# Analog Cellular Neural Network With Application to Partial Differential Equations With Variable Mesh-Size

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

## BACKGROUND

A design of VLSI analog CMOS circuit for solving partial differential equations (PDE) was presented in [1]. Elliptic PDE with constant coefficients were considered. In [1] a numerical method of discretizing the differential equations on n by n mesh is applied. The partial derivatives are approximated by the difference quotients. When these are substituted back into differential equation, a linear system of algebraic equations is obtained. The $n^2$ unknowns of this system are the solutions of the discretized differential equation at the mesh points. The resultant coefficient matrix of this linear system of equations is positive definite. A neural network approach is used. Due to a very regular structure and sparsity of the coefficient matrix, the electrical circuit of the neural network results into a cellular structure which is very suitable for the VLSI analog CMOS implementation.

In solving PDE by the use of finite-difference approximations an inherent error due to the discretization technique exists, Intuitively, the accuracy of any finite-difference solution could be increased by decreasing the mesh size. However, the nature of the very solution of the PDE has also an effect upon the inherent accuracy of a finite-difference approximation. Therefore, in selecting an appropriate mesh size, the nature of the function that constitutes the solution of PDE must be taken into consideration. In this paper we will consider how, within the limited size of the neural network, the accuracy of the introduced method can be improved by using a variable mesh size.

## THEORETICAL CONSIDERATION

For simplicity we will consider an one-dimensional problem defined as:

$$\frac{\partial^2 u}{\partial x^2} = f(x) \tag{1a}$$

where

$$x_0 \le x \le x_{n+1},\ \ u(x_0) = u_0,\ \ u(x_{n+1}) = u_{n+1} \tag{1b}$$

Assume that f(x) is a twice-continuously differentiable function on region $[x_0, x_{n+1}]$, and that u(x) satisfies given boundary conditions (1b).

Let: $x_0 < x < x_1 < x_2 < \ldots < x_n < x_{n+1}$ be an arbitrary subdivision of $[x_0, x_{n+1}]$. Define $h_i = x_{i+1} - x_i$, i = 0, 1, 2, . . . , n, and approximate $\frac{\partial^2 u}{\partial x^2}$ by the second central difference quotient:

$$\frac{\partial^2 u}{\partial x^2} \cong \frac{\frac{u_{i+1} - u_i}{h_i} - \frac{u_i - u_{i-1}}{h_{i-1}}}{\frac{h_i + h_{i-1}}{2}} \tag{2}$$

By substituting (2) into (1) a set of linear equations is obtained:

$$-\frac{1}{h_{i-1}}u_{i-1} + \left(\frac{1}{h_{i-1}} + \frac{1}{h_i}\right)u_i - \frac{1}{h_i}u_{i+1} + \frac{h_{i-1} + h_i}{2} f(x_i,y_i) = 0$$

$$i = 1, \ldots, n \tag{3}$$

System (3) can be written in the matrix form:

$$\mathbf{Au} + \boldsymbol{\Phi} + \mathbf{b} = \mathbf{0} \tag{4}$$

The coefficient matrix **A** of system (3) is symmetric, positive definite [2]:

$$A = \begin{pmatrix} \left(\frac{1}{h_0}+\frac{1}{h_1}\right) & -\frac{1}{h_1} & 0 & & & & \\ -\frac{1}{h_1} & \left(\frac{1}{h_1}+\frac{1}{h_2}\right) & -\frac{1}{h_2} & & & & \\ & & \ddots & & & & \\ & 0 & -\frac{1}{h_{i-1}} & \left(\frac{1}{h_{i-1}}+\frac{1}{h_i}\right) & -\frac{1}{h_i} & 0 & \\ & & & & \ddots & & \\ & & & & -\frac{1}{h_{n-1}} & \left(\frac{1}{h_{n-1}}+\frac{1}{h_n}\right) \end{pmatrix} \tag{5}$$

Vectors $\Phi$ and $\mathbf{b}$ in (4) are obtained as:

$$\Phi = \begin{pmatrix} \frac{h_0+h_1}{2} f(x_1) \\ \frac{h_1+h_2}{2} f(x_2) \\ \vdots \\ \frac{h_{i-1}+h_i}{2} f(x_i) \\ \vdots \\ \frac{h_{n-1}+h_n}{2} f(x_n) \end{pmatrix}, \mathbf{b} = \begin{pmatrix} -\frac{u_0}{h_0} \\ \\ -\frac{u_{n+1}}{h_{n+1}} \end{pmatrix} \tag{6}$$

Since matrix $\mathbf{A}$ in (4) is positive definite, then a convex energy function can be defined as:

$$E(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T A\, \mathbf{v} + \mathbf{v}^T \varphi \quad \text{where } \varphi = \mathbf{b} + \Phi \tag{7}$$

Function $E(\mathbf{v})$ is uniformly convex and has a unique minimum in $\mathbf{v}$ [2] assuming $\varphi$ is continuous and bounded. A neural network [3], [4] which tries to minimize the above energy function $E(\mathbf{v})$ is designed such that:

$$\frac{du_i}{dt} = -\frac{\partial}{\partial v_i} E(v_1, v_2, \ldots, v_n) \tag{8}$$

where $u_i$ is the input of the i-neuron of the network and $v_i = g_i(u_i) = u_i$ is its output. Thus:

$$\frac{du_i}{dt} = -\sum_k a_{i,k} v_k - \varphi_i \quad v_i = u_i\,, \quad i = 1, \ldots, n \tag{9}$$

where $a_{i,k}$ is an element of matrix $\mathbf{A}$ at the intersection of the i row and k column. Due to the special structure of matrix $\mathbf{A}$, the obtained neural network is locally connected.

## EXAMPLE

As an example we consider a one-dimensional problem of solving the Poisson's equation of the electrical potential in a semiconductor bar with a pn junction. Assume that the semiconductor bar is extended in the x direction and that region for $x < 0$ is doped p type and the region for $x > 0$ is n type.

Let's assume that charge distribution in the bar can be approximated by an expression [5]:

$$\rho_v = 2\rho_{vo} \operatorname{sech}\frac{x}{a} \tanh\frac{x}{a} \tag{10}$$

as shown in Fig. 1 [5]. The maximum charge density $\rho_{v,max} = \rho_{vo} = eN_a = eN_d$ occurs at $x = 0.881a$ and depends on the acceptor and donor concentrations $N_a$ and $N_d$. The Poisson's equation that describes the semiconductor bar is given as:

$$\nabla^2 V = -\frac{\rho_v}{\varepsilon} \tag{11}$$

Subject to the charge distribution assumed above, this is one-dimensional problem:

$$\frac{d^2V}{dx^2} = -\frac{2\rho_{vo}}{\varepsilon} \operatorname{sech}\frac{x}{a} \tanh\frac{x}{a} \tag{12}$$

The exact solution of equation (11) can be obtained by integration:

$$V = \frac{4\rho_{vo}a^2}{\varepsilon}\left(\tan^{-1} e^{\frac{x}{a}} - \frac{\pi}{4}\right) \tag{13}$$

where the integrating constants are determined for the condition that the electric field $E_x$ must approach zero as $x \to \pm\infty$, and that the zero reference of potential was selected at the junction $x = 0$ (Fig. 1). The circuit is simulated for a particular numerical case:

$$\rho_{vo} = 250\frac{C}{m^2}\,,\ \varepsilon_r = 12\,,\ a = 10^{-6}\,m\,.$$

Although equation (12) has an exact analytical solution, we will solve the problem by applying the cellular neural network.

## CMOS CIRCUIT IMPLEMENTATION

For the one dimensional example considered in the above section, the resulting neural network is a row of cells. A possible circuit structure of the neural network cell is shown in Fig. 2.

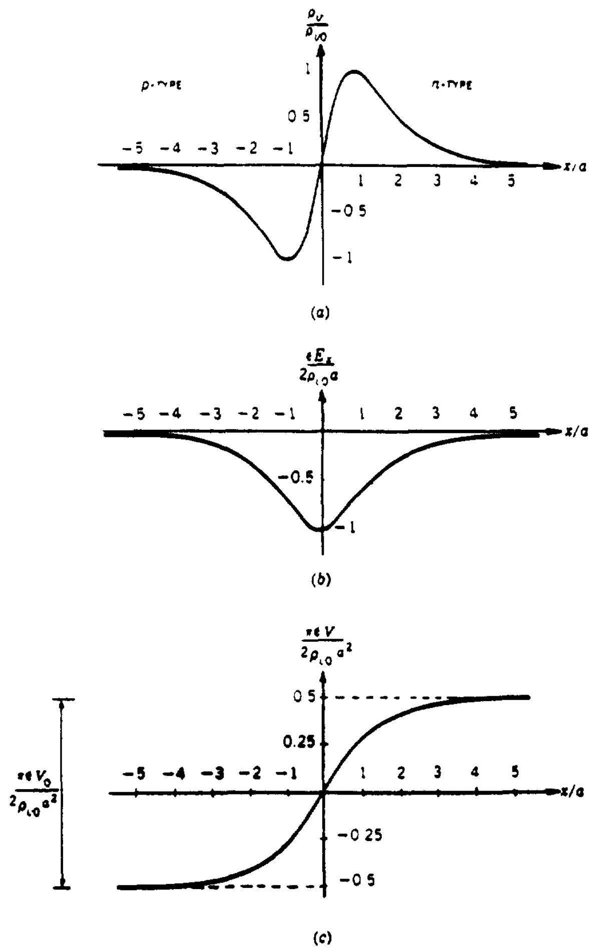


Figure 1. (a) The charge density, (b) the electric field intensity, and (c) the potential are plotted for a pn junction as functions of distance from the center of the junction. The p-type material is on the left, and the n-type is on the right.

The proposed cell is described by the following dynamic equation:

$$\frac{du_i}{dt} = \frac{1}{CR_{i-1}} v_{i-1} - \left(\frac{1}{CR_{i-1}} + \frac{1}{CR_i}\right) u_i + \frac{1}{CR_i} v_{i+1} + \frac{I_i}{C}$$

$$v_i = u_i \tag{14}$$

At the steady state $\frac{du_i}{dt} = 0$, and the circuit steady states, $u_i$, represent solutions of equations (3), i.e. the approximate solution of equation (1) at the mesh points.

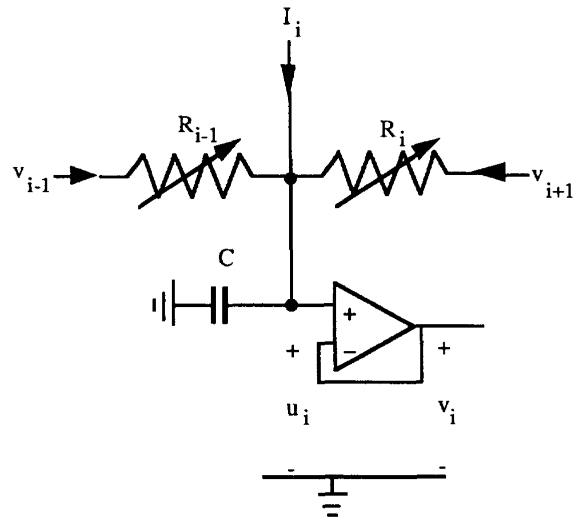


Figure 2. Neural network cell for the variable mesh problem

By comparing equations (3) and (14), the following parameter relations are determined:

$$h_i = C R_i \ ,$$

$$I_i = C \, \frac{h_{i-1} + h_i}{2} \, f(x_i) \tag{15}$$

The mesh size h can be changed simply by changing resistance $R_i$ in the neural cell.

In general, the error inherent in the finite-difference approximation is proportional to powers of the mesh size h. This error depends also on the nature of the function that represents the solution of the PDE. Since this function cannot be predicted prior to the actual solution, the approximation of the fourth derivative can be used as a useful estimate of the error [6]. The fourth partial derivative approximation is given by the following expression:

$$\left(\frac{\partial^4}{\partial x^4}\right)_i \cong \frac{1}{h^4}\left(u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2}\right) \tag{16}$$

If the fourth derivative evaluated by equation (16) appears to be large, then the mesh size h should be reduced. This criteria is used to determine the proper mesh size so that the accuracy of the solution is increased.

Partial Differential equation (12), considered in the previous section, is solved by applying the proposed method. Our idea was to experiment with the mesh size within the limited size of the neural network (the number of cells is n=32). Three different cases are considered:

Case I One mesh size: $h = 0.1$ a

Case II One mesh size: $h = 0.4$ a

Case III Variable mesh size:

$$h_1 = 0.1\text{ a} \quad \text{for } 0 < x < 2\text{ a},$$

$$h_2 = 0.4\text{ a for } 2\text{ a} < x < 6.8\text{ a},$$

where $a = 10^{-6}$m.

The simulation results for the above cases and the exact analytical solutions for V(x) are shown in Table 1.

Table 1:Simulation results for the three different sizes of h

| x(μm) | $V_i$ (circuit simulation) | | | | | $V_i$ (calculation) |
|---|---|---|---|---|---|---|
| | Case I | Case II | | Case III | | |
| | h = 0.1a | h = 0.4a | Fourth derivative | variable h | Error % | |
| 0 | 0 | 0 | | 0 | 0 | 0 |
| | 0.43 | | | 0.470 | 0.04 | 0.4698 |
| 0.2 | 0.86 | | | 0.936 | 0.10 | 0.9350 |
| | 1.27 | | | 1.39 | 0.08 | 1.3911 |
| 0.4 | 1.68 | 1.86 | | 1.84 | 0.32 | 1.8341 |
| | 2.06 | | | 2.26 | 0.31 | 2.2607 |
| 0.6 | 2.43 | | | 2.67 | 0.07 | 2.668 |
| | 2.78 | | | 3.06 | 0.19 | 3.054 |
| 0.8 | 3.10 | 3.45 | $0.12\ h^4$ | 3.42 | 0.07 | 3.4175 |
| | 3.40 | | | 3.76 | 0.06 | 3.7576 |
| 1.0 | 3.68 | | | 4.08 | 0.14 | 4.0743 |
| | 3.93 | | | 4.37 | 0.05 | 4.3677 |
| 1.2 | 4.16 | 4.67 | $0.07\ h^4$ | 4.64 | 0.03 | 4.6386 |
| | 4.37 | | | 4.89 | 0.04 | 4.8879 |
| 1.4 | 5.56 | | | 5.12 | 0.06 | 5.1166 |
| | 4.73 | | | 5.33 | 0.07 | 5.3259 |
| 1.6 | 4.88 | 5.54 | $-0.16\ h^4$ | 5.52 | 0.05 | 5.5171 |
| | 5.01 | | | 5.69 | 0.03 | 5.6915 |
| 1.8 | 5.13 | | | 5.85 | 0.01 | 5.8503 |
| | 5.24 | | | 6.00 | 0.01 | 5.9947 |
| 2.0 | 5.33 | 6.15 | 0 | 6.13 | 0.06 | 6.1260 |
| | 5.41 | | | | | 6.2453 |
| 2.2 | 5.47 | | | | | 6.3535 |
| | 5.53 | | | | | 6.4516 |
| 2.4 | 5.58 | 6.56 | 0 | 6.53 | 0.16 | 6.5406 |
| | 5.62 | | | | | 6.6213 |
| 2.6 | 5.65 | | | | | 6.6943 |
| | 5.68 | | | | | 6.7605 |
| 2.8 | 5.70 | 6.83 | 0 | 6.80 | 0.30 | 6.8205 |
| | 5.71 | | | | | 6.8748 |
| 3.0 | 5.72 | | | | | 6.9239 |
| | 5.73 | | | | | 6.9684 |
| 3.2 | | 7.02 | | 6.99 | 0.26 | 7.0087 |
| 3.6 | | 7.14 | | 7.10 | 0.49 | 7.1350 |
| 4.0 | | 7.23 | | 7.18 | 0.55 | 7.2198 |
| 4.4 | | 7.28 | | 7.23 | 0.64 | 7.2766 |
| 4.8 | | 7.32 | | 7.27 | 0.54 | 7.3147 |
| 5.2 | | 7.35 | | 7.29 | 0.68 | 7.3402 |
| 5.6 | | 7.37 | | 7.30 | 0.77 | 7.3573 |
| 6.0 | | 7.38 | | 7.31 | 0.79 | 7.3688 |
| 6.4 | | 7.39 | | 7.31 | 0.63 | 7.3565 |

Case I gives the worst results because not the whole region of x upon which the excitation function (the right side of equation 12) is distributed was covered by the mesh. This was the case due to the limited number of cells and small mesh size.

In case II, with the mesh size $h = 0.4$ a, the error is a consequence of a crude mesh. In Table 1, the fourth derivative calculated by applying equation 16 is also shown. It can be concluded that for x in the region from 0 to 2a the approximated value of the fourth derivative is significant. For that reason, the mesh parameter h is reduced from 0.4a to .1a in the x region from 0 to 2a, as shown in case III.

In the case III, where the two different mesh steps were used, the most accurate solution of the partial differential equation is obtained with an relative error less than 1%.

## CONCLUSIONS AND FUTURE WORK

In this work we presented an approach to solve a class of partial differential equations using neural networks. The energy function for the neural network is defined such that it is realized by analog neurons that are locally connected to its neighbors, thus avoiding the connection problem for a large number of neurons. The technique allows us to use variable mesh sizes, and thus to control the accuracy of the solution for a particular number of neuron cells. Currently, we are implementing this technique using CMOS technology. A CMOS chip is being designed for fabrication and measurements.

## REFERENCES


[1] D. Gobovic, M.E. Zaghloul, "Design of Locally Connected CMOS Neural Cells to Solve the Steady-State Heat Problem," *Proc. of 36th Midwest Symposium on Circuits and Systems*, 1993.

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

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

[4] 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.

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

[6] W.J. Karplus, Analog Simulation, McGraw-Hill, 1958.