frac{\partial \{f(x_1,x_2) - y(x_1,x_2)\}}{\partial \boldsymbol{n}} \\
&= \frac{\partial e(x_1,x_2)}{\partial \boldsymbol{n}} \qquad on \quad B_2 \qquad (20)
\end{aligned}
$$

where E_1^{after}, E_2^{after} and E_3^{after} are the errors of the Laplace equation and the errors of the boundary conditions on B_1 and B_2 respectively. $f(x_1,x_2)$ is the analytical solution and $e(x_1,x_2)$ is a function which represents the error between the approximate solution and the analytical solution. Our aim is to estimate this error function $e(x_1,x_2)$. Eqs.(18)–(20) can be approximated as:

$$\nabla^2 y_e(x_1,x_2) \approx E_1^{after} \qquad on \quad R \qquad (21)$$

$$y_e(x_1,x_2) \approx E_2^{after} \qquad on \quad B_1 \qquad (22)$$

$$\frac{\partial y_e(x_1,x_2)}{\partial \boldsymbol{n}} \approx E_3^{after} \qquad on \quad B_2 \quad (23)$$

where $y_e(x_1,x_2)$ is the output of RBF networks for estimating the error of the approximate original solution obtained by the "Neuro-Fuzzy FEM". The structure of RBF networks is the same as shown in Figure 1. Representing the left-hand sides of Eqs. (21)–(23) as $F_e(\boldsymbol{x})$, $G_e(\boldsymbol{x})$, and $H_e(\boldsymbol{x})$ respectively , the cost function is defined as:

$$
\begin{aligned}
E_e(\boldsymbol{x}) &= \frac{1}{2} \cdot \{(F_e(\boldsymbol{x}) - E_1^{after})^2 \\
&\quad + K_3 \cdot \left(G_e(\boldsymbol{x}) - E_2^{after})\right)^2 \\
&\quad + K_4 \cdot \left(H_e(\boldsymbol{x}) - E_3^{after})\right)^2 \} (24)
\end{aligned}
$$

where K_3, K_4 are the positive constants.

The solution method of this error estimation problem is the same as in the original Neuro-Fuzzy FEM.

IV. Numerical Examples

In this section, we apply the above proposed Neuro-Fuzzy FEM to solve a partial differential equation with various boundary conditions and to estimate the errors of the approximate solution obtained by the Neuro-Fuzzy FEM. First, we take two examples of Laplace equations with the Dirichlet condition and Neumann condition, which can be solved analytically (Case 1). Second, we take three examples of Laplace equations with discontinuous boundary conditions which cannot be solved analytically (Case 2). Lastly, we calculate the errors of the approximate solution by using the proposed errors estimator and compare it with the error between the approximate solution and the analytical solution.

Case 1

(1) The Dirichlet condition:

$$
\begin{aligned}
\nabla^2 f(x_1,x_2) &= 0 \\
& \quad \text{on} \quad R = \{-1.0 \le x_1 \le 1.0, \\
& \qquad\qquad 0 \le x_2 \le \pi\}
\end{aligned}
$$

the boundary condition

$$
f(x_1,x_2) = \begin{cases} 0 & (x_2 = 0\ , \\ & \ x_2 = \pi) \\ \cosh 1.0 \sin x_2 & (x_1 = -1.0, \\ & \ x_1 = 1.0) \end{cases}
$$

basis functions	3×6
collocations	11×21

(2) The Neumann condition:

$$
\begin{aligned}
\nabla^2 f(x_1,x_2) &= 0 \\
& \quad \text{on} \quad R = \{0 \le x_1 \le 1.0, \\
& \qquad\qquad 0 \le x_2 \le 1.0\}
\end{aligned}
$$

the boundary conditions

$$f(x_1,x_2) = 0 \quad (x_2 = 0,\ x_2 = 1.0)$$

$$\frac{\partial f(x_1,x_2)}{\partial x_1} = \begin{cases} 0 & (x_1 = 0) \\ \sin \pi x_2 & (x_1 = 1.0) \end{cases}$$

basis functions	3×6
collocations	11×21

Case 2

(1)

$$
\begin{aligned}
\nabla^2 f(x_1,x_2) &= 0 \\
&\text{on} \quad R = \{0 \le x_1 \le 1.0, \\
&\qquad\qquad 0 \le x_2 \le 1.0\}
\end{aligned}
$$

the boundary condition

$$
f(x_1,x_2) = \begin{cases} 1.0 & (x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases} \tag{25}
$$

basis functions	6×3
collocations	21×11

(2)

$$
\begin{aligned}
\nabla^2 f(x_1,x_2) &= 0 \\
&\text{on} \quad R = \{0 \le x_1 \le 1.0, \\
&\qquad\qquad 0 \le x_2 \le 1.0\}
\end{aligned}
$$

the boundary condition

$$
f(x_1,x_2) = \begin{cases} 1.0 & (x_1 \in [0.1,0.3] \bigcup [0.7,0.9], \\ & \ x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases}
$$

basis functions	6×3
collocations	21×11

(3)

$$
\begin{aligned}
\nabla^2 f(x_1,x_2) &= 0 \\
&\text{on} \quad R = \{0 \le x_1 \le 1.0, \\
&\qquad\qquad 0 \le x_2 \le 1.0\}
\end{aligned}
$$

the boundary condition

$$
f(x_1,x_2) = \begin{cases} 1.0 & (x_1 \in [0, 0.5], \\ & \ x_2 = 0) \\ 0.0 & (\text{otherwise}) \end{cases} \tag{26}
$$

basis functions	6×3
collocations	21×11

The data points (collocations) are chosen uniformly in the region R and B, and the initial positions of the Gaussian functions are chosen as shown in Figure 2. In Case 1 and Case 2, the number of collocations is 11×21 and the number of the Gaussian functions is 3×6.

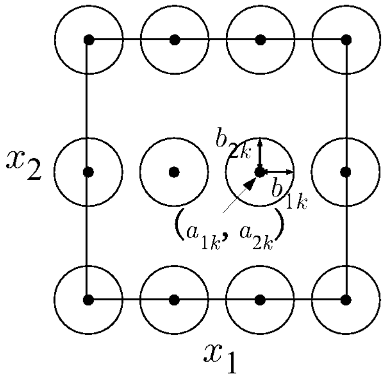

Fig. 2. Initial positions of Gaussian functions. 4×3 Gaussian functions are used.

The values of K_1 and K_2 in Eq.(10) are set to 30 for all problems. The computational results of Case 1 –(1) and (2) are shown in Figures 3 and 5 respectively. The computational results of Case 2 –(1),(2) and (3) are shown in Figures 7, 8 and 9 respectively. The change in values of the cost function as learning proceeds, is shown in Table. 1. Errors between the approximate solutions and the analytical solutions in Case 1 and Case 2 are shown in Figures 4 and 6 respectively.

Figures 10 and 11 show the estimated errors of the approximate solution in Case 1– (1) and (2) obtained by the proposed error estimator. Comparing Figures 10 and 11 with Figures 4 and 6 respectively, it can be seen that the errors between the analytical solution and the approximate solution are estimated to some extent.

By adding the estimating error y_e to the approximate solution y ,we can get a more accurate solution.

V. Conclusion

In this paper, we have proposed a neuro-fuzzy solution method to partial differential equations. Since the finite element is fuzzified by Gaussian membership function and an adaptive learning scheme is adopted , we call it the Neuro-Fuzzy Finite Element Method. The major difficulty of the proposed FEM is to determine the learning constants and the weights of the cost function. These must be determined by trial and error. The proposed error estimator can be a helpful tool for deciding

Table 1. The change in values of the cost function as learning proceeds.

Number of iterations	0	1000	5000
Case1-(1)	1367.45 (1367.45)	11.3844 (0.0665)	–
Case1-(2)	300.00 (300.00)	37.9031 (3.5509)	–
Case2-(1)	630.00 (630.00)	60.5376 (7.2092)	–
Case2-(2)	300.00 (300.00)	70.1851 (16.5188)	58.9153 (12.0638)
Case2-(3)	330.00 (330.00)	89.3439 (21.7600)	63.8277 (16.2115)

Note: () indicates the error for the boundary condition.

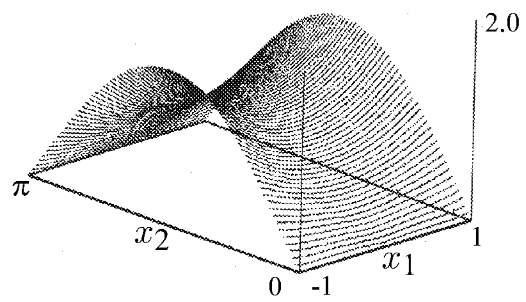

Fig. 3. Approximate solution of the Laplace equation with the Dirichlet condition, after 1000 learning iterations. (Case 1 – (1))

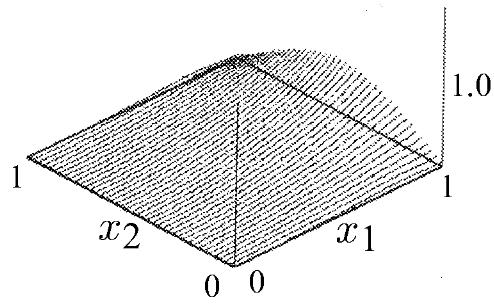

Fig. 5. Approximate solution of the Laplace equation with the Neumann condition, after 5000 learning iterations. (Case 1 – (2))

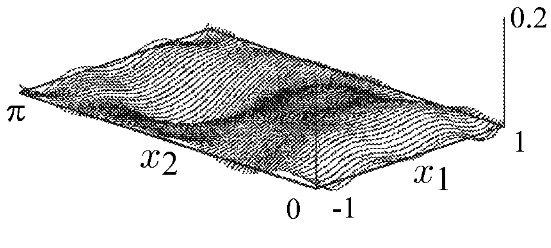

Fig. 4. Errors between the approximate solution and the analytical solution. (Case 1 – (1))

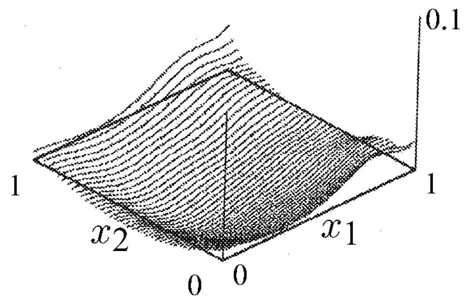

Fig. 6. Errors between the approximate solution and the analytical solution. (Case 1 – (2))

these parameters values. An another disadvantage is that it requires long computational time. However, with the development of high speed computers, Neuro-Fuzzy FEM can be a convenient solution method of partial differential equations. Neuro-Fuzzy FEM has the following advantages:

1. As the adaptive fuzzy meshing is provided, no finite element grid is needed.
2. It requires simpler algorithms and a smaller memory size.
3. Accuracy of the solution can be improved with the increase of collocations and basis functions.
4. A method of error estimation based on the same scheme is realized.

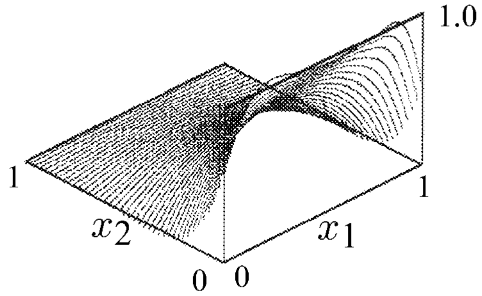

Fig. 7. Approximate solution of the Laplace equation with the boundary condition of Eq.(25), after 1000 learning iterations. (Case 2 – (1))

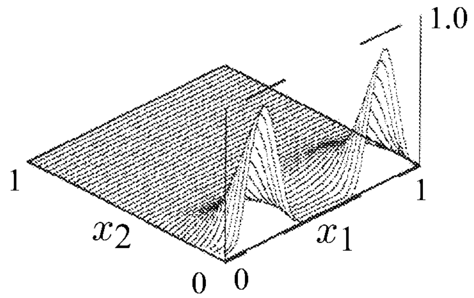

Fig. 8. Approximate solution of the Laplace equation with the boundary condition of Eq.(26), after 1000 learning iterations. (Case 2 – (2))

References

[1] H.Ichihashi, "Iterative fuzzy modeling and a hierarchical network," *Proc. of IFSA 4th World Congress, Belgium,* pp.49-52, 1991.

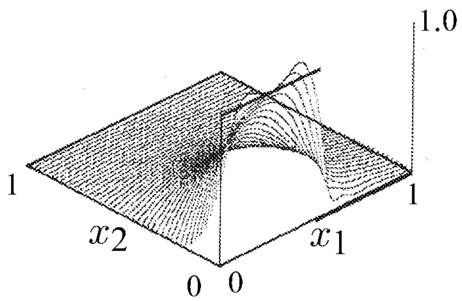

Fig. 9. Approximate solution of the Laplace equation with the boundary condition of Eq.(26), after 1000 learning iterations. (Case 2 – (3))

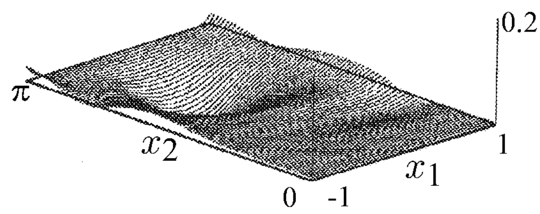

Fig. 10. Estimating error of the approximate solution in Case 1 – (1), after 1000 learning iterations.

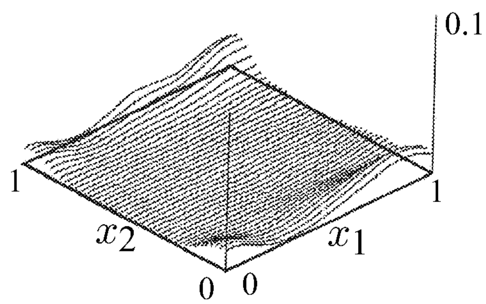

Fig. 11. Estimating error of the approximate solution in Case 1 – (2), after 1000 learning iterations.

[2] H.Ichihashi, "Learning inverse dynamics model of a manipulator in a hierarchical fuzzy model," *Proc. IMACS/SICE RM^2 '92, Kobe, september,* pp.41-46, 1992.

[3] H.Ichihashi and I.B.Turksen, "A Neuro-Fuzzy approach to data analysis of pairwise comparisons," *Int. J. Approximate Reasoning,* Vol.9, pp.227-248, 1993.

[4] S.Lee and R.M.Kil, "A Guassian potential function network with hierarchically self-organizing learning," *Neural Networks,* Vol.4, No.2, pp.207-224, 1991.

[5] J.Moody and C.Darken, "Learning with localized receptive fields," *Proc. of 1988 Connectionist Models Summer School, Morgan-Kaufmann,* pp.133-143, 1988.

[6] J.Moody and C.Darken, "Fast learning networks of locally-tuned processing units," *Neural Computation,* Vol.1, No.2, pp.281-294, 1989.

[7] K.Nagasaka, H.Ichihashi and R.Leonard, "Neuro-Fuzzy GMDH and its application to modeling grinding characteristics," *Int. J. of Production Reserch* (to appear).

[8] H.Ohtsubo and M.Kitamura, "Element

by element a posteriori error estimation and improvement of stress solutions for two-dimensional elastic problems," *Int. J. of Numerical Methods in Engineering,* Vol.29, pp.223-244, 1990.

[9] H.Ohtsubo and M.Kitamura, "Numerical investigation of element-wise a posteriori error estimation in two and three dimensional elastic problems," *Int. J. of Numerical Methods in Engineering,* Vol.34, pp.969-977, 1992.

[10] T.Poggio and F.Girosi, "A theory of networks for approximation and learning," *A.I. Memo No.1140, C.B.I.P. paper,* No.31, 1989.

[11] T.Poggio and F.Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Sciences,* Vol.247, pp.978-982, 1990.

[12] T.Poggio and S.Edelman, "A network that learns to recognize 3D objects," *Nature,* Vol.343, pp.263-266, 1990.

[13] D.E.Rumelhart, J.L.McClelland and the PDP Research Group, "Parallel Distributed Processing," *(Cambridge, MA: MIT Press),* 1987.

[14] M.Sugeno and G.T.Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems,* Vol.28, pp.15-33, 1988.

[15] L.A.Zadeh, "Fuzzy algorithms," *Information Control,* Vol.12, pp.94-102, 1968.

[16] L.A.Zadeh, "Outline of a new approach to analysis of complex systems and decision processes," *IEEE Trans. on System Man and Cybernetics,* Vol.SMC-2, pp.28-44, 1973.

[17] O.C.Zienkiewicz and J.Z.Zhu, "A simple error estimator and adaptive procedure for practical engineering analysis," *Int. J. of Numerical Methods in Engineering,* Vol.24, pp.337-357, 1987.

[18] O.C.Zienkiewicz and J.Z.Zhu, "Error estimates and adaptive refinement for plate bending problem," *Int. J. of Numerical Methods in Engineering,* Vol.28, pp.2839-2853, 1989.