Design of Locally Connected CMOS Neural Cells to Solve the Steady-State Heat Flow Problem

D. Gobovic and M. E. Zaghloul
Department of Electrical Engineering and Computer Science
The George Washington University
Washington, DC 20052
Telephone: (202) 994-3772
e-mail: zaghloul@seas.gwu.edu

Abstract - A locally connected CMOS neural network is used to solve the steady-state heat flow problem. The proposed theory is based on the formulation of an energy function where the solution of the problem is obtained at the minimum of the function. An electrical circuit that tends to find the minimum of the energy function is designed. The circuit has a cellular neural structure where each cell is implemented as a neuron. The proposed method is illustrated by an example of heat flow in the plate. The circuit simulation results agree with the numerical results obtained by the software program Matlab.

Introduction

This paper is concerned with the solution of a system of partial differential equations. Such problems with appropriate boundary and initial conditions occur in a number of engineering areas. Finite element method is usually used to analyze such systems which requires extensive computation for large systems. In this work, unlike other cellular neural network structures, the partial differential equations of the system under consideration are formulated as an energy function. The minimum of the energy function yields the solution of the problem. The technique is implemented by locally connected neurons. In the following sections, the theoretical background is first introduced then the heat flow problem is considered.

Theoretical Formulation

To derive our approach to the solution of partial differential equations, we will consider a two-dimensional equation:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \qquad (1)$$

defined on region R where u(x,y) is a continuous unknown scalar function which satisfies some given boundary conditions $u(\Omega)=\Phi$ at the boundary Ω of region R, f(x,y) is a given function, and x and y are space variables.

The unknown function u(x,y) can be approximated by a set of values u(i,j) (u_{ij} is also used for short notation in this text) at points P_{ij}. At each interior grid point P_{ij}, the partial derivatives of u(x,y) with respect to x and y in equation (1) can be replaced by difference equations:

$$\frac{\partial^2 u(x,y)}{\partial x^2} \text{ at } P_{ij} \doteq \frac{u(i,j-1) + u(i,j+1) - 2\,u(i,j)}{h^2}$$
$$\frac{\partial^2 u(x,y)}{\partial y^2} \text{ at } P_{ij} \doteq \frac{u(i-1,j) + u(i+1,j) - 2\,u(i,j)}{h^2} \qquad (2)$$

Partial differential equation (1) is then approximated by a set of linear equations:

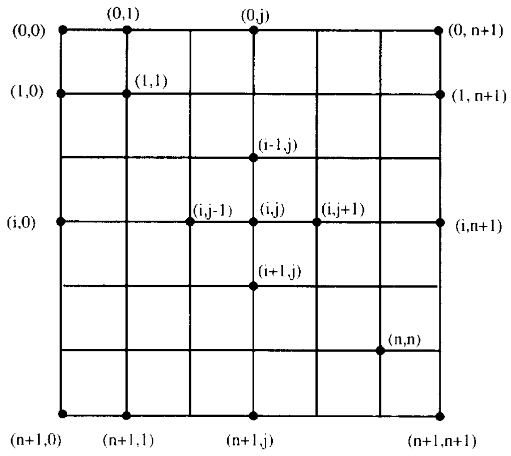

Figure 1: Neighboring mesh points

$$-u(i-1,j) - u(i+1,j) - u(i,j-1) - u(i,j+1) + 4\,u(i,j) + h^2 f(i,j) = 0 \qquad (3)$$
$$i,j = 1, \ldots, n$$

System (3) is a set of $m = n^2$ equations in the m unknowns u(i,j), i,j = 1, . . . , n. The values of u(i,j) at boundary points are assumed to be given by the boundary conditions:

$$u(i,0) = \Phi(P_{i,0}), \quad u(i,n+1)) = \Phi(P_{i,n+1}),$$
$$u(0,j) = \Phi(P_{0,j}), \quad u(n+1,j) = \Phi(P_{n+1,j}). \qquad (4)$$

where i,j = 1, 2, . . . , n. Then system (3) may be written as:

$$\mathbf{A}\mathbf{u} + h^2\mathbf{F} + \mathbf{b} = \mathbf{0} \qquad (5)$$

where **F** denotes a vector of the given function values f(i,j) at mesh points P_{ij}, and **b** is the vector that includes boundary conditions. The system matrix **A** is an $n^2 \times n^2$ symmetric, positive definite, block tri-diagonal matrix.

Define an energy function as:

$$E(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T \mathbf{A}\,\mathbf{v} + \mathbf{v}^T \varphi \qquad \text{where} \qquad \varphi = \mathbf{b} + h^2\,\mathbf{F} \qquad (6)$$

Due to the special structure of matrix **A**, function E(**v**) is uniformly convex and has a unique minimum in **v** [1] assuming φ is continuous and bounded. A neural network [2], [3] which tries to minimize the above energy function E(**v**) is designed such that:

$$\frac{du_{ij}}{dt} = -\frac{\partial}{\partial v_{ij}} E(v_{11}, v_{12}, \dots, v_{1n}, v_{21}, \dots, v_{n1}, \dots, v_{nn}) \tag{7}$$

where u_{ij} is the input of the ij-neuron in the network and $v_{ij} = g_{ij}(u_{ij}) = u_{ij}$ is its output. Thus:

$$\frac{du_{ij}}{dt} = -\sum_{kl} a_{ij,kl}\, v_{kl} - \varphi_{ij}$$
$$v_{ij} = u_{ij}, \quad i,j = 1, \dots, n \tag{8}$$

where $a_{ij,kl}$ is an element of matrix **A** at the intersection of ij row and kl column.

Because of the special structure of matrix **A**, a neural network that has a cellular structure can be built. The proposed structure is locally connected as in [4], [5], however, our approach and proposed circuit is different. With the simple weight template, a cellular structured circuit to solve the partial differential equation (1), can be designed. A possible circuit structure with local weights is illustrated in the next sections.

Problem of the Steady-State Heat Flow

As an example, a classical problem of the steady-state heat flow in a plate is considered [6]. Such a process is described by the Poisson's equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \tag{9}$$

where u(x,y) denotes the temperature at the point (x,y) and f(x,y) is the heat source or forcing term. The f(x,y) term models the gain in heat from a source (a flame, for example) or loss in heat from a sink (a refrigerator). The temperature distribution also depends on the boundary conditions (conditions at the edges of the plate). A simple physical plate is shown in Fig. 2.

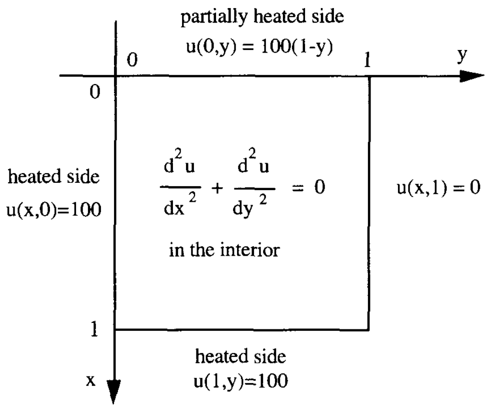

Fig. 2 Laplace's equation model for a heated plate

The temperature distribution inside the plate is modeled by the Laplace's equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{10}$$

Suppose that the temperature in the plate at the start is a room temperature. Thus, the temperature must change and eventually reach a new steady state which reflects the new given condition. The equation that models this process is the heat equation:

$$\frac{du}{dt} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{11}$$

The problem solution u(x,y,t) is now a function of time. The initial condition:

$$u(x,y,0) = 20 \tag{12}$$

models the initial room temperature (20°C in our example). In the new steady state du/dt decays to zero and the solution of equation (11) decays to the solution of equation (10). The steady state is not dependent on time t.

To find the steady state solution of this problem we will apply the proposed method. The (x,y) domain of the plate is replaced by a rectangular mesh with the mesh size h, as shown in Fig. 3.

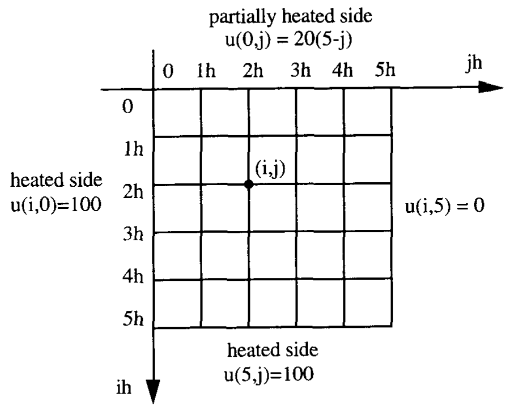

Fig. 3 Discretization of a plate domain by a rectangular grid of points

Difference formulas (2) are used to replace partial differentials in equation (10). This creates a difference equation at each grid point. A set of 16 equation in 16 unknowns u(i,j) for i,j=1,...,4 is obtained. This set of difference equations can be written in the matrix form:

$$\mathbf{A}\,\mathbf{u} + \boldsymbol{\varphi} = \mathbf{0} \tag{13}$$

Vector φ incorporates the given boundary conditions and for this particular problem is given by:

$$\varphi = (180\ 60\ 40\ 20\ 100\ 0\ 0\ 0\ 100\ 0\ 0\ 0\ 200\ 100\ 100\ 100)^t$$

The elements of vector φ take in account boundary temperatures expressed in °C.

To solve the above set of equations, we simulated a 4x4 cellular neural network shown in Fig. 4 where the boundary conditions are applied with the scaling factor of 20 (i.e., 100°C corresponds to 5V). The initial state of each cell is set to u_{ij} = 1V (equivalent to 20°C). An implementation of the network cell (i,j) is proposed in Fig. 5

The dynamic equation that describes the (i,j) cell of Fig. 5 is:

$$C\frac{du_{ij}}{dt} + \frac{u_{ij} - v_{i-1,j}}{R} + \frac{u_{ij} - v_{i+1,j}}{R} + \frac{u_{ij} - v_{i,j-1}}{R} + \frac{u_{ij} - v_{i,j+1}}{R} = I_{ij}$$
$$v_{ij} = u_{ij} \tag{14}$$

Equation (14) can be written for each (i,j) cell, i,j = 1, 2, . . . , n (n=4 in our example). A set of equations for the entire cellular neural network is:

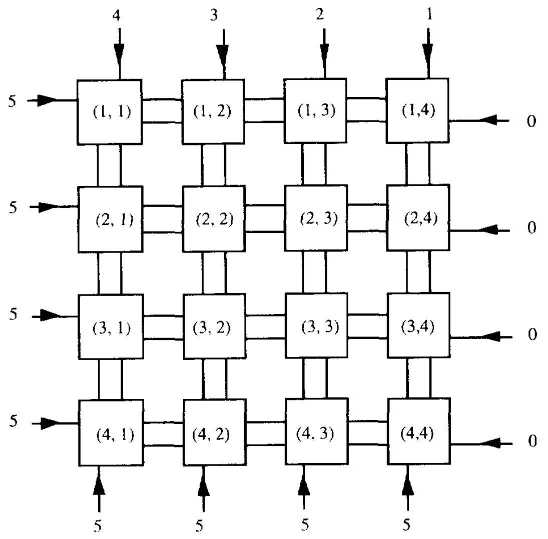

Fig. 4 4x4 Cellular neural network for the heat plate example

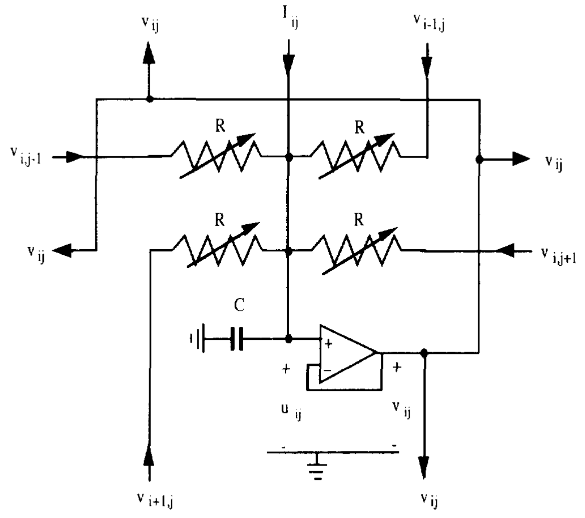

Fig. 5 An example of a cell circuit

$$\tau \frac{du_{ij}}{dt} = -4\, u_{ij} + v_{i-1,j} + v_{i+1,j} + v_{i,j-1} + v_{i,j+1} + R\, I_{ij}$$

$$v_{ij} = u_{ij} \quad , \quad i,j = 1, 2, \ldots, n \tag{15}$$

where $\tau = RC$ is the time constant of the circuit.

Equation (15) has the form described earlier by equation (8) where $RI_{ij} = -\varphi_{ij}$. Equation (8) converges to the steady state, and therefore equation (15) will converge too. In the steady state $du_{ij}/dt = 0$ and $v_{ij} = u_{ij}$, so that the set of steady-states u_{ij} gives the desired solution of equation (13).

In our simulation of the considered example, the selected circuit parameters for an (i,j) cell (given in Fig. 5) are R = 100kΩ and C = 1pF. The circuit time constant is τ = 25ns. The Spice simulation of the circuit is performed, and the obtained steady-states of the cellular network are shown in Fig. 6. The steady-state values of the circuit voltages, u_{ij}, represent the scaled steady-state temperatures of the given plate at points (i,j). The obtained results match to the solutions of equation (13) obtained by MatLab [7].

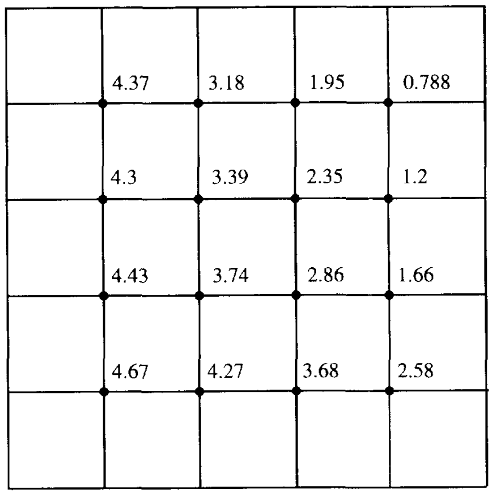

Fig. 6 Steady-states of the 4x4 cellular neural network

In this example, we have illustrated how the proposed neural network with the locally connected cells can be used in solving the two-dimensional problem of the heat flow which is described by the partial differential equation.

CONCLUSION

In this paper, a cellular neural-type network is developed for solving a class of partial differential equations. The proposed method is based on the energy function which is defined so that the minimum of the function is the solution of the set of partial difference equations that approximate the original partial differential equation. The design procedure of the neural network which finds the minimum of the energy function is given. The topological structure of the neural network is locally connected and suitable for the VLSI implementation. The proposed method is illustrated on the example of the steady-state heat flow in a plate. A possible implementation of the neural network cell is given, and the SPICE simulation of the network is performed. The obtained results agree with the numerical computation.

The proposed approach is general and can be used for solving other classes of differential equations.

REFERENCES

[1] J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970, pp. 110-115.

[2] J. M. Zurada, *Introduction to Artificial Neural Systems,* West Publishing Company, 1992.

[3] D. W. Tank. J. J. Hopfield, "Simple 'Neural' Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Transactions on Circuits and Systems*, Vol. CAS-33, No. 5, May 1986, pp. 533-541.

[4] L. O. Chua, L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 10, October 1988, pp. 1257-1272.

[5] L. O. Chua, L. Yang, "Cellular Neural Networks: Applications," *IEEE Transactions on Circuits and Systems*, Vol. 35, No. 10, October 1988, pp. 1273-1290.

[6] J. Rice, *Numerical Methods, Software, and Analysis: IMSL Reference Edition*, McGraw-Hill, 1983.

[7] *MATLAB for MS-DOS Personal Computers, User's Guide*, The MATH WORKS, Inc.

[8] W. H. Hayt, Jr., *Engineering Electronics*, McGraw-Hill, 1989, pp. 200-204.