COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING, Vol. 10, 195–201 (1994)

NEURAL-NETWORK-BASED APPROXIMATIONS FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

M. W. M. G. DISSANAYAKE AND N. PHAN-THIEN
Department of Mechanical Engineering, The University of Sydney, Sydney, N.S.W. 2006, Australia

SUMMARY

A numerical method, based on neural-network-based functions, for solving partial differential equations is reported in the paper. Using a 'universal approximator' based on a neural network and point collocation, the numerical problem of solving the partial differential equation is transformed to an unconstrained minimization problem. The method is extremely easy to implement and is suitable for obtaining an approximate solution in a short period of time. The technique is illustrated with the aid of two numerical examples.

1. INTRODUCTION

Continuum mechanics problems often lead to a set of non-linear partial differential equations (PDEs) together with a set of boundary conditions. The non-linearities are due to either the material responses or the nature of the boundary conditions. Spatial numerical techniques for solving these problems, which include finite-difference, finite-element and boundary-element methods, have been well developed and well understood. A concise summary of some of the methods are given in Johnson.[1]

Needless to say, the numerical implementation of any method is not always a straightforward task: there are issues involving the element scheme, the solver and the mesh generation that have to be resolved. Yet, in some cases, due largely to the exploratory nature of the research, we would like to have quickly available a reasonably accurate solution of the set of PDEs at hand without having to decide on various issues. The neural-network-based approximation is one such method that can deliver a good-quality solution with minimal effort from the user. In this paper we describe the method and illustrate it by way of a linear and a non-linear problem.

2. PROBLEM FORMULATION

We will now consider the following problem:

$$\mathscr{L}u = f, \quad x \in \Omega \tag{1}$$

$$\mathscr{B}u = g, \quad x \in \partial\Omega \tag{2}$$

where Ω is a bounded open domain with boundary $\partial\Omega$, f is a given function, $\mathscr{L}$ and $\mathscr{B}$ are some differential operators. The problem may be multidimensional, but there is no need to further classify the problem beyond this point. A so-called strong solution of (1) and (2) is an element of some underlying space $\mathscr{V}$ which satisfies both the PDE and the boundary conditions.

CCC 0748–8025/94/030195–07

Received 14 August 1992
Revised 27 August 1993

In a traditional method of solution, say the finite-element, the domain is replaced by a partition of non-overlapping elements, and we search for an approximate solution in a finite-dimensional space consisting of, say, piece-wise-linear functions. The original equations, (1) and (2), are abandoned in favour of their equivalent variational forms, or weighted residual forms; the precise description of the process can be found in Johnson.[1]

However, if there is a 'universal' approximator in this underlying space, that is, a function that has enough parameters and can be arbitrarily close to any element of the space by a judicious choice of the parameters, then this approximator can be a good choice for the approximate solution sought for. The closeness of two functions is described by the concept of denseness in $\mathscr{V}$. For a more mathematical definition, see Hornik *et al.*[2]

Suppose $u_a(x, \beta_i)$ is such a universal approximator in $\mathscr{V}$, where β_i represents the parameters, then one can look for an approximate solution

$$u = u_a(x, \beta_i) \tag{3}$$

The numerical problem is therefore reduced to an unconstrained optimization problem: given the objective function

$$h = \int_\Omega \| \mathscr{L}u_a - f \|^2 \, \mathrm{d}V + \int_{\partial\Omega} \| \mathscr{B}u_a - g \|^2 \, \mathrm{d}S \tag{4}$$

find the values of β_i that minimize h.

Another possibility, which will not be considered in this paper, is to work with the weak formulation of (1–2), and use u_a as both the test function and the approximate solution in the spirit of the finite-element method.

It remains now to find a good universal approximator. A multilayer feedforward network,[3] sometimes referred to as a neural network, provides such a function. A schematic diagram of a multilayer feedforward network is given in Figure 1. The network comprises an interconnected layer of computational elements known as neurons. A neuron has multiple inputs and a single output. Within a neuron, each input is weighted and combined (also suitably biased) to produce a single value. This value is then operated on by a transfer function. The output value of a neuron, therefore takes the form

$$o = \phi\left(\sum_{i=1}^{k} w_i x_i + b_i\right) \tag{5}$$

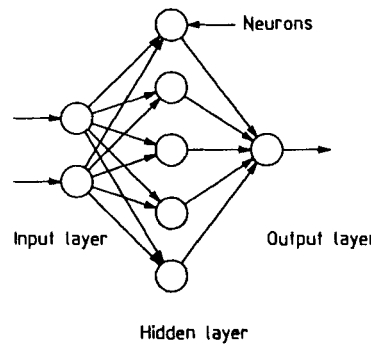


Figure 1. Schematic diagram of a multilayer feedforward network

where k is the number of inputs, the x_i represent the input values, and w_i and b_i are the weights and the bias associated with the neuron.

The transfer function ϕ is generally a one-dimensional non-linear monotonic function that can be easily evaluated. In the numerical examples presented in this paper, a sigmoidal function given below was used for this purpose:

$$o = \frac{1}{1 + \exp\left(- \frac{1}{\sum_{i=1}^{k} w_i x_i + b_i} \right)} \tag{6}$$

Outputs of certain neurons are used as inputs to other neurons in the network. The network, therefore, provides a non-linear mapping of the inputs to the outputs. Hornik *et al.*[2] have given rigorous mathematical proof to show that such networks can be used to approximate any Borel measurable function, provided there are a sufficient number of neurons, layers and interconnections. A neural network with input as x and output as u may, therefore, be used as the approximate solution proposed in (3), where weights w_i and biases b_i associated with all neurons combine to form β_i.

It is now possible to compute $\mathscr{L}u$ and $\mathscr{B}u$, in closed form, in terms of x. We propose to use point collocation and enforce (4) at selected locations in Ω and $\partial\Omega$. The resulting objective function may then be minimized to find values for w_i and b_i – hence the approximate solution to (1) and (2).

3. NUMERICAL EXAMPLES

To illustrate the technique proposed in the preceding Section, we present the following two numerical examples.

3.1. Example I: A linear Poisson equation

In order to test the accuracy and efficiency of the new technique, we consider the following two-dimensional inhomogeneous partial differential equation:

$$\nabla^2 u = \sin(\pi x)\sin(\pi y) \tag{7}$$
$$0 \leqslant x \leqslant 1$$
$$0 \leqslant y \leqslant 1$$

with $u = 0$ along the whole boundary.

The exact solution can be found to be

$$u = -\frac{1}{2\pi^2} \sin(\pi x)\sin(\pi y) \tag{8}$$

The problem was solved by approximating the potential u with neural networks with configurations 2-3-3-1 (i.e. two input nodes followed by two layers of three nodes each and one output node), 2-5-5-1 and 2-10-10-1. Equations (7) and the boundary conditions were enforced at nodes of 5×5, 10×10 and 20×20 meshes covering the domain $0 \leqslant x \leqslant 1$ and $0 \leqslant y \leqslant 1$. The objective function given in (4) was minimized using a quasi-Newton method. Finite-difference gradients were used during the minimization. Selection of the network

configuration is somewhat arbitrary and was done based on our previous experience with neural networks.

As an index to examine the effectiveness of these alternative schemes, we use the norm of the error of the solution N_e, defined as

$$N_e = \sqrt{\left[\frac{\sum_{i=1}^{N}(u_{ei}^2 - u_i^2)}{\sum_{i=1}^{N} u_{ei}^2}\right]} \tag{9}$$

where u_i and u_{ei} are the calculated and exact potentials at the grid point i, and N is the total number of the grid points. In calculating N_e, grid points of a uniform 20×20 mesh were used.

Figure 2 shows N_e as a function of the number of iterations used in the minimization of (4), for the 2-5-5-1 neural network and 10×10 mesh. A comparison of the minimum value of N_e achieved for all the test conditions is given in Table I. It is clear from these results that the

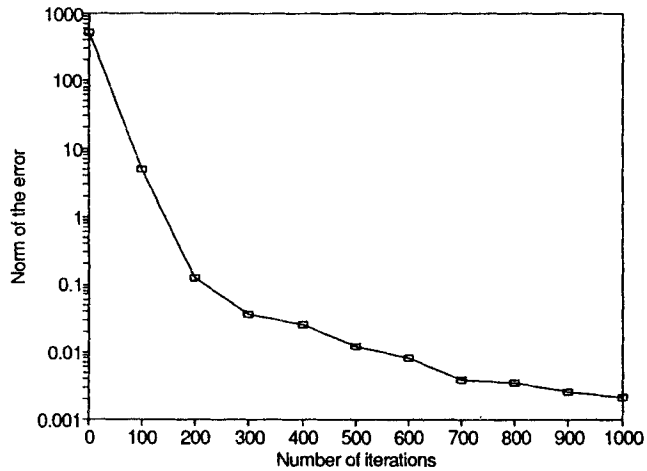


Figure 2. Graph of N_e against number of iterations

Table I. Effect of mesh size on minimum norm N_e for different network sizes

Network size	Mesh size	N_e
2-3-3-1	5×5	3·43 E − 2
	10×10	2·87 E − 2
	20×20	1·16 E − 2
2-5-5-1	5×5	2·42 E − 2
	10×10	7·03 E − 4
	20×20	6·57 E − 4
2-10-10-1	5×5	2·40 E − 3
	10×10	1·40 E − 4
	20×20	1·21 E − 4

proposed technique yields accurate answers. As expected, increasing the number of variables in the approximation as well as increasing the mesh size increase the accuracy of the solution. It also appears, however, that for the present example differences in the accuracy of the solutions are not significant except for the 2-3-3-1 configuration and the 5×5 mesh size.

3.2. Example II: Thermal conduction with non-linear heat generation

As an example of the application of the new method to non-linear problems, consider the steady-state temperature distribution in a homogeneous solid with non-linear heat generation. The temperature field can be described by the following partial differential equation:

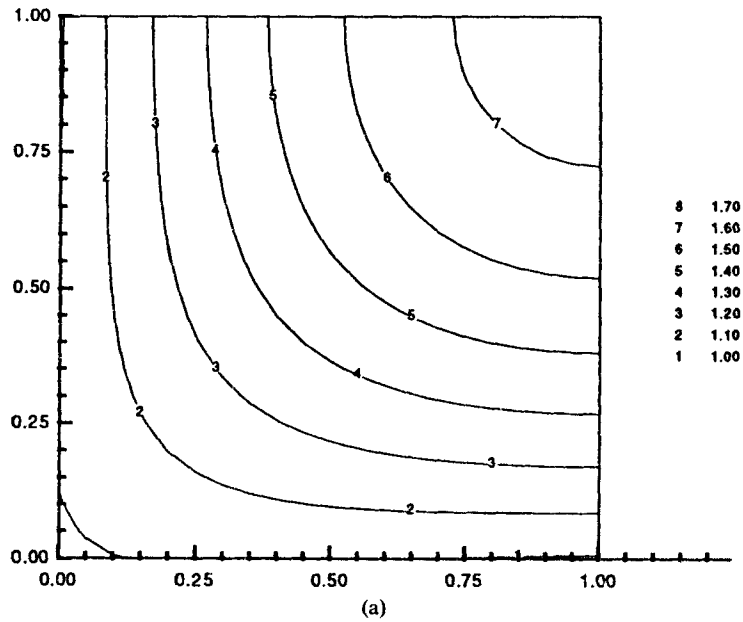
$$\nabla^2 T + f(T) = 0, \quad \text{in } \Omega \tag{10}$$

The domain Ω is taken to be a square section of side unity: $x \in [0, 1], y \in [0, 1]$. The boundary conditions are: the edges $x = 0$ and $y = 0$ are maintained at temperature of unity ($T = 1$), while the edges $x = 1$ and $y = 1$ are maintained adiabatically (i.e. $\partial T/\partial x = 0$ at $x = 1$ and $\partial T/\partial y = 0$ at $y = 1$).

The proposed technique is applicable to any general bounded function $f(T)$, but we shall use the method for a particular case in which the heat generation follows an exponential law such as $f(T) = \varkappa e^T$. It has long been known that in some technical applications the exponential law for non-linear heat generation holds.[4] The parameter $\varkappa$ expresses the ratio of heat generation to heat conduction. It may also be temperature-dependent. For the sake of simplicity, we set $\varkappa = 0{\cdot}5$ in the calculation.

Contours of the predicted temperature field obtained with a 2-5-5-1 neural network and 7×7, 10×10 and 20×20 meshes are shown in Figure 3.

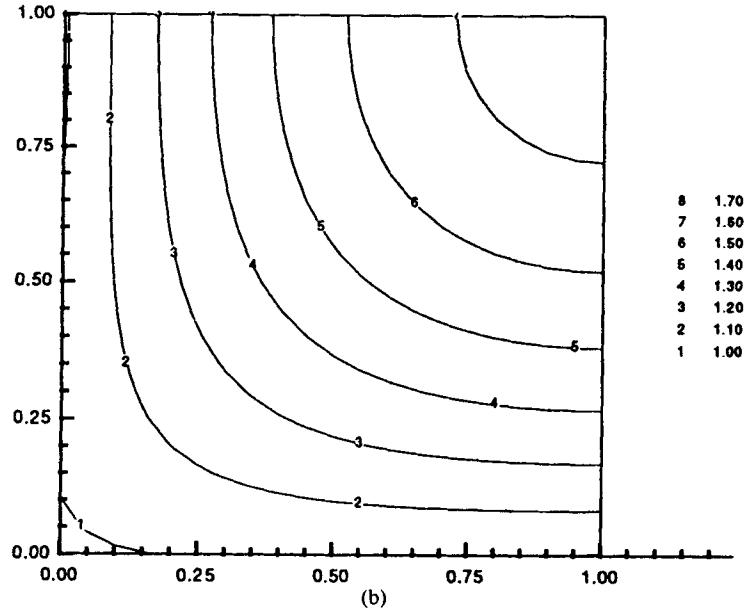
In order to estimate the accuracy of the solutions obtained we take the result reported in



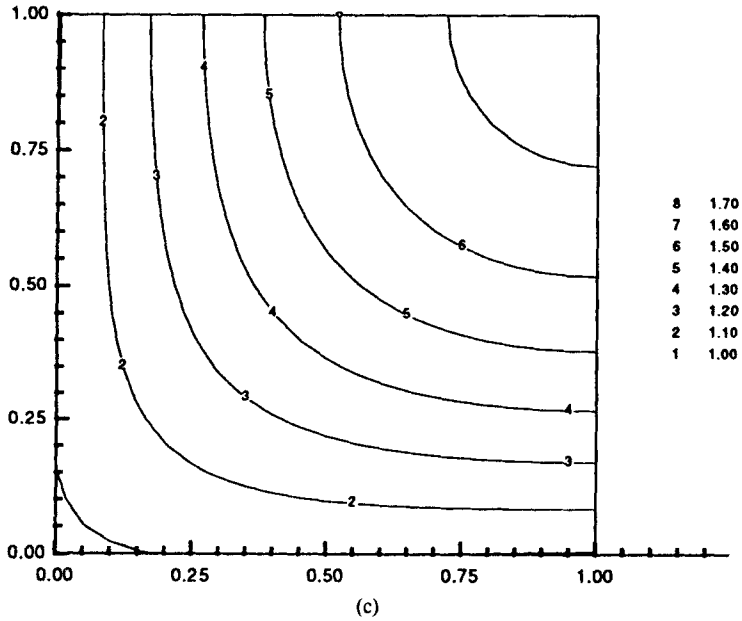
(a)

Figure 3. Temperature contours for (a) 7×7 mesh, (b) 10×10 mesh and (c) 20×20 mesh

Reference 5, for the finest discretization (21×21 mesh), and that shown in Figure 4 as the exact one, and measure the error of the calculations with the coarser meshes relative to this result. The norm of the error calculated based on (9) for all mesh sizes (7×7, 10×10 and 20×20) was less than 10^{-5}, which indicates that the results obtained are accurate. The method did not converge for a mesh of 5×5.



(b)



(c)

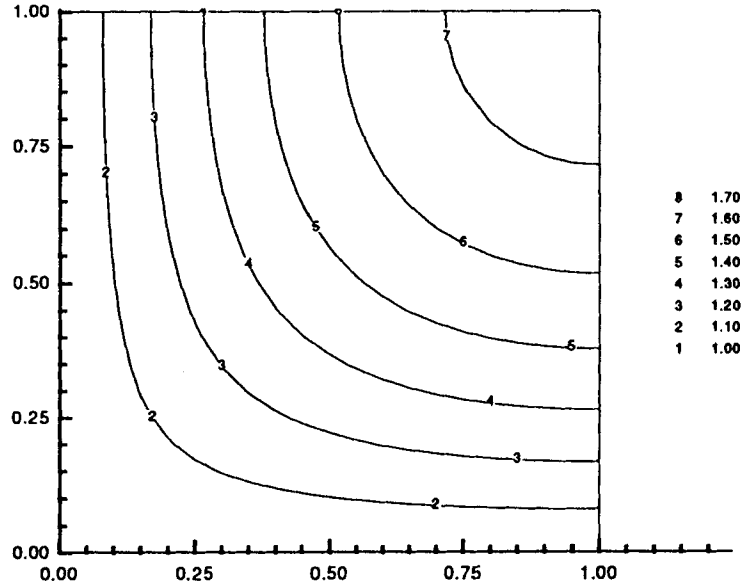


Figure 4. Temperature contours reported in Reference 5

4. CONCLUSIONS

In this paper we have proposed a method based on artificial neural networks for approximate solutions to partial differential equations. The method requires a minimal amount of effort to implement and produces a good-quality solution within a reasonable amount of time. The results of the numerical examples presented indicate the feasibility of using the proposed technique for solving inhomogeneous as well as non-linear problems.

REFERENCES

1. C. Johnson, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.
2. K. Hornik, M. Stinchcombe and H. White, 'Multilayer feedforward networks are universal approximators', *Neural Networks*, **2**, 359–366 (1989).
3. D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning representations by error propagation', in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland (Eds.), Vol. 1, MIT Press, Cambridge, 1986, pp. 318–362.
4. A. Lippke, 'Analytical solution and sine function approximations in thermal conduction with non-linear heat generation', *J. Heat Tranfer*, **113**, 5–11 (1991).
5. R. Zheng and N. Phan-Thien, 'Transforming the domain integrals to the boundary using approximate particular solutions: a boundary element approach for non-linear problems', *Appl. Numer. Math.*, **10**, in press.