J. C. Chedjou[1], and K. Kyamakya[1]

University of Klagenfurt (Austria) (1)

Use of CNN processors for ultra-fast solution ODE's and PDE's: A renaissance of the analog computer

***Abstract**. Setting analog cellular computers based on cellular neural networks systems (CNNs) to change the way analog signals are processed is a revolutionary idea and a proof as well of the high importance devoted to the analog simulation methods. This paper provides basics of the methods based on the CNNs paradigm that can be exploited for analog simulation of very complex systems which are modelled by ODEs and/or PDEs (an implementation on chip using CNN technology is possible even in FPGA). A proof of concept of the simulation approach developed in this paper is validated by solving some complex ODEs and/or PDEs models and by comparing the results obtained with those available in the literature (benchmarking). The computation based CNNs paradigm is advantageous as it provides accurate and ultra-fast solutions of very complex ODEs and PDEs.*

1 Introduction

The classical numerical approach based on Von Neumann to solve complex differential equations is exposed to divergences which manifest themselves by floating point over flow messages during computations. These divergences are generally caused by the high degree of stiffness in the systems [7-11]. Another problem faced by the classical numerical method is the difficulty to predict the duration of transient phases [7-11]. This duration does increase with the complexity of the systems. Further, the classical numerical method appears to be very time consuming (i.e. slow) and also, of less accuracy due to accumulation of round-off errors during computations [7-11]. This justifies the need of an approach which is robust to problems faced by the classical numerical method.

Nowadays analog computation is returning as an important alternative computing paradigm after being eclipsed by digital computation in the second half of the twentieth century. This is due to the fact that analog computing provides ultra fast and real-time computing leading to results with high accuracy when compare to the classical numerical method which provides results with less accuracy when dealing with complex/stiff nonlinear differential equations.

The state-of-the-art shows/presents the cellular neural networks (CNN) paradigm as being an attractive alternative to conventional numerical computation [1, 2]. It has been intensively shown that CNN is an analog computing paradigm which performs ultra fast calculations and provides accurate results [1, 2]. Interestingly, a speedup of the analog computing process is possible by an implementation on reprogrammable computing (i.e. Field-programmable gate array (FPGA))

This paper attempts to improve both the computing time and the accuracy of results which are some key factors to which is generally prone the classical numerical method when dealing with complex/stiff ODEs and PDEs. We explain and show the possibility of deriving the appropriate CNN- templates to solve complex ODEs and/or PDEs. Using our approach, these equations are mapped to a CNN array in order to facilitate the templates calculation. On the other hand complex/stiff PDEs are transformed into ODEs having array structures. This transformation is achieved by applying the method of finite difference. This method is based on the Taylor's series expansion. Applying the Taylor's series expansion leads to both the discretization of PDEs in space and the mapping of the resulting set of coupled ODEs to a CNN array.

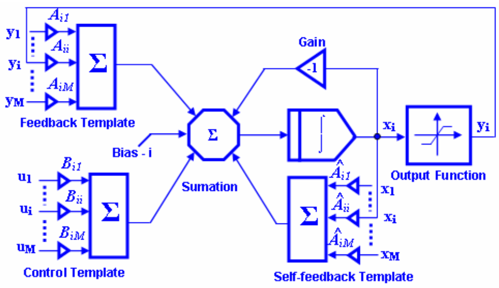
The structure of the paper is as follows. Section 2 provides a brief overview of the cellular neural networks paradigm. Section 3 explains/shows the approach to solve complex/stiff ODEs with the CNN- paradigm. Section 4 applies the same approach to solving complex/stiff PDE. Some sample results are obtained showing the efficiency of our approach. The last section (section 5) deals with concluding remarks and proposal of some open research questions.

2 Overview of Cellular Neural Networks (CNN)

The concept of Cellular Neural Networks (CNN) was introduced by Leon O. Chua and Yang [1]. The cellular neural networks processor is built using identical analog processing elements called cells [1-2]. These cells can be arranged in a k-dimensional square grid which is the most commonly used CNN type amongst many others namely the spherical CNN [4] and the star CNN [3] just to name a few. In these types of CNN, cells are locally connected (i.e each cell is connected to its neighborhood) via programmable weights called templates. These templates are changed to make it programmable the CNN cell array. Hence, the essential/fundamental of the technology based on the CNN paradigm is located in templates. These templates are sets of matrices which are space invariant (i.e. cloning templates) if the values of templates do not depend on the position of the cell [1, 2]. This condition/case is particularly interesting to materialize spatial discretization. A complete (or full) specification of the dynamics of a CNN cell array requires the definition of Dirichlet (or boundary) conditions. The CNN computing platform developed in this paper exploits the structure of the state- control CNN (SC-CNN) [1, 2] which is modeled by Eq. (1).

$$\dot{x}_i = -x_i + \sum_{j=1}^{M}\left[\hat{A}_{ij}x_j + A_{ij}y_j + B_{ij}u_j\right] + I_i \quad (1)$$

The coefficients $\widehat{A}_{ij}$, A_{ij} and B_{ij} are the self-feedback template, feedback template and control template, respectively. The schematic representation of a state-control CNN (SC-CNN) cell coupled to (M-1) neighbouring cells is shown in Fig. 1. I_i is the bias value and y_i is the nonlinear output sigmoid function of each cell. u_j denotes the input value and x_i represents the state of each cell.



Figs.1: SIMULINK Graphical representation of a SC-CNN Cell coupled to (M-1) Neighbours.

3 Solving nonlinear ODE with CNN

We consider the case of the Rayleigh oscillator which is a well-known prototype of nonlinear and stiff ordinary differential equation. The Rayleigh equation is modeled by Eq. (2).

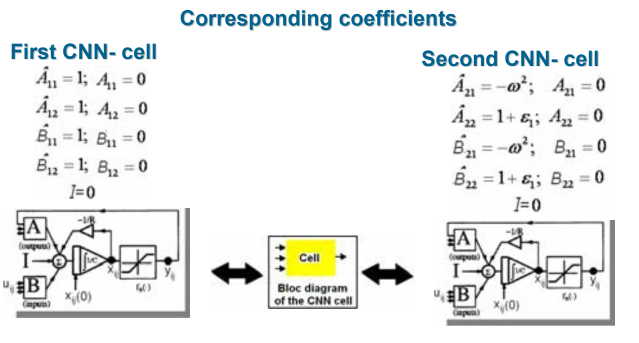
(2)

$$\frac{d^2x}{dt^2} - \varepsilon_1\left[1-\left(\frac{dx}{dt}\right)^2\right]\left(\frac{dx}{dt}\right) + \omega x + k\sin(2\pi F_1) = 0$$

Solving Eq. (2) using the cellular neural networks paradigm requires some mathematical manipulations. These manipulations are performed to achieve mapping of Eq. (2) in the form of Eq. (1). This mapping leads to the set of first order ODEs shown in Eq. (3).

$$(3)\quad \begin{cases} \dfrac{dx}{dt} = y \\ \dfrac{dy}{dt} = \varepsilon_1 y - \omega^2 x - \varepsilon_1 y^3 + k\sin(2\pi F_1) \end{cases}$$

The mapping of Eqs. (3) into the form of Eq. (1) and a good identification of the corresponding coefficients lead to the following conclusions. (a) Two CNN- cells are needed to solve Eq. (2); (b) The corresponding templates of each CNN- cell are defined in Fig. 2.



Figs.2: Schematic representation of the 2 CNN- cells needed to solve the Rayleigh equation with corresponding templates.

Using the values of the templates in Fig. 2, the complete CNN- processor to solve the Rayleigh equation has been obtained on SIMULINK as shown in Fig. 3. Fig. 3 also shows a chaotic phase portrait of the attractor of the Rayleigh oscillator for the set of parameters $\varepsilon_1 = 2.3$, $F_1 = 0.004$, $k = 2.398$, and $\omega = 1$.

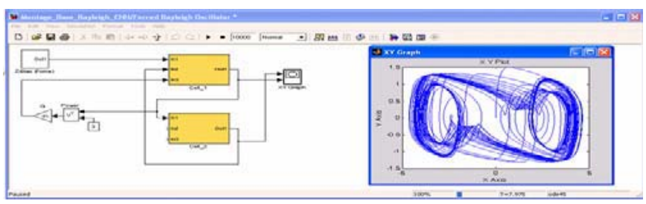


Fig.3: SIMULINK representation of the complete CNN- simulator of the Rayleigh equation and chaotic phase portrait.

Using the same set of parameters and the same computing platform (SIMULINK) we have performed the direct numerical simulation of the Rayleigh equation. The results obtained are shown in Fig. 4. Comparing the chaotic phase portrait displayed by the CNN- simulator implemented on-top of SIMULINK with the chaotic phase portrait obtained from a direct simulation of the Rayleigh equation on the SIMULINK platform, a very good agreement is obtained. More interestingly, it has been found that with the CNN-processor, the simulation time is approximately 100 time faster than the simulation time of a direct simulation on SIMULINK. This is clearly shown in Fig. 3 and Fig. 4 on which the dimensioning simulation time is shown. This conclusion confirms the ultra fast computing potentialities/capabilities of the analog CNN- simulator.

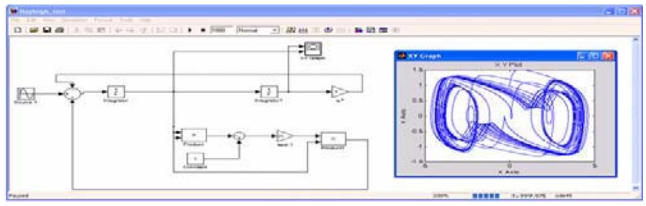


Fig.4: SIMULINK graph to solve the Rayleigh equation and chaotic phase portrait.

4 Solving nonlinear PDE with CNN

Partial differential equations are good prototypes for modelling traffic flow which is an interesting phenomenon of modern world. The modelling process though very important to get insights on/of the traffic dynamics appears challenging as we all experience it daily traffic problems are still not well understood. The reasons of this misunderstanding are justified by: the dynamic nature of the traffic dynamics which is time varying; the unpredictable or stochastic nature of the traffic dynamics which could be caused/due by/to accidents, maintenances of roads, or unpredictable events/situations influencing the traffic. These are sample examples showing the nonlinear dynamical character of the traffic dynamics. Thus, nonlinear equations are good candidates to describe the dynamics of traffic. At the macroscopic level, nonlinear partial differential equations are use for the modeling process. Specifically, the Burgers equation has been intensively used for this purpose. However, it is well known that deriving exact analytical solutions of complex/stiff nonlinear partial differential equations is challenging/impossible. The numerical simulation of complex nonlinear partial differential equations is exposed to transient phenomena, accumulation of round-off errors during computation and stiffness caused by the high degree of nonlinearity. This simulation is very time consuming as well when dealing with complex nonlinear partial differential equation. The key issue when simulating complex partial differential equations is the Dirichlet conditions (i.e. boundary conditions) which are generally very difficult to define during the numerical simulation of equations modeling engineering systems. This justifies the need of appropriate and efficient simulation tools which are robust to the problems faced by the classical simulation tools for solving complex partial differential equations. It should be worth noticing that this issue is still unsolved by the state-of-the-art as the available traffic simulation algorithms/tools are slow and provide results with low accuracy.

This paper develops a conceptual computing framework for complex/stiff nonlinear ODE and PDEs. The concept developed is based on the paradigm of cellular neural networks which could be implemented on FPGA.

Amongst equations of practical interest in engineering, the Burgers equation (Eq. 3) has been intensively used for many applications (e.g. modeling Traffic flow, modeling Shock Waves propagation, modeling Fluids dynamics, and modeling Heat propagation) [5]. This is a nonlinear wave equation which can be used to describe the corresponding density waves from several models of traffic flow. Another interesting application of the Burgers equation is the modeling and optimal control of traffic flow.

(3) $$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} - \beta u(x,t) \frac{\partial u(x,t)}{\partial x}$$

α and β are constants and $u(x,t)$ is the density of the wave. Eq. (3) is a nonlinear equation with traveling wave solutions. In the conservation law form (i.e. $\alpha = 0$) the Burgers equation reduces to a first order hyperbolic PDE. This equation is illustrated with the well-known model of highway traffic theory, namely the Lighthit-Whitham-Richards (LWR) [6].

It can be shown that the solution of Eq. (3) is envisaged in the form

(4a) $$u(x,t) = g[x - vt]$$

(4b) $$v(x,t) = \alpha + \beta u(x,t)$$

g is an unknown function which depends on the initial-boundary value problem (to be specified/defined). The solution in Eqs. (4) describes a right-moving wave with a velocity $v(x,t)$ depending on the density of the wave. This dependence can lead to many striking effects namely breaking and formation of shock fronts. This last corresponds to bunching of cars on a highway.

In order to solve the PDE in Eq. (3) with the paradigm of cellular neural network, the spatial discretization method (i.e. the finite difference method) is performed in order to transform the proposed PDEs into sets of ordinary differential equations ODEs which are suitable forms the CNN paradigm can solve. Specifically, we first discretize the function spatially by using the Taylor series expansion and next we use the CNN paradigm to account for temporal change (i.e. solving ODE in time domain).

The second order Taylor's series expansion (of the solution in Eq. (3)) around a fixed point x_0 can be envisaged in the form

(5) $$u(x,t) = u(x_0,t) + (x - x_0)u'(x_0,t) + \frac{(x - x_0)^2}{2} u''(x_0,t)$$

Considering two neighboring points situated at a distance Δx_0 around the fixed point x_0 (i.e. left and right to x_0) can lead to the following mathematical formulation:

(6) $$x_i = x_0 \pm \Delta x_0$$

Eq. (6) can be used to derive new forms of Eq. (5) as follows:

(7a) $$u(x_0 + \Delta x_0, t) = u(x_0,t) + \Delta x_0 u'(x_0,t) + \frac{(\Delta x_0)^2}{2} u''(x_0,t)$$

(7b) $$u(x_0 - \Delta x_0, t) = u(x_0,t) - \Delta x_0 u'(x_0,t) + \frac{(\Delta x_0)^2}{2} u''(x_0,t)$$

Thus, one can use Eqs. (7) to deduce the Taylor's series expansion of the first and second derivatives around the fixed point x_0 as follows:

(8a)
$$u''(x_0,t) = \frac{u(x_0 + \Delta x_0,t) + u(x_0 - \Delta x_0,t) - 2u(x_0,t)}{(\Delta x_0)^2}$$

(8b) $$u'(x_0,t) = \frac{u(x_0 + \Delta x_0,t) - u(x_0 - \Delta x_0,t)}{2\Delta x_0}$$

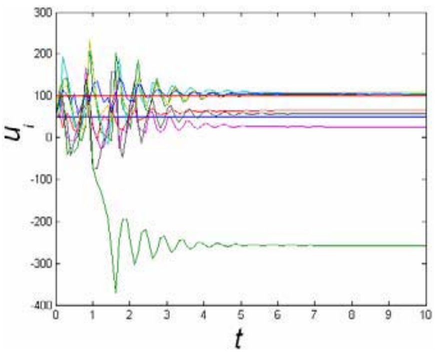
Eq. (6) can be used to construct a spatial domain which is made-up of a number of grid-points x_i arranged in a regular form, Δx_0 being the distance between them (i.e. grid-points). Therefore the time evolution of the solution $u(x,t)$ in Eq. (3) is obtained at each grid-point x_i. This leads to a general solution $u(x_i,t) = u_i$ which can be obtained from the following first order ordinary differential equation derived by substituting Eqs. (8) into Eq. (3).

(9) $$\frac{du_i}{dt} = \left(\frac{\alpha}{(\Delta x_0)^2}\right)[u_{i+1} - 2u_i + u_{i-1}] - \left(\frac{\beta}{2\Delta x_0}\right) u_i [u_{i+1} - u_{i-1}]$$

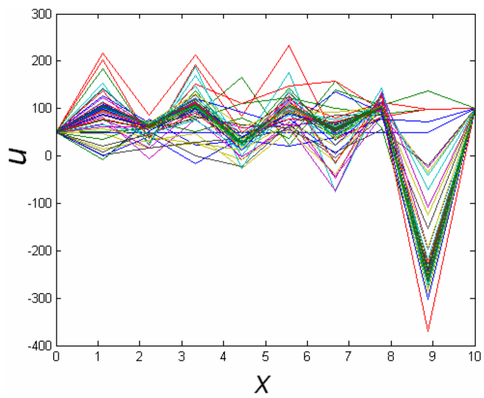
The spatial domain is built from a number of grid-points localized by the position x_i, the index i being an integer. Therefore Eq. (9) clearly shows that the analog computing of partial differential equations (PDEs) is possible by transforming them into ordinary differential equations (ODEs) which are expressed in the form of Eq. (9). This equation is a set a coupled first order coupled ODEs, the number of equations being fixed by the index i.

To solve Eq. (9) by the cellular neural networks (CNN) paradigm, we apply some algebraic manipulation on Eq. (9) to transform it into the form of i first order ODEs which are further identified with the SC-CNN model in Eq. (1). Here, the total number of cells used to built the complete CNN-processor is fixed by the index i.

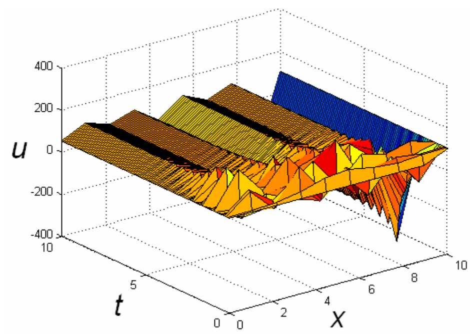
Our various numerical computations using the CNN paradigm were very slow as for the sake of obtaining accurate results we were obliged to use a huge amount of CNN-cells. It was possible to obtain some sample results up to the maximal number of 10 CNN- cells ($i_{max} = 10$). Fig. 5a shows the temporal evolution of the solution in Eq. (3) obtained by our CNN- processor built with 10 cells for the parameters $\alpha = \frac{1}{2}$ and $\beta = 1$. Using the same set of parameters, we have also performed the direct numerical simulation of Eq. (3) on MATLAB. Fig. 5b shows the spatial evolution of the solution in Eq. (3) while in Fig. 5c is shown the spatio-temporal evolution of the solution in Eq. (3). The results in Fig. 5a obtained using CNN- simulator were generally very close to the same plots obtained using simulation on MATLAB. However, a divergence was observed for $i \leq 8$. This divergence can be explained by the fact that the accuracy of the CNN- simulator decreases with decreasing number of cells constituting the CNN- simulator.



Figs.5a: Results of the CNN- simulator showing the temporal evolution of the solution in Eq. (3) for $\alpha = 1/2$, $\beta = 1$ and $i = 10$.



Figs.5b: Results of a direct simulation on MATLAB showing the spatial evolution of the solution in Eq. (3) for $\alpha = 1/2$, $\beta = 1$ and $i = 10$.



Figs.5c: Results of a direct simulation on MATLAB showing the spatio-temporal evolution of the solution in Eq. (3) for $\alpha = 1/2$, $\beta = 1$ and $i = 10$.

5 Conclusion

This work has presented a general concept for solving complex/stiff equation with the cellular neural networks (CNN) paradigm. The concept has been applied on an ordinary differential equation (Rayleigh) and a partial differential equation (Burgers). It has been shown that some algebraic mathematical manipulations must be performed in order to map the differential equations under investigation into the same form as the equation describing the dynamics of the CNN processor. Hence, the corresponding templates are obtained by the identification process. The results obtained (using the concept proposed in this paper) were generally very close to the results obtained by simulating on MATLAB and/or SIMULINK. Another interesting result obtained has shown/demonstrated the fast computing potentialities/capabilities of the CNN processor.

An interesting research direction under consideration is the on-chip (FPGA) implementation of the computation platform developed in this work in order to speedup the computation process.

REFERENCES

[1] L.O Chua and L. Yang, Cellular Neural Networks: Theory, *IEEE Trans. On Circuits and Systems*, Vol.35, 1988, 1257-1272

[2] G. Manganaro, P. Arena, L. Fortuna, Cellular Neural Networks: Chaos, Complexity and VLSI Processing, *Springer-Verlag Berlin Heidelberg*, 1999, 44-45.

[3] M. Ito, and L.O Chua, Star cellular neural networks for associative and dynamic memories: International Journal of Bifurcation and Chaos, vol. 14, 2004, 1725-1772.

[4] T. Yang, K. R. Crounse and L.O Chua, Spherical cellular nonlinear networks: International Journal of Bifurcation and Chaos, vol. 11, 2001, 241-257.

[5] T. Roska, L.O Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, Simulating nonlinear waves and partial differential equations via CNN- Part I: Basic techniques, *IEEE Trans. On Circuits and Systems*, Vol. 42, 1995, 807-815.

[6] J.-P. Aubin, A.M. Bayen, P. T. Saint-Pierre, Computation and control of solutions to the Burgers equation using viability theory, proceeding of the American Control Conference, vol. 6, 2005, 3906-3911.

[7] J. C. Chedjou, K. Kyamakya, I. Moussa, H. –P. Kuchenbecker, W. Matis, "Behavior of a self-sustained Electromehanichal Transducer and Routes to Chaos", Journal of vibrations and Acoustics, ASME Transactions, Vol. 128, pp 282-293, 2006.

[8] J. C. Chedjou, K. Kyamakya, Van Duc Nguyen, I. Moussa and J. Kengne" Performance evaluation of analog systems simulation methods for the analysis of nonlinear and chaotic modules in communications", ISAST Transactions on Electronics and Signal processing, N°1, vol.2, 2008, pp71-82.

[9] J. C. Chedjou, H. B. Fotsin, P. Woafo, and S. Domngang, "Analog Simulation of the Dynamics of a van der Pol oscillator coupled to a Duffing oscillator," IEEE Transactions on Circuits and Systems-I, vol. 48, pp. 748-757, 2001.

[10] J. C. Chedjou, K. Kyamakya, W. Mathis, I. Moussa, A. Fome the, A. V. Fono, "Chaotic Synchronization in Ultra Wide Band Communication and Positioning Systems," Journal of Vibration and Acoustics, Transactions on the ASME, Vol. 130, 2008, 011012/1 - 011012/12

[11] J. C. Chedjou, "On the Analysis of Nonlinear Electromechani cal Systems With Applications," Dr. –Ing. Dissertation, University of Hanover, Germany, Published under the code: ISBN 3-8322-3750-X, Shaker Verlag (Germany), march 2005.