



# RK STARTERS FOR MULTISTEP METHODS ON HOLE-FILLER CNN SIMULATION

**S.Senthilkumar**

Department of Mathematics,  
National Institute of Technology,  
Tiruchirappalli-620 015, Tamilnadu, INDIA  
E-mail: [ssenthilkumar1974@yahoo.co.in](mailto:ssenthilkumar1974@yahoo.co.in)

## ABSTRACT

The goal of this paper is focused on developing an efficient design strategy for simulating CNN arrays and hole-filling is implemented using RK step multistep methods to improve the performance of the image or handwritten character recognition. This approach is carried out by analyzing the important features of the hole-filler template and the dynamic process of CNN using well known RK step multistep methods to obtain a set of inequalities satisfying its output characteristics as well as the parameter range of the hole-filler template. Simulation results and comparison have also been presented to show the efficiency of the Numerical Integration Algorithms. In this article, the use of new improved fourth and fifth order linear and nonlinear Runge-Kutta methods in starting procedures for the well known RK step multistep methods is adapted to yield greater accuracy compared with using the standard fourth order Runge-kutta method in the application to problems involving discontinuities and severe gradients where the stepsize is frequently changed.

**Keywords:** *Cellular Neural Networks, Hole-filler template, eight stage seventh order limiting formulas, Ordinary Differential Equations, Non-linear Runge-Kutta, Multi-step methods, Various RK-Fourth Order Means.*

## 1. INTRODUCTION

It is understood that the characteristics of Cellular Neural Networks (CNNs) are analog, time-continuous, non-linear dynamical systems and formally belong to the class of recurrent neural networks. CNNs have been proposed by Chua and Yang [1,2], and they have found that CNN has many important applications in signal and real-time image processing [14]. Roska et al. [3] have presented the first widely used simulation system which allows the simulation of a large class of CNN and is especially suited for image processing applications. It also includes signal processing, pattern recognition and solving ordinary and partial differential equations etc. Runge-Kutta (RK) techniques have become very popular for computational purpose [4,5]. RK algorithms are used to solve differential equations efficiently that are equivalent to approximate the exact solutions by matching 'n' terms of the Taylor series expansion. Oliveria [9] introduced the popular RK-Gill algorithm for evaluation of effectiveness factor of immobilized enzymes. Chi-Chien Lee and Jose Pineda de Gyvez [10] introduced Euler, Improved Euler, Predictor-Corrector and Fourth-Order

(quartic) Runge-Kutta algorithms in Raster CNN simulation. Gear [4], Hindmarsh [5], Krogh [6] and Shampine and Gordon [8] discussed about the starting a multistep method using (k-1) values needed to start a k-step method by knowing the initial value. In this article, the Hole Filling CNN simulation problem is solved with different approach fourth order linear and non linear methods and a fifth order method [15] as starters for solving the explicit fourth and fifth order multistep methods or the s-step Adam formula Lambert [7]

## 2. STRUCTURE AND FUNCTIONS OF CELLULAR NEURAL NETWORK

The general CNN architecture consists of  $M*N$  cells placed in a rectangular array. The basic circuit unit of CNN is called a cell. It has linear and nonlinear circuit elements. Any cell,  $C(i,j)$ , is connected only to its neighbor cells (adjacent cells interact directly with each other). This intuitive concept is known as neighborhood and is denoted by  $N(i,j)$ . Cells not in the immediate neighborhood have indirect effect because of the propagation effects of the dynamics of the network. Each cell

has a state  $x$ , input  $u$ , and output  $y$ . For all time  $t > 0$ , the state of each cell is said to be bounded and after the transient has settled down, a cellular neural network always approaches one of its stable equilibrium points. It implies that the circuit will not oscillate. The dynamics of a CNN has both output feedback (A) and input control (B) mechanisms. The dynamics of a CNN network cell is governed by the first order nonlinear differential equation given below:

$$c \frac{dx_{ij}(t)}{dt} = \frac{-1}{R} x_{ij}(t) + \sum_{c(k,l) \in N(i,j)} A(i,j;k,l) y_{kl}(t) + \sum_{c(k,l) \in N(i,j)} B(i,j;k,l) u_{kl}(t) + I, \quad (1)$$

$$1 \leq i \leq M; 1 \leq j \leq N.$$

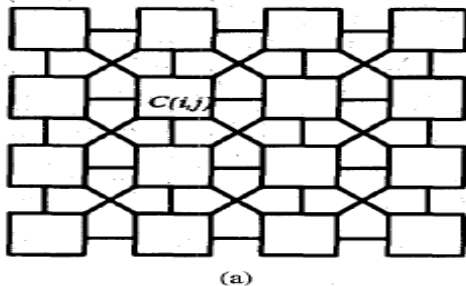
and the output equation is given by,

$$y_{ij}(t) = \frac{1}{2} \left[ |x_{ij}(t) + 1| - |x_{ij}(t) - 1| \right], \quad (2)$$

$$1 \leq i \leq M; 1 \leq j \leq N.$$

where  $c$  is a linear capacitor,  $x_{ij}$  denotes the state of cell  $C(i,j)$ ,  $x_{ij}(0)$  is the initial condition of the cell,  $R$  is a linear resistor,  $I$  is an independent current source,  $A(i,j;k,l)y_{kl}$  and  $B(i,j;k,l)u_{kl}$  are voltage controlled current sources for all cells  $C(k,l)$  in the neighborhood  $N(i,j)$  of cell  $C(i,j)$ , and  $y_{ij}$  represents the output equation. For simulation purposes, a discretized form of equ. (1) is solved within each cell to simulate its state dynamics. One common way of processing a large complex image is using a raster approach [1,2]. This approach implies that each pixel of the image is mapped onto a CNN processor. That is, it has an image processing function in the spatial domain that is expressed

$$\text{as } g(x,y) = T(f(x,y)) \quad (3)$$



where  $g(\cdot)$  the processed image,  $f(\cdot)$  is the input image, and  $T$  is an operator on  $f(\cdot)$  defined over the neighborhood of  $(x,y)$ . It is an exhaustive process from the view of hardware implementation. For practical applications, in the order of 250,000 pixels, the hardware would require a large amount of processors which would make its implementation unfeasible. An different option to this scenario is multiplex the image processing operator.

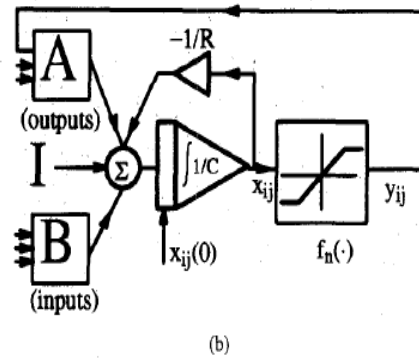


Fig. 1. Cellular Neural Networks: (a) Array Structure and (b) Block Diagram

### 3. HOLE-FILLER TEMPLATE DESIGN

The Hole-Filler is a cellular neural network, which fills up all the holes and remains unaltered outside the holes in a bipolar image. Let  $R_x = 1, C = 1$  and let  $+1$  represent for the black pixel and  $-1$  for the white pixel. If the bipolar image is input with  $U = \{u_{ij}\}$  into CNN and the images having holes enclosed by the black pixels then initial state values are set as  $x_{ij}(0) = 1$ . From the equation (2) the output values are  $y_{ij}(0) = 1, 1 \leq i \leq M, 1 \leq j \leq N$ .

Consider the templates  $A, B$  and the independent current source  $I$  are

$$A = \begin{bmatrix} 0 & a & 0 \\ a & b & a \\ 0 & a & 0 \end{bmatrix}, \quad a > 0, b > \quad (3)$$



$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = -1$$

Where the template parameters  $a$  and  $b$  are to be determined. In order to make the outer edge cells become the inner ones, normally auxiliary cells are added along the outer boundary of the image, and their state values are set to zeros by circuit realization, resulting in the zero output values. The state equation (1) can be rewritten

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{c(k,l) \in N(i,j)} A(i,j;k,l)y_{kl}(t) + 4u_{ij}(t) - I \quad (4)$$

For instance, here the cells  $C(i+1,j)$ ,  $C(i-1,j)$ ,  $C(i,j+1)$  and  $C(i,j-1)$  are the non-diagonal cells. Here, two problems each of the two sub-problems are considered.

**Problem 1:** The input value  $u_{ij} = 1$  for cell  $C(i,j)$ , signaling the black pixel. Because the initial state value of the cell  $C(i,j)$ , has been set to 1,  $x_{ij}(t) = 1$ , and from equation (2) its initial output value is also  $y_{ij}(t) = 1$ . According to the hole-filler demands, its eventual output should be  $y_{ij}(\infty) = 1$ . To obtain this result we set

$$\frac{dx_{ij}(t)}{dt} \geq 0 \quad (5)$$

Substituting this input  $u_{ij} = 1$  and equation (4) into equation (5), we obtain

$$\begin{aligned} \frac{dx_{ij}(t)}{dt} &= -x_{ij}(t) + a[y_{(i-1)j}(t) \\ &+ y_{(i+1)j}(t) + y_{i(j-1)}(t) \\ &+ y_{i(j+1)}(t)] + by_{ij}(t) + 3 \end{aligned} \quad (6)$$

Combining equations (6) and (7) and considering the minimum value of  $x_{ij}(t) = 1$  this case yields

$$\begin{aligned} &a[y_{(i-1)j}(t) + y_{(i+1)j}(t) \\ &+ y_{i(j-1)}(t) + y_{i(j+1)}(t)] \\ &+ by_{ij}(t) + 2 \geq 0 \end{aligned} \quad (7)$$

To facilitate our discussion, two sub-problems namely cell  $C(i,j)$ , inside as well as outside the hole (boundary) are distinguished.

**Sub-problem 1.1:** The cell  $C(i,j)$  is inside the holes. Since  $x_{ij}(0) = 1$  from equation (2) its initial output value  $y_{ij}(0) = 1$ . Considering equations (6) and (7),  $y_{ij}(t) \geq 1$ . According to the hole-filler demands, its initial output of non-diagonal black pixels (grey level) should not be changed inside the holes. The weighting coefficients of  $a$  and  $b$  are equal to +4 and +1, respectively. Since  $A(i,j;i,j) > 1/R_x$  the parameter  $b$  is found to be  $b > 1$ , that is

$$\begin{aligned} 4a+b+2 &\geq 0 \\ b &> 1 \end{aligned} \quad (8)$$

**Sub-problem 1.2:** The cell  $C(i,j)$ , is outside the holes. To satisfy equation (8), we need to check only the minimum value on the left-hand side of equation (8). This is true when there are four non-diagonal white pixels around the cell  $C(i,j)$ , where the weighting coefficient of  $a$  in equation (8) is -4. Since  $y_{ij}(t) \geq 1$  the weight of  $b$  is equal to 1. Combining this with  $b > 1$  gives

$$\begin{aligned} 4a+b+2 &\geq 0 \\ b &> 1 \end{aligned} \quad (9)$$

**Problem 2:** The input value of cell  $C(i,j)$  is  $u_{ij} = 1$ , signaling the white pixel. Substituting this input value in equation (5) gives

$$\begin{aligned} \frac{dx_{ij}(t)}{dt} &= -x_{ij}(t) + a[y_{(i-1)j}(t) \\ &+ y_{(i+1)j}(t) + y_{i(j-1)}(t) \\ &+ y_{i(j+1)}(t)] + by_{ij}(t) - 5 \end{aligned} \quad (10)$$

**Sub-problem 2.1:** The cell  $C(i,j)$ , is inside the holes. Since  $x_{ij}(0) = 1$ , from equation (2) its initial output value is  $y_{ij}(t) = 1$ . According to the hole-filler demands, the holes should be filled by the black pixels, whereas its initial black pixels remain unaltered:

$$\frac{dx_{ij}(t)}{dt} \geq 0 \quad (11)$$

Combining equations (11) and (12) and considering  $x_{ij}(t) \geq 1$  yields



$$\begin{aligned}
 & a[y_{(i-1)j}(t) + y_{(i+1)j}(t) \\
 & + y_{i(j-1)}(t) + y_{i(j+1)}(t)] \quad (12) \\
 & + by_{ij}(t) - 6 \geq 0
 \end{aligned}
 \begin{array}{l}
 3a+b-6 < 0 \\
 b > 1
 \end{array} \quad (16)$$

where we use the minimum value of in equation (11). Since the cell is inside the holes, its initial output of non-diagonal black pixels remain unchanged. The coefficients of a and b are equal to +4 and +1, respectively. Combining this with  $b > 1$  gives

$$\begin{aligned}
 & 4a+b-6 \geq 0 \quad (13) \\
 & b > 1
 \end{aligned}$$

**Sub problem 2.2:** The cell  $C(i,j)$  is outside the holes. Since  $x_{ij}(0)=1$ , from equation (2) its initial output value is  $y_{ij}(0) = 1$ . According to the hole-filler demands, the final output of this cell should be white, and in this case  $y_{ij}(\infty) = -1$

$$\frac{dx_{ij}(t)}{dt} \geq 0 \quad (14)$$

Combining equations (11) and (15) and considering  $x_{ij}(t) \leq 1$  we get

$$\begin{aligned}
 & a[y_{(i-1)j}(t) + y_{(i+1)j}(t) \\
 & + y_{i(j-1)}(t) + y_{i(j+1)}(t)] \quad (15) \\
 & + by_{ij}(t) - 6 < 0
 \end{aligned}$$

where we use the maximum value of  $x_{ij}(t)$  in equation (11)

The initial condition is  $y_{ij}(0) = 1$ . Now the problem arises how the output of cell  $C(i,j)$ , be changed to -1 and where does this change to begin. First we consider the situation where the change begins from the inside of the bipolar image. If the maximum value on the left-hand side in equation (16) is less than zero, equation (16) holds. Inside the image and outside the holes, the maximum weights of a and b are +4 and +1, respectively. This case was described by equation (14). In fact, the change of the output of the cell  $C(i,j)$ , is like a wave propagating from the edges to the inside of the image and it is verified from the simulated result. Therefore, we should first consider the edge cell  $C(i,j)$ ,  $i=1$  or  $M$ ,  $j=1$  or  $N$ . For this the maximum weight of a in equation (16) is +3, which is also the maximum weight of a outside the holes. The maximum weight of b is +1, occurring at the initial time:

Combining Problems 1 and 2, we obtain

$$\begin{aligned}
 & 3a+b-6 < 0 \\
 & 4a+b+2 \geq 0 \\
 & 4a+b-6 \geq 0
 \end{aligned} \quad (17)$$

#### 4. NUMERICAL INTEGRATION TECHNIQUES

The CNN is described by a system of nonlinear differential equations. Therefore, it is necessary to discretize the differential equation for performing behavioral simulation. For computational purposes, a normalized time differential equation describing CNN is used by Nossek et al.,[11]

$$\begin{aligned}
 f'(x(n\tau)) &= \frac{dx_{ij}(n\tau)}{dt} = -x_{ij}(n\tau) \\
 &+ \sum_{c(k,l) \in N_r(i,j)} A(i,j;k,l)y_{kl}(n\tau) \quad , \\
 &+ \sum_{c(k,l) \in N_r(i,j)} B(i,j;k,l)u_{kl} + I \\
 &1 \leq i \leq M; 1 \leq j \leq N \\
 y_{ij}(n\tau) &= \frac{1}{2} \left[ |x_{ij}(n\tau) + 1| - |x_{ij}(n\tau) - 1| \right], \\
 &1 \leq i \leq M; 1 \leq j \leq N
 \end{aligned} \quad (18)$$

where  $\tau$  is the normalized time. For the purpose of solving the initial-value problem, well established Single Step methods of numerical integration techniques are used [13].

These methods can be derived using the definition of the definite integral

$$\begin{aligned}
 & x_{ij}((n+1)\tau) - x_{ij}(n\tau) = \\
 & \int_{\tau_n}^{\tau_{n+1}} f'(x(n\tau))d(n\tau) \quad (19)
 \end{aligned}$$

The Fourth-Order (quartic) Runge-Kutta is the most widely used single step algorithm in the CNN behavioral raster simulation. The method vary in the way they evaluate the integral presented in (19).

#### 4.1 Fourth Order RK Method Based on Different Means

Traditionally the numerical solution of initial value problem such as single step methods use



information from only the last computed point  $(x_i, y_i)$  to evaluate  $(x_{i+1}, y_{i+1})$ . In contrast, multistep methods use information from several previous points  $(x_i, y_i), (x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2}), \dots$ . It is apparent that the multistep methods cannot be used until a few points in the numerical solution are obtained. To obtain these several previous values, a linear or non-linear method as a starter for solving by a multistep method can be used. As a result, for the four-step Adam's formula, it is possible to apply fourth order linear or non-linear methods using a variety of means [16] which can be written in the form

$$y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^3 \text{Means} \right] \quad (20)$$

Where

means = principal means include the arithmetic mean (AM), Geometric mean (GM), Contraharmonic mean (CoM), Centroidal mean (CeM), Root Mean Square (RMS), Harmonic mean (HM), and Heronian mean (HeM) which involve  $k_i, 1 \leq i \leq 4$ .

In table 1 the term 'h' represents step size and  $k_1, k_2, k_3,$  and  $k_4$  can be expressed as,

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + a_1 h, y_n + h a_1 k_1)$$

$$k_3 = f(x_n + (a_2 + a_3)h, y_n + h a_2 k_1 + h a_3 k_2)$$

$$k_4 = f(x_n + (a_4 + a_5 + a_6)h, y_n + h a_4 k_1 + h a_5 k_2 + h a_6 k_3)$$

**TABLE 1.** Rank (Less Error in the order of Ascending): Fourth Order RK with Various Means.

Different Means	$y_{n+1} =$
AM	$y_n + \frac{h}{6} [k_1 + 2(k_2 + k_3) + k_4]$
CM	$y_n + \frac{2h}{9} \left[ \frac{k_1^2 + k_1 k_2 + k_2^2}{k_1 + k_2} + \frac{k_2^2 + k_2 k_3 + k_3^2}{k_2 + k_3} + \frac{k_3^2 + k_3 k_4 + k_4^2}{k_3 + k_4} \right]$

HM	$y_n + \frac{2h}{3} \left[ \frac{k_1 k_2}{k_1 + k_2} + \frac{k_2 k_3}{k_2 + k_3} + \frac{k_3 k_4}{k_3 + k_4} \right]$
CoM	$y_n + \frac{h}{3} \left[ \frac{k_1^2 k_2^2}{k_1 + k_2} + \frac{k_2^2 k_3^2}{k_2 + k_3} + \frac{k_3^2 k_4^2}{k_3 + k_4} \right]$
HeM	$y_n + \frac{h}{9} [k_1 + 2(k_2 + k_3) + k_4 + \sqrt{ k_1 k_2 } + \sqrt{ k_2 k_3 } + \sqrt{ k_3 k_4 }]$
GM	$y_n + \frac{h}{3} [\sqrt{ k_1 k_2 } + \sqrt{ k_2 k_3 } + \sqrt{ k_3 k_4 }]$
RMS	$y_n + \frac{h}{3} \left[ \sqrt{\frac{k_1^2 + k_2^2}{2}} + \sqrt{\frac{k_2^2 + k_3^2}{2}} + \sqrt{\frac{k_3^2 + k_4^2}{2}} \right]$

Where, the given initial value problem is

$$\dot{y} = \frac{dy}{dx} = f(x, y) \text{ and } a_1, a_2, a_3, a_4, a_5 \text{ and } a_6 \text{ are}$$

known constants on the type of fourth order RK methods.

Five-step Adam's formula is adapted using a new fifth order arithmetic mean (AM) method [15] which can be written as

$$y_{n+1} = y_n + h \left[ \sum_{i=1}^4 w_i \left( \frac{k_i + k_{i+1}}{2} \right) \right] \quad (21)$$

where

$$\begin{aligned} k_1 &= f(y_n) \\ k_2 &= f(y_n + h a_1 k_1) \\ k_3 &= f(y_n + h a_2 k_1 + h a_3 k_2) \end{aligned} \quad (22)$$

$$k_4 = f(y_n + h a_4 k_1 + h a_5 k_2 + \left( \frac{1}{2} - a_4 - a_5 \right) h k_3)$$

$$k_5 = f(y_n + h a_6 k_1 + h a_7 k_2 + h a_8 k_3 + (1 - a_6 - a_7 - a_8) h k_4)$$

Wazwaz [16] has performed numerical tests for several problems of IVPs in the form of  $y' = f(x, y)$  using fourth order RK formulas based on a variety of means. He has concluded that the level of accuracy in each formula adapted depends upon the error terms which mainly depend on the nature of the function  $f(x,y)$ . It is of interest to mention that no investigation has so far been performed and the level of accuracy of each fourth order RK formulas on the problem of hole-filler cellular numerical network simulation. In view of this a modest effort has been made in the present paper to investigate the efficiency of the various RK formulas based on variety means in the problem of hole-filler cellular numerical network simulation.

**4.2 EXPLICIT FOURTH AND FIFTH ORDER MULTISTEP ADAM'S METHOD**

A well known four - step explicit Adam-Bashforth method [7] is of the form

$$y_{n+1} = y_n + \frac{h}{24} [55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}] \tag{23}$$

To solve the above equation it is necessary to have  $f_i = 0,1,2,3$  by using fourth order method in Eq. (5).

The five-step fifth order multistep Adam's method is of the form

$$y_{n+1} = y_n + \frac{h}{720} [1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} + 251f_{n-4}] \tag{24}$$

To solve the above equation it is necessary to have  $f_i = 0,1,2,3,4$  by using fourth order method in Eq. (6) and Eq. (7).

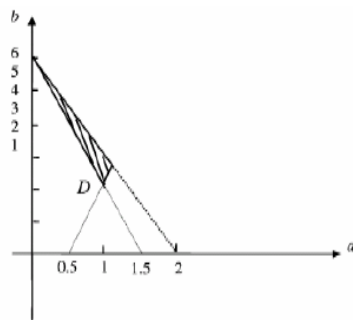


Figure. 2 Range of the Template

Step Size	Euler Method	Euler Method
(h)	(Ts)	(Tc)
0.5	2.5	6.5
0.6	14.7	15.5
0.7	28.3	32.5
0.8	30.7	35.0
0.9	32.6	36.8
1.0	33.6	37.9
1.5	36.8	44.8
2.0	43.2	47.4
2.5	45.6	50.6
3.0	49.3	53.5

Modified Euler	Modified Euler	RK-Fourth order
(Ts)	(Tc)	(Ts)
2.4	6.0	2.4
13.7	15.0	12.5
27.4	32.0	27.2
30.0	34.6	29.6
32.0	36.6	31.6
33.0	37.6	32.8
36.2	44.7	36.0
43.0	46.4	42.8
44.5	50.4	44.3
49.0	52.1	48.2

RK-Fourth order	RK-Adams	RK-Adams
(Tc)	(Ts)	(Tc)
5.8	2.33	4.7
11.4	11.2	10.3
30.0	25.5	28.4
32.4	28.1	30.5
34.2	30.1	32.3
36.2	31.0	34.5
43.1	34.6	41.4
46.2	40.9	44.5
49.3	42.5	47.5
52.0	46.4	50.5

Table 1: Comparative simulated results of the Hole-Filler Template Design using different numerical integration techniques

where Ts and Tc represents Settling Time and Computation Time.



Figure.3 Image before and after Hole-Filling .

## 5. SIMULATION RESULTS AND COMPARISONS

All the simulated outputs presented below here are performed using a high power workstation, and the simulation time used for comparisons is the actual CPU time used. The Settling time  $T_s$  and computation time  $T_c$  for different step sizes are presented in the Table-1 for the purpose of comparison. The settling time  $T_s$  is the time from start of computation until the last cell leaves the interval  $[-1.0, 1.0]$  which is based on a specified limit (e.g.,  $|dx / dt| < 0.01$ ). The computation time  $T_c$  is the time taken for settling the network and adjusting the cell for proper position once the network is settled. The simulation shows the desired output for every cell. We use +1 and -1 to indicate the black and white pixels, respectively. We have marked the selected template parameters  $a$  and  $b$ , which are restricted to the shaded area as shown in Figure-2 for the simulation. The speed is one of the major concerns in the simulation. Hence, determination of the maximum step size that still

yields convergence for a template can be helpful in speeding up the system. The speed-up can be achieved by selecting an appropriate  $(\Delta t)$  for that particular template. The table1 show the Comparative simulated results of the Hole-Filler Template Design using different numerical integration techniques.

## 6. CONCLUSION

In this paper a different numerical integration algorithms are used for Hole Filling CNN simulation. The importance of the simulator is capable of performing Hole-Filling simulation for any kind as well as any size of input image or hand written character. It is a powerful tool for researchers to investigate the potential applications of CNN. It is pertinent to pin-point out here that fourth and fifth order linear and nonlinear methods outperforms well in comparison with the Explicit Euler, RK-Gill and various means. It is of interest to mention that using the fourth and fifth order linear and nonlinear multi-step method is proved to be feasible and effective by theoretic analysis and simulation. It is observed that the hole is filled and the outside image remains unaffected that is, the edge of the images are preserved intact. The templates of the cellular neural network are not unique and this is important in its implementation. The significance of this research work (hole-filler) is to improve the performance of handwritten character recognition. In many language scripts, numerals and in images etc there are many holes and the CNN described above can be used in addition to the Connected Component Detector. As a remarkable note that that each pixel receives the information only from its immediate neighbors, while the result  $u_{xij}(\infty)$  is completely global in nature.

## REFERENCES

- [1]. Chua, L.O., Yang, L.: Cellular Neural Networks: Theory, IEEE Transactions on Circuits and Systems, 35 (1988) 1257 – 1272.
- [2]. Chua, L.O., Yang, L.: Cellular Neural Networks: Applications, IEEE Transactions on Circuits and Systems, 35 (1988) 1273 – 1290.
- [3]. Roska et al.: CNNM Users Guide, Version 5.3x, Budapest (1994).



- [4] Gear, C. W. (1980). Runge-Kutta Starters For Multistep Methods, ACM Transactions on Mathematica Software, 6(3), 263-279.
- [5] Hindmarsh, A. C. (1974). GEAR: Ordinary Differential Equation Solver, Rep. UCID-30001, Rev. 3, Lawrence Livermore Lab., Livermore, California.
- [6] Krogh, F. T. (1972). VODQ/SVDQ/DVDQ-Variable Order Integrators for the Numerical Solution of Ordinary Differential Equations. Sec. 314, Subroutine Write-up, Jet Propulsion Lab., Pasadena, California.
- [7] Lambert, J. D. (1991). Numerical Methods for Ordinary Differential Systems, John Wiley and Sons, London.
- [8] Shampine, L. F. and Gordon, M. K. (1975). Computer Solution of Ordinary Differential Equations: The Initial Value Problem, Freeman, San Francisco.
- [9]. Oliveira, S.C.: Evaluation of effectiveness factor of immobilized enzymes using Runge-Kutta-Gill method: how to solve mathematical undetermination at particle center point?, Bio Process Engineering, 20 (1999) 185-187.
- [10]. Chi-Chien Lee, Jose Pineda de Gyvez: Single-Layer CNN Simulator, International Symposium on Circuits and Systems, 6 (1994) 217 – 220.
- [11]. Nossek, J.A., Seiler, G., Roska, T., Chua, L.O.: Cellular Neural Networks: Theory and Circuit Design, Int. J. of Circuit Theory and Applications, 20, (1992) 533-553.
- [12]. L.O. Chua and T. Roska, “The CNN Universal Machine Part 1: The Architecture”, in Int. Workshop on Cellular Neural Networks and their Applications (CNNA), (1992), 1-10.
- [13]. W. H. Press, B. P. Flannery, S.A. Teukolsky, and W.T. Vetterling, “Numerical Recipes. The Art of Scientific Computing”, Cambridge University Press, New York, (1986).
- [14]. Rafael .C. Gonzalez, Richard .E. Woods
- Steven .L. Eddin, “Digital Image Processing using MATLAB”, Pearson Education Asia, 2005.
- [15] Evans, O. J. and Yaakub, A. R., A New Fifth Order Weighted Runge-kutta Formula, Intern. J. Computer Math., (1996), 1-17
- [16] Abdul-Majid Wazwaz , “A Comparison of Modified Runge-Kutta formulas based on Variety of means”, International Journal of Computer Mathematics, 1994, 50, 105-112.