CNN-based ultrafast solver of stiff ODEs and PDEs for enabling realtime Computational Engineering

J.C. Chedjou and K. Kyamakya
Transportation Informatics Group, Institute of Smart Systems Technologies, University of Klagenfurt, Klagenfurt, Austria

Abstract

Purpose – This paper seeks to develop propose and validate, through a series of presentable examples, a comprehensive high-precision and ultra-fast computing concept for solving stiff ordinary differential equations (ODEs) and partial differential equations (PDEs) with cellular neural networks (CNN).

Design/methodology/approach – The core of the concept developed in this paper is a straightforward scheme that we call "nonlinear adaptive optimization (NAOP)", which is used for a precise template calculation for solving any (stiff) nonlinear ODEs through CNN processors.

Findings – One of the key contributions of this work, this is a real breakthrough is to demonstrate the possibility of mapping/transforming different types of nonlinearities displayed by various classical and well-known oscillators (e.g. van der Pol-, Rayleigh-, Duffing-, Rössler-, Lorenz-, and Jerk-oscillators, just to name a few) unto first-order CNN elementary cells, and thereby enabling the easy derivation of corresponding CNN-templates. Furthermore, in case of PDEs solving, the same concept also allows a mapping unto first-order CNN cells while considering one or even more nonlinear terms of the Taylor's series expansion generally used in the transformation of a PDEs in a set of coupled nonlinear ODEs. Therefore, the concept of this paper does significantly contribute to the consolidation of CNN as a universal and ultra-fast solver of stiff differential equations (both ODEs and PDEs). This clearly enables a CNN-based, real-time, ultra-precise, and low-cost Computational Engineering. As proof of concept a well-known prototype of stiff equations (van der Pol) has been considered; the corresponding precise CNN-templates are derived to obtain precise solutions of this equation.

Originality/value – This paper contributes to the enrichment of the literature as the relevant state-of-the-art does not provide a systematic and robust method to solve nonlinear ODEs and/or nonlinear PDEs using the CNN-paradigm. Further, the "NAOP" concept developed in this paper has been proven to perform accurate and robust calculations. This concept is not based on trial-and-error processes as it is the case for various classes of optimization methods/tools (e.g. genetic algorithm, particle swarm, neural networks, etc.). The "NAOP" concept developed in this frame does significantly contribute to the consolidation of CNN as a universal and ultra-fast solver of nonlinear differential equations (both ODEs and PDEs). An implantation of the concept developed is possible even on embedded digital platforms (e.g. field-programmable gate array (FPGA), digital signal processing (DSP), graphics processing unit (GPU), etc.); this opens a broad range of applications. On-going works (as outlook) are using NAOP for deriving precise templates for a selected set of practically interesting PDE models such as Navier Stokes, Schrödinger, Maxwell, etc.

Keywords Differential equations, Computers

Paper type Research paper

COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering
Vol. 30 No. 4, 2011
pp. 1334-1350

0332-1649
DOI 10.1108/03321641111133235

1. Introduction

The last decades have witnessed a tremendous attention on solving nonlinear and stiff models (ordinary differential equations (ODEs) and/or partial differential equations

(PDEs)) with the cellular neural networks (CNN) paradigm (Chua and Yang, 1988). The interest devoted to solving stiff models can be explained by their multiple potential applications especially in the so-called Computational Engineering context. Indeed, nonlinear models have been intensively used to understand, predict and describe the dynamical behavior of various engineering or natural systems. In the field of transportation and logistics, for example, traffic models do take the form of ODEs and/or PDEs (Uzunova *et al.*, 2008). Still, in the field of transportation, various image processing tasks which are of high importance for visual sensors in advance driver assistant systems (e.g. contrast enhancement, segmentation, edge detection, etc.) can be expressed through solving corresponding stiff ODEs and/or PDEs (Zhu and Mumford, 1998).

Diverse contributions have been made to develop analytical, numerical and even hardware-based approaches to solve stiff ODEs and/or PDEs. Amongst these contributions some have retained our attention namely "the solutions of PDEs and ODEs using the CNN-paradigm". In fact, the flexibility of the CNN-paradigm and its huge potential to enable a renaissance of the old "analog computing" through an emulation on digital platforms (e.g. FPGA or GPU, etc.) to perform ultra-fast and accurate computing of nonlinear models are some of its strongest points. Nevertheless, the relevant state of the art does not provide significant information related to a straight-forward method to calculate the CNN-templates needed for solving stiff ODEs and/or PDEs with the CNN-paradigm. Despite some intensive works developed in this direction, it is still unclear how to solve PDEs and/or ODEs with good accuracy or high precision. Only approximate solutions exist, for example the use of CNN processors in an approximation of numerical solutions of PDEs involving the finite difference method (Roska *et al.*, 1995; Negro *et al.*, 2005; Krstic *et al.*, 2003; Niu *et al.*, 2001; Kozek *et al.*, 1995; Kozek and Roska, 1994). This late approach does not provide accurate results due to the Taylor series' expansion, which does consider only up to the first order (i.e. linear expansion). A further interesting published approach to solve PDEs is the group of learning schemes involved in an approximated solution of PDEs through CNN processors (Puffer *et al.*, 1995; Aarts and Veer, 2001; Tsoulos *et al.*, 2009; Chua *et al.*, 1995; Puffer *et al.*, 1996; Nossek, 1998). This late approach does require some initial solutions along with some critical parameter settings of the equations under investigation in order to enable the training process. This is a clearly significant drawback as it is not always possible to provide this data/information whenever dealing with stiff ODEs and/or PDEs.

Our aim in this paper is therefore to contribute to the enrichment of the relevant state of the art by proposing/developing a systematic methodology (based on the CNN-paradigm) which should help to clear some of the problems actually unsolved by the classical above described approaches. The key challenge thereby is developing a CNN-based computing concept for performing both ultra-fast and high-precision computing of stiff differential equations. The proposed method is based on a nonlinear adaptive optimization (NAOP) scheme to which we give the acronym "NAOP". For proof of concept, the novel approach developed in this paper is applied to derive solutions of selected classical and well-known examples of stiff ODEs. In the following, the flexibility of the approach developed is extensively discussed and we then do show/explain an easy extension of this approach to similarly efficiently solving stiff PDEs.

The rest of the paper is organized as follows. Section 2 presents an in-depth description of the novel concept. The quintessence of NAOP is explained and we thereby describe the scheme for deriving appropriate CNN-templates values for any given

nonlinear ODEs. Section 3 does then focus on the proof of concept through a selected nonlinear differential equation that is solved using the new concept developed in this paper: the van der Pol equation. For this, corresponding "precise" templates are calculated through NAOP. In Section 4, the possible extension of the novel scheme, involving NAOP for solving PDEs is discussed. And finally, a series of concluding remarks are presented in Section 5 along with the presentation of some interesting open research questions (outlook) that are under investigation in some of our on-going works.

2. The concept of "NAOP" for CNN template calculation for solving stiff ODEs

This section describes the approach based on the NAOP for solving ODEs. The overall flow diagram of this approach is schematically displayed by the synoptic shown in Figure 1.

The NAOP is performed by a complex "computing", "module/entity/procedure" which does work on two inputs. The first input contains wave-solutions of the models describing the dynamics of a CNN- network model built from state-control templates (equation (1)):

$$\frac{dx_i}{dt} = -x_i + \sum_{j=1}^{M} \left[\hat{A}_{ij} x_j + A_{ij} Y_j^* + B_{ij} u_j \right] + I_i \tag{1a}$$

x_i is the state of the CNN-processor, Y_j^* is the nonlinear sigmoid function of the CNN-processor, u_j is the input of the CNN-processor, $\hat{A}_{ij}$ is the state-control template, A_{ij} is the feedback template, B_{ij} is the feed-forward template, and I_i is the threshold. The source of nonlinearity in the model described by equation (1a) is expressed by the sigmoid function Y_j^* which is expressed in the following mathematical form:

$$Y_j^* = \frac{|x_i + 1| - |x_i - 1|}{2} \tag{1b}$$

The vector flow $\vec{\Theta}$ of the mathematical model representing the dynamics of the state control CNN (equation (1a)) is defined as follows:

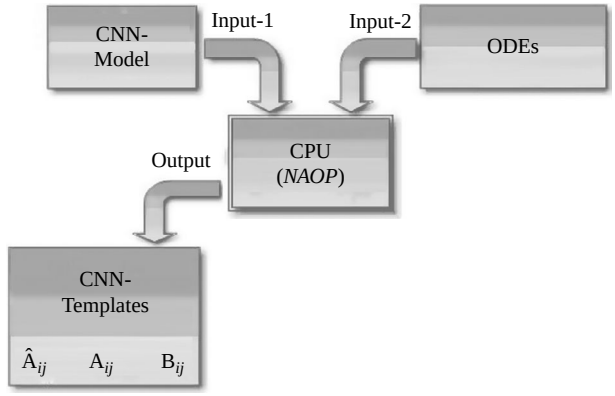


Figure 1.
Synoptic representation of the key steps involved in the NAOP approach used for a precise template calculations for solving both linear and nonlinear differential equations

$$\vec{\Theta}\left(x_i, \frac{dx_i}{dt}, \frac{d^2x_i}{dt^2}, \frac{d^3x_i}{dt^3}, \ldots \frac{d^nx_i}{dt^n}\right) \tag{1c}$$

where x_i represents the temporal evolution of the state variable of the mathematical model of the state control CNN and d^nx_i/dt^n represents the nth derivative of x_i.

The second input contains wave-solutions of the model or better the linear/nonlinear differential equation, under investigation which could be re-written in the following simplified form as a set/couple of second order ODEs (equation (2)):

$$\frac{dy_i}{dt} = F\left(y_i, y_i^n, \dot{y}_i^m, z_i, z_i^n, \dot{z}_i^m, t\right) \tag{2a}$$

$$\frac{dz_j}{dt} = F\left(y_j, y_j^n, \dot{y}_j^m, z_j, z_j^n, \dot{z}_j^m, t\right) \tag{2b}$$

$$z_j = \frac{dy_i}{dt} \tag{2c}$$

The ODE under investigation (equations (2a), (2b) and (2c) is represented in the state space "phase portrait" in terms of the components of the vector flow $\vec{\Phi}$ defined as follows:

$$\vec{\Phi}\left(y_i, \frac{dy_i}{dt}, \frac{d^2y_i}{dt^2}, \frac{d^3y_i}{dt^3}, \ldots \frac{d^ny_i}{dt^n}\right) \tag{2d}$$

Where y_i are coordinates representing the temporal evolution of the solution of a given ODE under investigation and d^ny_i/dt^n represents the nth derivative of y_i. The output of the NAOP system will generate, after extensive iterative computations or "training" steps, appropriate CNN-templates to solve the corresponding ODEs described in equation (2) when the convergence of the training process is achieved.

The global process to derive the CNN-templates (i.e. NAOP) can be summarized as follows. The learning/training process is based on a mapping between the two inputs of the "NAOP" procedure. A convergence to local minima is the key purpose governing this template calculation process, the so-called "NAOP". To achieve this, various basins of attraction are investigated sequentially, and corresponding CNN-templates are determined for those various initial conditions. If some local attractors diverge from a local minimum, new sets of initial conditions are automatically generated to annihilate the divergence leading to a possible convergence to a local minimum (this is the unique/only solution of the "mapping based training process" which provides CNN-templates corresponding to the real solution of the ODEs under investigation (equation (2)). A large number of randomly generated attractors (either regular or chaotic) are obtained through various numerical simulations whereby each attractor corresponds to a specific set of CNN-templates. An attempt to map these attractors to those generated by the model (i.e. ODEs) under investigation is performed (through the temporal mapping in the nth dimensional phase space of equation (1c) and (2d) showing the respective dynamics of both the CNN-model (equation (1a)) and the ODE under investigation (equations (2a), (2b) and (2c)). This mapping is performed in a sequential process leading to the convergence to a local minimum when the mapping is achieved successfully. However, it should be worth a mentioning that a successful mapping of equations (1) with equation (2) is achieved

through a correct (or perfect) modeling of the "objective function and related constraints". When this condition is achieved it becomes very easy to find the optimal solution (i.e. the local minimum) using the "NAOP" concept. This is a strong point of the concept developed in this paper as it is well-known that many optimization concepts presented in the relevant literature are subjected to a key/main difficulty (i.e. the difficulty to achieve convergence) due to the well-known inherent local minimum problem of the Hopfield neural network (Hopfield and Tank, 1985; Smith, 1999). It is further worth a mentioning that the quintessence of the "NAOP" concept is in the core an adaptive training process that is very comparable to the concept developed for the training of Hopfield neural networks towards an efficient tracking of local minima. Nevertheless, NOAP has been demonstrated capable of mapping all known nonlinearity of ODEs unto appropriate templates of a first-order CNN processor matrix. Further, the NAOP has the potentiality to overcome trial-and-error training processes leading to a flexible and robust convergence of the training processes.

The overall process to derive the CNN-templates can be summarized as follows. The ODEs under investigation are represented in the state space "phase portrait" in terms of the components of the vector flow $\vec{\Phi}$ defined in equation (2d). Similarly, the vector flow $\vec{\Theta}$ of the mathematical model representing the dynamics of the state control CNN is defined in equation (1c). The objective function is derived based on the statement/condition that a good mapping must be achieved between the mathematical ODE model under investigation and the model of the state control CNN. Equivalently to this statement, the two vectors flow $\vec{\Phi}$ and $\vec{\Theta}$ do evolve, at long term, on a common trajectory in the phase space representation. Therefore, the objective function can be formulated/expressed in the following mathematical form:

$$Min\left[\sum_{n=0}^{\infty}\left(\frac{d^n x_i}{dt^n}-\frac{d^n y_i}{dt^n}\right)^2\right] \tag{3}$$

n is an unknown integer corresponding to the order of the ODE under investigation. The next step of the modeling process is concerned with the formulation of the related constraints. Constraints are defined to make sure the statement above (equation (3)) is fulfilled at long term (i.e. in time domain). This constraint can be formulated mathematically as follows:

$$\frac{d^n x_i}{dt^n}=\frac{d^n y_i}{dt^n} \quad (\text{for all t}) \tag{4}$$

The Lagrange function $L\left(x_i, x_i^{(1)}, \ldots x_i^{(n)}, y_i, y_i^{(1)}, \ldots y_i, {}^{(n)}\, \lambda_i, \gamma_i\right)$ is obtained by combining the objective function with the related constraints. This function is formulated mathematically as follows:

$$L=\sum_{i=0}^{\infty}\left(\frac{d^n x_i}{dt^n}-\frac{d^n y_i}{dt^n}\right)^2+\sum_{i=0}^{\infty}\lambda_i\left(\frac{d^n x_i}{dt^n}-\frac{d^n y_i}{dt^n}\right) \tag{5}$$

x_i^n represents the *nth* derivative of x_i, y_i^n represents the nth derivative of y_i, λ_i are multiplier-neurons and γ_i are coefficients of the ODE" under investigation. The corresponding CNN-templates (which are calculated/determined through the identification process based on the training/learning method) are expressed in terms of γ_i. Mention that the key steps involved in the complete technique of solving stiff ODEs

(using the CNN-paradigm) are twofold. The first step is an "offline optimization process", the objectives being the minimization of the Lagrange function $L\left(x_i, x_i^{(1)}, \ldots x_i^{(n)}, y_i, y_i^1, \ldots y_i^n, \lambda_i, \gamma_i\right)$ in order to derive the appropriate corresponding CNN-templates. This is achieved using the basic differential multiplier method (BDMM) (Wang *et al.*, 1993; Platt and Barr, 1998), which is the combination of two gradient techniques. The first technique is based on gradient decent. Here, the state variables of the network slide downhill, opposite to the gradient to find the minimum of the function. The second technique is related to the application of gradient ascent. In this case, the maximum function is obtained by proceeding in the positive direction of the gradient. We apply the BDMM to the "Lagrange function" in equation (5) to derive equation (6). Specifically, gradient descent is applied on decision variables/neurons and gradient ascent is applied on multiplier variables/neurons of the Lagrange function modeled in equation (5), leading to equation (6):

$$\frac{dx_i}{dt} = -\alpha \frac{\partial L}{\partial x_i} \tag{6a}$$

$$\frac{d\lambda_i}{dt} = +\beta \frac{\partial L}{\partial \lambda_i} \tag{6b}$$

Equation (6) is the characteristic model of the BDMM. This model (from which the CNN-templates of ODEs are calculated) reveals the coupling between the dynamics of decision neurons (x_i) and the dynamics of multiplier neurons (λ_i). α and β are step sizes for updating decision neurons and multiplier neurons, respectively. The second step is related to the design and implementation of the CNN-computing platform. The CNN-templates derived through an offline optimization process are exploited to compute solutions of the stiff ODEs under investigation as it is clearly shown in Figure 3. Indeed, the values of the CNN-templates are inserted in the CNN-computing platform designed (Figure 3) to solve ODEs. As proof of concept of the method developed in this paper, the novel technique is being applied in the next section for solving a specific and well-known prototype of ODEs namely the van der Pol oscillator. It is well-known that this is a well-known prototype of nonlinear and self-sustained oscillator which undergoes stiff dynamics in its relaxation states. Here, the stiffness (Hairer *et al.*, 1996) is characterized by the abrupt variation observed in the nonlinear temporal dynamics of the van der Pol oscillator (this is the relaxation state of the van der Pol oscillator which is observed for large values of ε (i.e. $\varepsilon \gg 10$ in equation (7)).

3. Applications to solving nonlinear ODEs

We restrict our analysis to the case of the van der Pol oscillator which is a good prototype of a well-known self-sustained oscillator having the interesting characteristic of being able to generate sinusoidal-, quasi-periodic-, and relaxation-oscillations (i.e. a stiff dynamics) (equation (7)):

$$\frac{d^2y}{dt^2} - \varepsilon(1 - y^2)\frac{dy}{dt} + \omega^2 y = 0 \tag{7}$$

Two possible states can be generated by equation (7). The first is the sinusoidal or almost sinusoidal state ($\varepsilon \ll 1$). The second one is the quasi-periodic state ($\varepsilon \ll 1$) which could lead to relaxation oscillations (i.e. stiff dynamics) for large values of ε ($\varepsilon \gg 10$). We now want to solve equation (7) using the CNN-paradigm. We first envisage the case

where $\varepsilon = 0.25$ and $\omega = 1$. For these parameter values the NAOP concept has been exploited to calculate the corresponding CNN-templates after convergence of the training process. This convergence is clearly shown by the plots in Figures 2 and 3 showing the temporal evolution of the state-control templates $\hat{A}_{ij}$ (Figure 2) and the feedback templates A_{ij} (Figure 3). As it appears in Figure 2, the convergence is achieved after a long transient phase displayed by the global training network (i.e. NAOP). From Figure 2, one can easily read the corresponding CNN-templates that are then used to solve the van der Pol equation.

The template values in Figure 2 have been used/inserted in Figure 3 to obtain the solution of equation (7) through the CNN-paradigm. Indeed, Figure 3 is a general representation in SIMULINK of a CNN processor platform to solve second-order nonlinear ordinary differential equations (ODEs). The key contribution of our approach, which is a breakthrough, is that we are now capable of transforming/mapping any type of nonlinearity displayed by nonlinear coupled and uncoupled ODEs unto the type of nonlinearity displayed by the elementary first-order CNN cell model. As proof of concept of the approach developed in this paper, we have used the CNN-templates derived by this scheme to obtain the exact solutions of equation (7). The graphical representation of the CNN processors for second order ODEs shown in Figure 3 has been used for rapid prototyping purposes; thus an implementation of the NAOP concept on embedded digital platforms (i.e. DSP or FPGA or GPU platforms) is then straight-forward. A direct

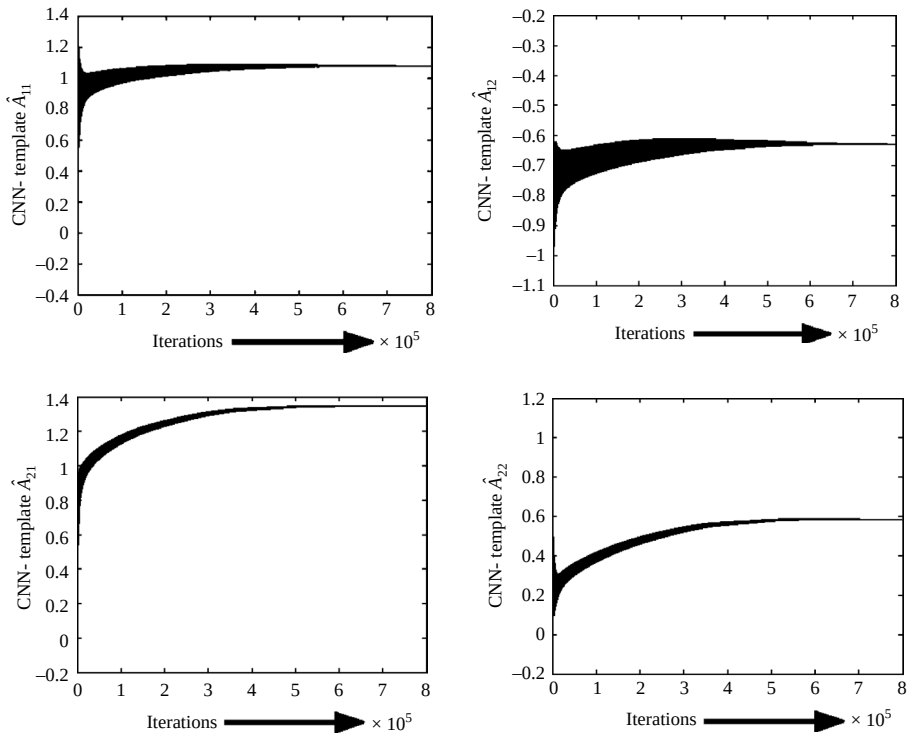
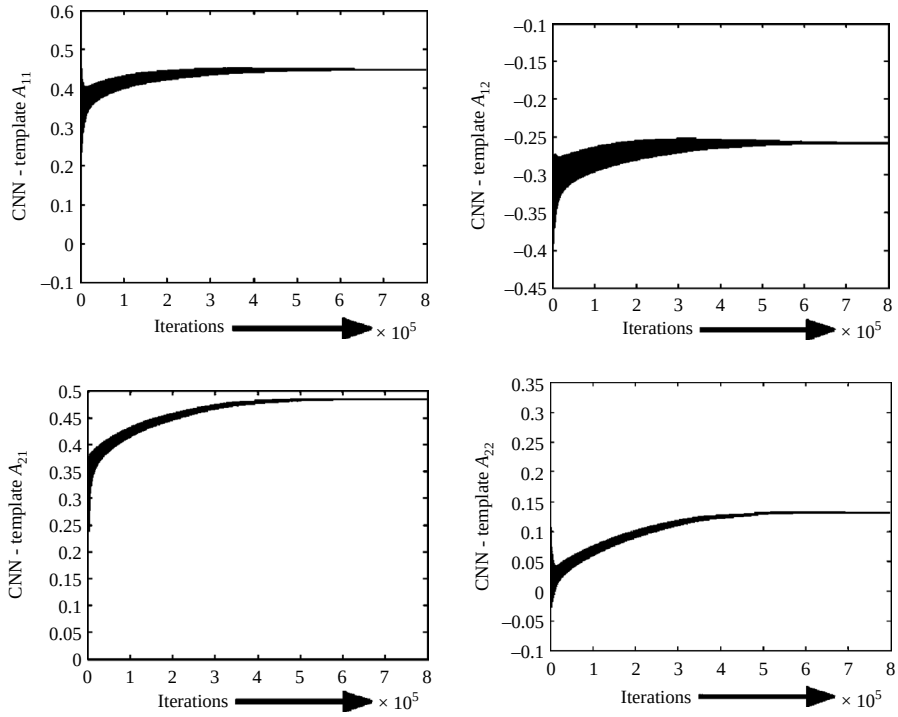


Figure 2.
Convergence of the corresponding state control CNN-templates as achieved by the NOAP process to solve equation (7) with $\varepsilon = 0.25$ and $\omega = 1$

Note: The values of the CNN-templates are: $\hat{A}_{11} = 1.0770$, $\hat{A}_{12} = -0.6300$, $\hat{A}_{21} = 1.3450$ and $\hat{A}_{22} = 0.5850$



Note: The values of the CNN-templates are: $A_{11} = 0.4473$, $A_{12} = -0.2586$, $A_{21} = 0.4846$ and $A_{22} = 0.1310$

Figure 3.
Convergence of the corresponding feedback CNN-templates as achieved by the NOAP process to solve equation (7) with $\varepsilon = 0.25$ and $\omega = 1$

numerical simulation of the same equation, i.e. equation (7) has also been performed using MATLAB and a comparison between these two results is shown in Figure 4. As it clearly shown in Figure 4, the results in Figure 5 and Figure 7, are solutions of equation (7) using the approach based on the CNN-paradigm developed in this paper for $\epsilon = 0.25$ and $\omega = 1$ (Figure 5) and $\epsilon = 1$ and $\omega = 1$ (Figure 7). Similarly, the results of the same equation (equation (7)) obtained through a direct numerical simulation with MATLAB are shown in Figure 6 and Figure 8 for $\epsilon = 0.25$ and $\omega = 1$ (Figure 6) and $\epsilon = 1$ and $\omega = 1$ (Figure 8). Comparing the results obtained by the two methods for different sets of the parameter settings, a very good agreement (i.e. waves having the same value of the amplitude of oscillations and same frequency of oscillations) is obtained. The comparison of (Figure 5 with Figure 6) and (Figure 7 with Figure 8) is a good benchmarking leading to the validation of the NAOP concept developed in this paper.

The method proposed in this paper exploits an optimization concept based on NAOP to derive the CNN-templates of stiff ODEs. Here, the optimization of the CNN-templates is achieved through an offline process. Several strong points of the novel concept developed in this paper (i.e. the optimization based on NAOP) can be derived which are:

- The flexibility of the concept (NAOP) which offers the possibility of mapping any type of nonlinear and stiff equation (ODEs and/or PDEs) into the nonlinear mathematical CNN-model in equation (1a). This offers the possibility of using the CNN-paradigm as a universal solver of nonlinear and stiff equations.

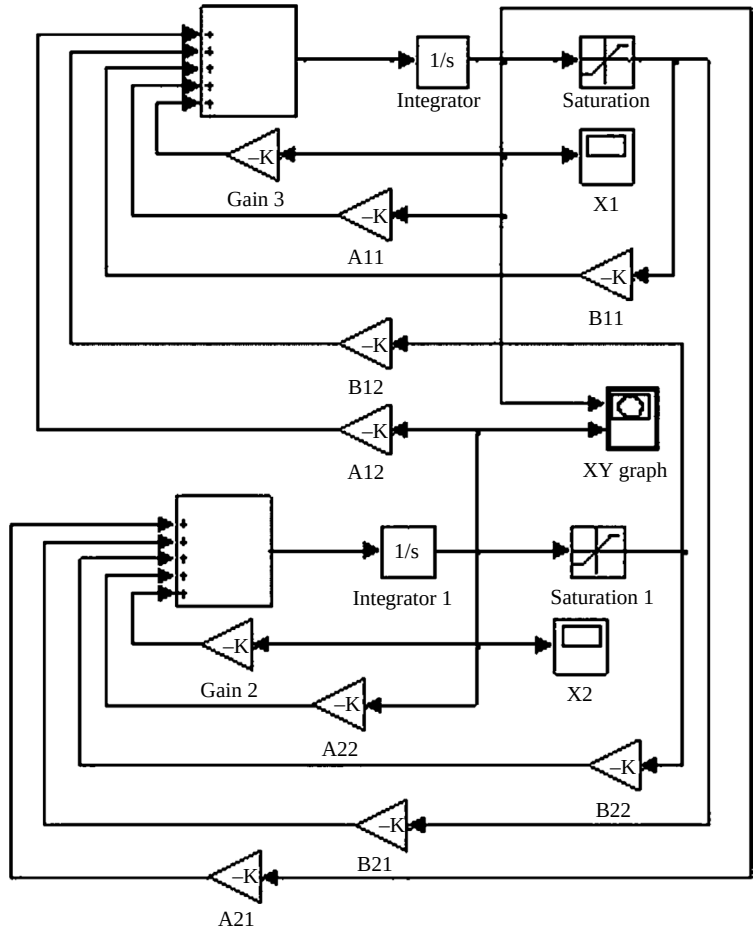


Figure 4.
SIMULINK graphical representation of the CNN-computing platform to solve equation (3)

- The possibility of deriving the CNN-templates through an offline process and the exploitation of these templates for the hardware implementation (FPGA, or emulation on DSP, etc.) of the computing platform based on CNN in order to achieve robust and ultra-fast computing (Chua and Yang, 1988; Zarandy *et al.*, 1999).
- The easy and straight-forward convergence to the local minimum (of the NAOP concept) for any type of nonlinear and stiff ODEs.
- The offline training process is based on the concept of neuron dynamics. This concept transforms the Langrange function described in equation (5) into sets of coupled ODEs of the first order (equation (6)). Therefore, the optimization problem is transformed into solving coupled ODEs of the first order. This allows the possibility of avoiding trial-and-error processes during optimization since a systematic analysis of the stability (i.e. both asymptotic and/or global stabilities) of the coupled ODEs can be performed. The solutions of the coupled models lead to the determination of the CNN-templates.

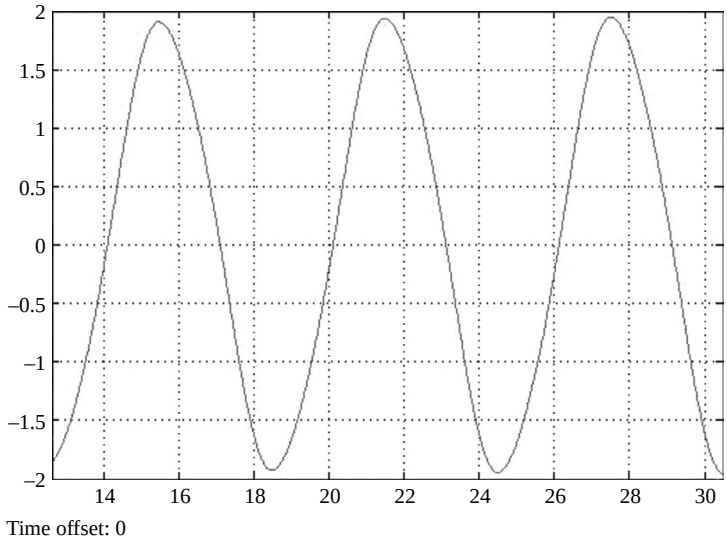


Figure 5.
Wave-form solution of equation (7) obtained by our new approach based on the CNN-paradigm for $\epsilon = 0.25$ and $\omega = 1$

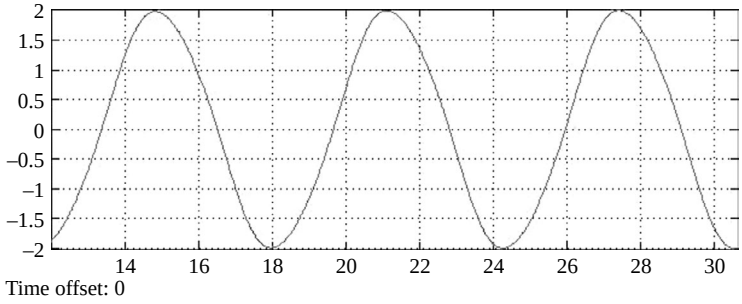


Figure 6.
Wave-form solution of equation (7) obtained through direct numerical simulation in MATLAB for $\epsilon = 0.25$ and $\omega = 1$

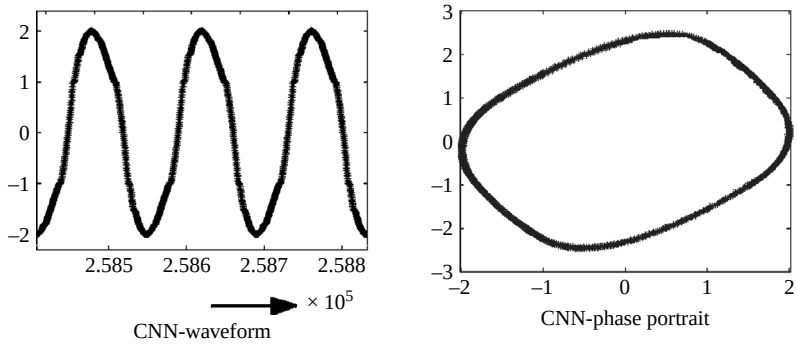


Figure 7.
Wave-form solution of equation (7) and corresponding phase portrait obtained by our new approach based on the CNN-paradigm for $\epsilon = 1$ and $\omega = 1$

- The convergence of (equation (6)) does not depend on initial conditions as this convergence leads to the same templates values regardless initial conditions. Therefore, the NAOP concept is not based on a trial-and-error process as it is the case of the classical optimization methods such as genetic algorithm, reinforcement learning, particle swarm, etc. (Puffer *et al.*, 1995; Aarts and Veer, 2001; Tsoulos *et al.*, 2009; Chua *et al.*, 1995; Puffer *et al.*, 1996; Nossek, 1998).

In our previous publications (Fasih *et al.*, 2008, 2009) we have demonstrated that the hardware implementation is suitable for performing ultra-fast computing and therefore is necessary for performing real-time computation. The method developed in this paper can be easily implemented on hardware in order to perform ultra-fast computing. This is one of the significant advantages (amongst many others) of the method developed in this paper while compared with the computing using Von Neumann architecture (i.e. MATLAB). It is well-known that this last computing platform is time consuming (i.e. very slow) when dealing with the computation of complex nonlinear ODEs and/or PDEs. This limited computing resource/performance is explained by the iterative nature of the computing on Von Neumann and the inherent accumulation of round-off errors which could result to the lack of convergence (Higham, 1996; Hairer *et al.*, 1996).

The method proposed in this paper is challenging as it shows/demonstrates a systematic and straightforward way to solve nonlinear ODEs by the CNN-paradigm. The key challenge has been the possibility and then the appropriate way/algorithmic of/for mapping any type of nonlinearity unto the nonlinearity displayed by the elementary CNN-cell. Therefore, the approach developed in this work is very flexible as it can be applied to solve different types of nonlinear and stiff ODEs. The template calculation scheme based on NAOP has also been successfully applied for solving Rayleigh, Lorenz and Rössler equations and corresponding CNN-templates have been successfully derived. One interesting issue under investigation is the establishment/development of a library of CNN template sets to solve the most common nonlinear and stiff ODEs including the ones already cited above.

The next section is addressing the generalization/extension of the approach developed in this paper to solving nonlinear and stiff PDEs. In fact, it will be shown that a discretization process could help to transform PDEs into sets of coupled or uncoupled nonlinear ODEs in order to make them solvable by the CNN-paradigm while thereby applying the scheme developed in this paper.

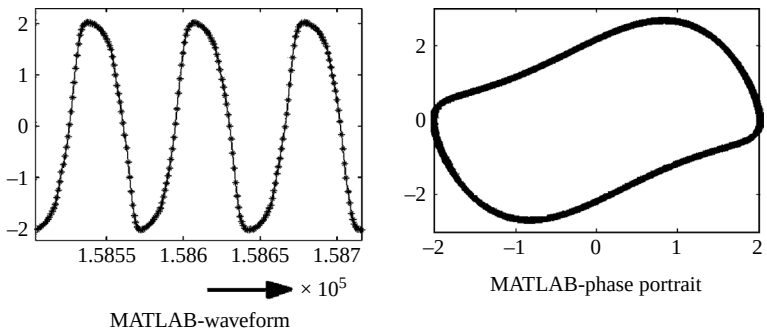


Figure 8.
Wave-form solution of equation (7) and corresponding phase portrait obtained through direct numerical simulation in MATLAB for $\epsilon = 1$ and $\omega = 1$

4. Extension of the NAOP scheme to solving stiff partial differential equations

This section explains the possibility of extending/applying the approach developed in this paper to solving PDEs. Unlike the traditional approach of solving stiff PDEs through CNN which takes into consideration only the linear terms of the Taylor's series expansion, we include the higher order derivative terms in the Taylor's series expansion of any given PDE in order to improve the accuracy of the obtained solutions. We consider, for illustration, the Burger's equation (8) which is a well-known prototype of partial differential equations and which is having multiple potential applications in the field of transportation:

$$\frac{\partial u}{\partial t} = \frac{1}{R}\,\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} \tag{8}$$

In order to solve equation (8) by the CNN-paradigm, applying an expansion (at the first-order) based on the Taylor's series does lead to the following equivalent form of equation (8):

$$\frac{du_i}{dt} = \frac{1}{R}\,\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{u_i[u_{i+1} - u_{i-1}]}{2h} \tag{9}$$

One can see that equation (9) is a well-known prototype of a set of first-order coupled nonlinear ODEs. As it appears in equation (9), the discretization performed has resulted into a set of coupled ODEs with quadratic nonlinear terms (i.e of types similar to Lorenz or Rössler). This type of nonlinearity is solvable by our approach NAOP developed in the preceding paragraph as we could already solve more complex types of nonlinearity (e.g. the nonlinearity in the van der Pol equation). As discussed in Section 1, taking the truncated Taylor's series (only the linear terms) has been done reluctantly in the many published works, since there has been no way so far, according to the literature, to deal with the increased complexity and the nonlinearity that appear otherwise. It is obvious that the results produced in the case of a linear approximation are *de facto* less precise. While considering the higher order (in this case second-order) derivative terms in order to increase precision, the Taylor's series expansion could be applied to equation (8) and this could lead to results presented in equation (10):

$$\begin{aligned}\frac{du_i}{dt} = \frac{1}{R}&\left[\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{u_{i+1} - 3u_i + 3u_{i-1} - u_{i-2}}{2h^2} - \cdots\right] \\ &- u_i\left[\frac{u_{i+1} - u_i}{h} - \frac{u_{i+1} - 2u_i + u_{i-1}}{2h} - \cdots\right]\end{aligned} \tag{10}$$

Therefore, while considering equation (10), it becomes obvious that the NAOP developed in this paper is a best candidate for a straightforward derivation of the appropriate CNN-templates to solve equation (10). In Chedjou and Kyamakya (2009), we have demonstrated the possibility of solving the Burger's equation using the CNN-paradigm.

5. Concluding remarks

We have proposed and validated a theoretical/concept based on the CNN-paradigm for ultra-fast, potentially low-cost and high-precision computing of stiff PDEs and ODEs.

Since we can solve these through CNN independently of the actual nonlinearity, we have reached a clear breakthrough that has the potential to enable a really "real-time" Computational Engineering.

The main benefit of solving PDEs and ODEs using CNN is the offered flexibility through NAOP to extract the CNN parameters through which CNN can solve any type of PDEs or ODEs. Another strong point of the CNN-paradigm is the then resulting ultra-fast processing depending on the CNN implementation: DSP, FPGA, GPU, or CNN-chip. One key objective of this work has been to advance the relevant state-of-the-art by proposing a novel framework to solve stiff PDEs and ODEs with high precision. To achieve this goal, we have proposed and demonstrated that the NAOP technique is a best and efficient scheme to cope with solutions of any ODEs or PDEs. The NAOP is a learning/training method for mapping the wave solutions of the models describing the dynamics of a CNN network to that of a given model (ODEs). Taking just these two inputs, the learning process leads to the convergence to a local minimum where the complete mapping of the two models is achieved and CNN-templates are produced.

Using the same technique, we proposed a high-precision computing of stiff PDEs while accounting even nonlinear terms (i.e. high-order terms) in the Taylor's series expansion used while transforming the PDEs unto a set of coupled nonlinear ODEs. In order to overcome the problem related to the speed of computation, an implementation either on FPGA or DSP or GPU of the concept developed in this work is possible and straight-forward.

An ongoing project under consideration is the design and implementation of a flexible computing platform to be used for real-time Computational Engineering (i.e. ultra-fast solution of stiff ODEs and/or PDEs). Appendix Figure A1 is an illustration of the complete architecture under design. The algorithm developed in this paper (i.e. NAOP) is used at the level of the central server. Indeed, the computing system in Appendix Figure A1 can process many requests submitted online. These requests are stiff ODEs and/or PDEs that will be submitted by customers for deriving their solutions. The requests submitted through the application programming interface will be processed at the level of the central server. A scheduling is planned at the level of the central server in order to allow a sequential processing of multiple job-requests. Appendix Figure A2 shows the basic blocs constituting the central server. Indeed, upon submission of several job requests, the central server will be able to schedule jobs and assign them for processing depending upon specific tasks (e.g. *task 1*: solving ODEs, *task 2*: solving PDEs or *task 3*: solving ODEs and/or PDEs and apply them for image processing, etc.). To achieve real-time Computational Engineering, the central server will use one of these emulation techniques (i.e. DSP, FPGA or GPU) for the processing. The method developed in this paper (i.e. the "NAOP" concept) is therefore the core of the complete architecture in Appendix Figure A1 as this method is based on a systematic and flexible principle to derive the corresponding CNN-templates of stiff ODEs and PDEs. The concept developed (i.e. "NAOP") exploits the CNN-paradigm as this latter is flexible for hardware implementation. Indeed, the graphical representation of the CNN processors for second order ODEs shown in Figure 4 is suitable for rapid prototyping purposes (a hardware implementation in DSP or FPGA or GPU platforms is then straight-forward). Such a computing platform is, therefore, a good prototype for real-time Computational Engineering.

References

Aarts, P.L. and Veer, P. (2001), "Neural network method for solving partial differential equations", *Journal of Neural Processing Letters*, Vol. 14 No. 3, pp. 261-71.

Chedjou, J.C. and Kyamakya, K. (2009), "Use of CNN processors for ultra-fast solution ODE's and PDE: a renaissance of the analog computer", *Proceedings of the XV International Symposium on Theoretical Electrical Engineering, Lübeck, Germany*, pp. 357-60.

Chua, L.O. and Yang, L. (1988), "Cellular neural networks: theory", *IEEE Transactions on Circuits and Systems*, Vol. 35 No. 10.

Chua, L.O., Hasler, M., Moschytz, G.S. and Neirynck, J. (1995), "Autonomous cellular neural networks: a unified paradigm for pattern formation and active wave propagation", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 42 No. 10, pp. 559-77.

Fasih, A., Chedjou, J.C. and Kyamakya, K. (2008), "Implementation of one dimensional CNN array on FPGA-A design based on Verilog HDL", *Proceedings of 1st International Workshop on Nonlinear Dynamics and Synchronization* (*INDS*)*, Aachen*, pp. 31-4.

Fasih, A., Schwarzlmüller, C., Chedjou, J.C. and Kyamakya, K. (2009), "Framework for FPGA based real-time machine vision direct convolution versus CNN", *ISAST Transactions on Electronics and Signal Processing*, Vol. 4, pp. 1-5.

Hairer, E., Norsett, S.P. and Wanner, G. (1996), *Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems*, 2nd ed., Vol. 14, Springer, Berlin.

Higham, N.J. (1996), *Accuracy and Stability of Numerical Algorithms*, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Hopfield, J.J. and Tank, D.W. (1985), "Neural computation of decisions in optimization problems", *Biological Cybernetics*, No. 52, pp. 141-52.

Kozek, T. and Roska, T. (1994), "A double time-scale CNN for solving 2-d Navier-stokes equations", *Proceedings of the 3rd IEEE International Workshop on Cellular Neural Networks and their Applications, Rome, Italy, December*.

Kozek, T., Chua, L.O., Roska, T., Wolf, D., Tetzlaff, R., Puffer, F. and Lotz, K. (1995), "Simulating nonlinear waves and partial differential equations via CNN-part II: typical examples", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 42 No. 10.

Krstic, I., Kandic, D. and Reljin, B. (2003), "Cellular neural networks – an analogous model for stress analysis of prismatic bars subjected to torsion", *FME Transactions*, Vol. 31, pp. 7-14.

Negro, C., Fortuna, L. and Vicari, A. (2005), "Modelling lava flows by cellular nonlinear networks (CNN): preliminary results", *Nonlinear Processes in Geophysics*, Vol. 12, pp. 505-13.

Niu, J.H., Wang, H.Z., Zhang, H.X. and Yan, J.Y. (2001), "Cellular neural network analysis for two-dimensional bioheat transfer equation", *Medical & Biological Engineering & Computing*, Vol. 39, pp. 601-4.

Nossek, J.A. (1998), "Design and learning with cellular neural networks", *International Journal of Circuit Theory and Applications*, Vol. 24, pp. 15-24.

Platt, J.C. and Barr, A.H. (1998), "Constrained differential optimization for neural networks", Scientific Report, Caltech-CS-TR-88-17, California Institute of Technology, Pasadena, CA.

Puffer, F., Tetzlaff, R. and Wolf, D. (1995), "A learning algorithm for cellular neural networks (CNN) solving nonlinear partial differential equations", *Proceedings of the International Symposium on Signals, Systems and Electronics, San Francisco, CA, October*, p. 504.

Puffer, F., Tetzlaff, R. and Wolf, D. (1996), "Modeling nonlinear systems with cellular neural networks", *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, May*, Vol. 6, pp. 3513-16.

Roska, T., Chua, L.O., Wolf, D., Kozek, T., Tetzlaff, R. and Puffer, F. (1995), "Simulating nonlinear waves and partial differential equations via CNN-Part I: basic techniques", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 42 No. 10.

Smith, K.A. (1999), "Neural network for combinatorial optimization: a review of more than a decade of research", *Informs Journal of Computing*, Vol. 11 No. 1, pp. 15-34.

Tsoulos, I.G., Gavrilis, D. and Glavas, E. (2009), "Solving differential equations with constructed neural networks", *Neurocomputing*, Vol. 72, pp. 2385-91.

Uzunova, M., Jolly, D., Nikolov, E. and Boumediene, K. (2008), "The macroscopic LWR model of the transport equation viewed as a distributed parameter system", *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, Cergy-Pontoise*, pp. 572-6.

Wang, Y.F., Cruz, J.B. and Mulligan, J.H. (1993), "Multiple training concept for back-propagation neural networks for use in associative memories", *Neural Networks*, Vol. 6 No. 8.

Zarandy, A., Roska, T., Szolgay, P., Zold, P.S., Foldesy, P. and Petras, I. (1999), "CNN chip prototyping and development system, design automation day on cellular visual microprocessor", *Proceedings of the European Conference on Circuit Theory and Design, Stresa, 29 August-2 September*, pp. 42-58.

Zhu, S.C. and Mumford, D. (1998), "Gibbs reaction and diffusion equations", *Proceedings of the 6th International Conference on Computer Vision, Mumbai*, p. 847.

Further reading

Abassy, T.A., El-Tawil, M.A. and El-Zoheiry, H. (2007), "Exact solutions of some nonlinear partial differential equations using the variational iteration method linked with Laplace transforms and the Padé technique", *Computers and Mathematics with Applications*, Vol. 54, pp. 940-54.

Striebel, M., Bartel, A. and Gunther, M. (2009), "A multirate ROW-scheme for index-1 network equations", *Applied Numerical Mathematics*, Vol. 59, pp. 800-14.

Sweilam, N.H. (2007), "Variational iteration method for solving cubic nonlinear Schrodinger equation", *Journal of Computational and Applied Mathematics*, Vol. 207, pp. 155-63.

Appendix

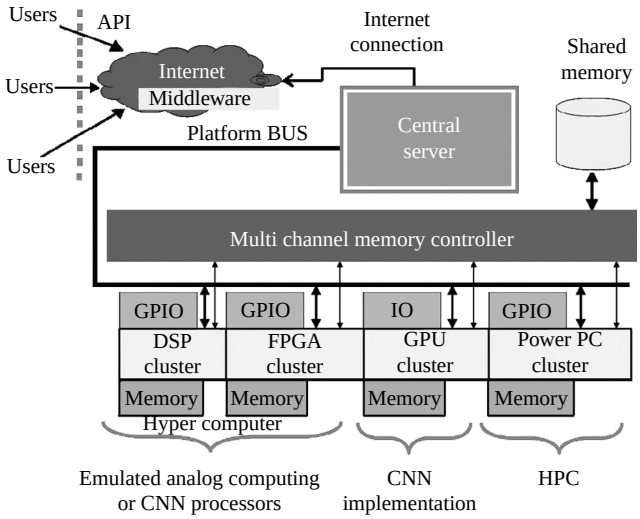


Figure A1.
Global architecture of the computing platform planned to enable a real-time Computational Engineering. Diverse users may access the CNN processor platforms in a remote way through the internet

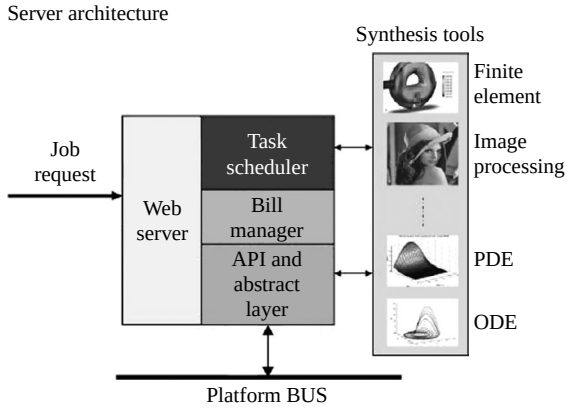


Figure A2.
Core idea of the server architecture intended for the CNN-based super-computing platform to enable real-time Computational Engineering. It is a detailed description of the central sever given in Figure 9

About the authors

J.C. Chedjou received in 2004 his Doctorate in Electrical Engineering at the Leibniz University of Hannover, Germany. He has been a DAAD (Germany) Scholar and also an AUF Postdoctoral Research Fellow. From 2000 to date he has been a Junior Associate Researcher in the Condensed Matter section of the ICTP (Abdus Salam International Centre for Theoretical Physics) Trieste, Italy. Currently, he is a Senior Researcher at the Institute for Smart Systems Technologies of the Alpen-Adria University of Klagenfurt in Austria. His research interests include electronics circuits engineering, chaos theory, analog systems simulation, cellular neural networks, nonlinear dynamics, synchronization and related applications in engineering. He has authored

and co-authored three books and more than 60 journals and conference papers. J.C. Chedjou is the corresponding author and can be contacted at: jean.chedjou@uni-klu.ac.at

K. Kyamakya obtained the "Ir. Civil" degree in Electrical Engineering in 1990 at the University of Kinshasa. In 1999, he received his Doctorate in Electrical Engineering at the University of Hagen in Germany. He then worked three years as Postdoctoral Researcher at the Leibniz University of Hannover in the field of mobility management in wireless networks. From 2002 to 2005, he was a Junior Professor for Positioning Location Based Services at Leibniz University of Hannover. Since 2005, he has been Full Professor for Transportation Informatics and Director of the Institute for Smart Systems Technologies at the University of Klagenfurt in Austria.