A Study on Third Order Runge-Kutta Techniques for Solving Practical Problems

Sukumar SENTHILKUMAR*

School of Computing Science and Engineering, Vellore Institute of Technology University, Vellore 632014, Tamilnadu, India

(*Corresponding author; e-mail: senthilkumar@vit.ac.in; ssenthilkumar1974@yahoo.co.in)

Received: 23 December 2013, Revised: 6 January 2014, Accepted: 25 June 2014

Abstract

In this paper, an analysis has been carried out to examine Nystrom third order, Heun third order and Classical Runge-Kutta third order methods to solve image processing and numerical problems which are demonstrated in brief. The methods adapted are fully capable to cope with the linearity and nonlinearity of the physical problems with versatile physical nature. Example problems and its corresponding results are exhibited which reveal the efficiency and reliability of the employed techniques. Furthermore, validity of an obtained solution is verified in comparison with the simulation output for an image processing problem and numerically computed results for an engineering problem and initial value problems.

Keywords: Nystrom third order, Heun third order, Classical Runge-Kutta third order, Initial value problem, Advanced fuzzy cellular non-linear / neural network, Image processing, Simulation.

Introduction

It is well known from the literature that differential equations appear in many areas of science and engineering. The basic principle of mathematical science is that, in order to solve a new problem, reduce it into a problem that has already been solved. Moreover, many problems in engineering and science can be put in mathematical problems especially into differential equations. The numerical computations / simulations are powerful tools to elucidate the characters and rules of nonlinear systems. They also serve as numerical experiments since real time experiments are sometimes impossible to carry out practically. Due to rapid increase in computational power, numerical simulations have been sophisticated to achieve higher spatial resolution, wider dynamic range, and inclusion of many more factors and effects. Many differential equations cannot be solved analytically; however, in science and engineering, a numeric approximation to the solution is often good enough to solve complex problems.

It is always better to obtain an exact solution for the given differential equations but, due to some complications like time consumption and more manual operations, it is not possible to find analytical solutions for such mathematical problems. Therefore, it is necessary to approximate (numerical) solutions. There are so many numerical methods available for solving such differential equation such as power series method, pointwise method, Taylor's method, Picard's method, Euler's method, Improved Euler's method, Modified Euler's method, Runge-Kuta second and fourth order method, predictor corrector method etc. Runge-Kutta (RK) techniques have become a very popular and efficient tool for computational purposes [1-4] due to many real time application problems are solved effectively. Particularly RK algorithms are adapted to solve differential equations efficiently that are equivalent to approximate the exact solutions by matching 'n' terms of the Taylor series expansion. Harrer *et al*. [5] introduced Explicit Euler, Predictor-Corrector and fourth-order Runge-Kutta algorithms which are used for simulating cellular neural networks. Lee and de Gyvez [6,7] introduced Euler, Improved Euler, Predictor-Corrector and Fourth-Order (quartic) Runge-Kutta algorithms for raster and time-multiplexing

CNN simulation. In this article, in order to achieve better results, AFCNN architecture and explicit third order RK algorithms are implemented exclusively for an image processing problem but on the other hand computer engineering numerical problem and initial value problems are solved directly using explicit third order Nystrom, third order Heun and third order Classical Runge-Kutta methods.

A brief note on explicit third order Classical Runge-Kutta, Nystrom and Heun methods

Consider a solution for an initial value problem of the form;

$u_t = f(t,u)$ by the s-stage Runge-Kutta method; (1)

$$u_{n+1} = u_n + h\sum_{i=1}^{s} b_i k_i \tag{2}$$

$$k_i = f(t_n + c_i h, u_n + h\sum_{l=1}^{s} a_{il} k_l \tag{3}$$

$$c_i = \sum_{j=1}^{s} a_{ij} \tag{4}$$

where b_i, c_i, a_{ij} are determined by the employed method. These coefficients are usually written in the Butcher array as;

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \tag{5}$$

It is significant to note that, if *A* is strictly lower triangular, the RK method is explicit. But, if *A* is lower triangular the RK method is semi-implicit, and if *A* is not lower triangular the RK method is implicit.

Third order explicit Classical Runge-Kutta method

The Butcher representation for general third order explicit Runge-Kutta method is given by;

$$\begin{array}{c|ccc} c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & W_1 & W_2 & W_3 \end{array} \tag{6}$$

In particular, the corresponding third order explicit classical Runge-Kutta method is;

$$\begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array} \tag{7}$$

Similarly, it is possible to construct different types of higher order optimal or nearly optimal explicit Runge-Kutta methods according to user needs for solving various real time application problems.

Third order explicit Classical Kutta method

$$\begin{array}{c|ccc} \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \tag{8}$$

Third order explicit Nystrom method

$$\begin{array}{c|ccc} \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{2}{8} & \frac{3}{8} & \frac{3}{8} \end{array} \tag{9}$$

Third order explicit Heun method

$$\begin{array}{c|ccc} \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \tag{10}$$

Employing explicit third order methods for image processing problem

The dynamics of a standard cellular neural network with a neighborhood of radius r are governed by a system of $n = MN$ differential equations [6-16]. CNN is a dynamic nonlinear system defined by coupling only identical simple dynamical systems called cells located within a prescribed sphere of influence, such as nearest neighbors. Fuzzy cellular neural network (FCNN) is a generalization of cellular neural networks (CNNs) by employing fuzzy operations in the synaptic law computation allowing the end user to combine the low level information processing capability of CNN's with the high level information processing capability, such as image understanding, of fuzzy systems. Fuzzy sets [7-24] provide a problem solving tool between the precision of classical mathematics and the inherent imprecision of the real world. The advanced / type-II fuzzy cellular non-linear network (AFCNN) is given below [17-27];

$$c\frac{dx_{ij}(t)}{dt}=\frac{-1}{R_x}x_{ij}(t)+\sum_{c(k,l)\in N_r(i,j)}A(i,j;k,l)y_{kl}(t)+\sum_{c(k,l)\in N_r(i,j)}B(i,j;k,l)u_{kl}(t)$$
$$+I+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigwedge}}A_{f\min}(i,j;k,l)y_{kl}+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigvee}}A_{f\max}(i,j;k,l)y_{kl}(k,l)+ \tag{11}$$
$$+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigwedge}}B_{f\min}(i,j;k,l)u_{kl}+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigvee}}B_{f\max}(i,j;k,l)u_{kl}(k,l)$$

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

Hence, the equation can be recasted as the 2D convolution representation.

$$c\frac{dx_{ij}(t)}{dt}=\frac{-1}{R_x}x_{ij}(t)+\sum_{c(k,l)\in N_r(i,j)}A(i,j;k,l)y_{kl}(t)+\sum_{c(k,l)\in N_r(i,j)}B(i,j;k,l)u_{kl}(t)$$
$$+I+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigwedge}}A_{f\min}(i,j;k,l)y_{kl}+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigvee}}A_{f\max}(i,j;k,l)y_{kl}(k,l)+ \tag{12}$$
$$+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigwedge}}B_{f\min}(i,j;k,l)u_{kl}+\underset{c(k,l)\in N_r(i,j)}{\widetilde{\bigvee}}B_{f\max}(i,j;k,l)u_{kl}(k,l)$$

where A_{fmin}, A_{fmin}, B_{fmin} and B_{fmax} are the feedback MIN, feedback Max, feedforward MIN and feedforward MAX templates respectively. Θ_{max} represents a 2D operation.

$$A_{f\max}\Theta_{\max}y_{ij}=\underset{C_{kl}\in N_r(i,j)}{\widetilde{\bigvee}}A_{f\max}(i,j;k,l)y_{kl}$$

and Θ_{min} indicates (13)

$$A_{f\min}\Theta_{\min}y_{ij}=\underset{C_{kl}\in N_r(i,j)}{\widetilde{\bigvee}}A_{f\min}(i,j;k,l)y_{kl}$$

Simulation results for an image processing problem

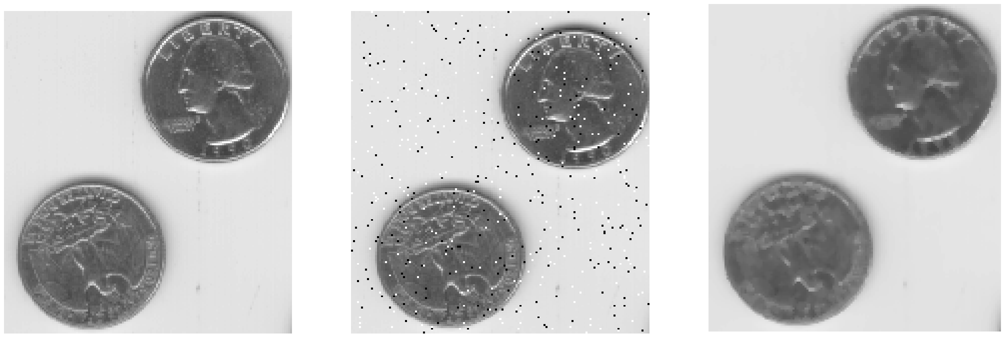
Digital image processing is the use of computer algorithms to perform complex or complicated image processing operations on digital images [25]. A detailed discussion on single layer / raster scheme and time-multiplexing approach for edge detection using a cellular neural network paradigm by the new fourth order four stage algorithms is given by Senthilkumar [26]. Furthermore, raster / single layer simulation using advanced fuzzy cellular neural network is also carried out by Senthilkumar [27] to obtain edge detection results for any given images. Digital images are often corrupted by random variations in intensity, illumination, or have poor contrast and cannot be used directly. Filtering: transform pixel intensity values to reveal certain image characteristics such as Enhancement improves contrast, Smoothing removes noises and Template matching detects known patterns [28].

Task prescription: The template filters out the noises in the given image [29].

$$\text{Template set: } A=\begin{matrix}0&0&0\\0&0&0\\0&0&0\end{matrix};\ B=\begin{matrix}b_{-1,-1}&b_{-1,0}&b_{-1,1}\\b_{0,-1}&b_{0,0}&b_{0,1}\\b_{1,-1}&b_{1,0}&b_{1,1}\end{matrix};\ B=\begin{matrix}1&1&1\\1&0&1\\1&1&1\end{matrix};\ I=1 \tag{14}$$

Global task:
Given: Static gray scale-image P;
Input: U(t) = P;
Initial State: X(0) = 0;
Boundary Conditions: Fixed type, $u_{ij} = 0$ for all virtual cells, denoted by [U] = 0.
Output: Y(t) => Y(∞) = Filtered image (**Figures 1 & 2**).

Local rules: $u_{i,j}(0) \rightarrow y_{ij}(\infty)$



(a) (b) (c)

Figure 1 (a) Input image, (b) noisy image and (c) output image.

(a) (b) (c)

Figure 2 (a) Original image, (b) noisy image and (c) filtered image;.

Solving numerical problems by explicit third order classical Runge-Kutta, Nystrom and Heun methods

Let us consider a simple problem $y' = y - t^2 + 1; y(0) = 0.5$ to compute approximate solution using explicit Nystrom third order method. The corresponding exact solution is given by;

$$y = t^2 + +2t + 1 - \frac{1}{2}e^t; 0 \le t \le 2. \quad (15)$$

Table 1 Comparison between exact and numerical solution.

Time (t)	Exact solution	Numerical solution
0.00	0.5	0.5
0.50	1.4256	1.42513020833333
1.00	2.6408	2.63960266113281
1.50	4.0091	4.00681897004445
2.00	5.3054	5.30160522926598

Initial Value problem and outputs

Let us consider a simple problem $y' = (x - y)/2, y(0) = 1$ to compute approximate solution using the explicit Heun third order method.

Table 2 Comparison between exact and numerical solution.

Time (t)	Exact solution	Numerical solution
0.00	1.0	1.0
0.25	0.8974	0.897491455078125
0.50	0.8364	0.8364036682372292
0.75	0.8118	0.8118695824237503
1.00	0.8195	0.8195940336507935
1.25	0.8557	0.8557865519338713
1.50	0.9171	0.9171020583080967
1.75	1.0005	1.0005885301147697
2.00	1.1036	1.103640815765855
2.25	1.2239	1.2239598764051842
2.50	1.3595	1.3595168167905574
2.75	1.5085	1.5085211426496503
3.00	1.6693	1.669392747887015

It is significant to note that, up to some extent the numerical solution matches the exact solution (**Table 1** and **2**). If *h* is small one can obtain almost an equivalent to the exact solution but it takes more time to complete its task (**Table 3**).

Computer engineering numerical problem and Solutions

A rectifier-based power supply requires a capacitor to temporarily store power when the rectified waveform from the AC source drops below the target voltage. To properly size this capacitor a first-order ordinary differential equation must be solved. For a particular power supply, with a capacitor of 150 µF, the ordinary differential equation to be solved is;

$$\frac{dv(t)}{dt} = \frac{1}{150 \times 10^{-6}} \left\{ -0.1 + \max\left(\frac{|18\cos(120\pi(t))| - 2 - v(t)}{0.04}, 0 \right) \right\} \quad (16)$$

$$v(0) = 0$$

Using a third order method, we found the voltage across the capacitor at $t = 0.00004\,\text{s}$ beginning at a step-size $h = 0.00002\,\text{s}$ (**Table 3**).

Table 3 Value of voltage at time, $t = 0.00004\,\text{s}$ for different step-sizes (h).

Step-size (*h*)	**Numerical solution V (0.00004)**
0.0000025	15.976
0.000005	15.975
0.00001	15.986
0.00002	26.647
0.00004	53.335

Conclusions

In order to solve image processing and numerical problems an analysis have been carried out using third order Nystrom, third order Heun and third order classical Runge-Kutta methods. Numerical results reveal the complete compatibility of employed algorithm for solving image processing and numerical problems. Few examples are presented to show the efficiency and simplicity of the methods employed in this paper which is stable for solving linear and non-linear autonomous as well as non-autonomous problems.

Acknowledgements

The author would like to extend his sincere gratitude to Government of India, eternally providing financial support and necessary facilities to carry out doctoral research work via Technical Quality Improvement Programme (TEQIP), Under Ministry of Human Resource Development, to National Institute of Technology [REC], Tiruchirappalli-620 015, Tamilnadu, India. URL: http://www.nitt.edu; Further, the author express his grateful thanks to Vellore Institute of Technology-University, Vellore-632 014, Tamilnadu, India, URL: http://www.vit.ac.in for offering infrastructure services, constant support and encouragement to accomplish this research task.

References

[1] WH Press, BP Flannery, SA Teukolsky and WT Vetterling. *Numerical Recipes*. The Art of Scientific Computing, Cambridge University Press, New York, 1986.

[2] JC Butcher. *The Numerical Analysis of Ordinary Differential Equations*. John Wiley & Sons, UK, 2003.

[3] JC Butcher. *The Numerical Analysis of Ordinary Differential Equations*. Runge-Kutta and general linear methods. John Wiley & Sons, UK, 1987.

[4] MK Jain, SRK Iyengar and RK Jain. *Numerical Methods for Scientific and Engineering Computation*. 5th ed. New Age International Publishers, New Delhi, India, 2010.

[5] H Harrer, A Schuler and E Amelunxen. Comparison of different numerical integrations for simulating cellular neural networks. *In*: Proceeding of the IEEE International Workshop on Cellular Neural Networks and their Applications, Budapest, 1990, p. 151-9.

[6] CC Lee and JP de Gyvez. Time multiplexing CNN simulator. *In*: Proceeding of the IEEE International Symposium on Circuits and Systems, London, 1999, p. 407-10.

[7] C Lee, JP de Gyvez. Single-layer CNN simulator. *In*: Proceeding of the IEEE International Symposium on Circuits and Systems, London, 1994, p. 217-20.

[8] AM Wazwaz. A comparison of modified Runge-Kutta formula based on a variety of means. *Int. J. Comput. Math.* 1994; **50**, 105-12.
[9] LO Chua and L Yang. *Cellular Neural Networks: Theory*. *In*: Proceeding of the IEEE Transactions on Circuits and Systems, London, 1988, p. 1257-72.
[10] LO Chua and L Yang. *Cellular Neural Networks: Applications*. *In*: Proceeding of the IEEE Transactions on Circuits and Systems, London, 1988, p. 1273-90.
[11] LO Chua and T Roska. *The CNN Universal Machine Part 1: The Architecture*. *In*: Proceeding of the International Workshop on Cellular Neural Networks and their Applications, Munich, 1992, p. 1-10.
[12] T Roska. *CNN Software Library*. Hungarian Academy of Sciences, Analogical and Neural Computing Laboratory, Budapest, 2000.
[13] T Roska, L Kek, L Nemes, Á Zarándy and M Brendel. *CNN Software Library (Templates and Algorithms) Version 7.1*. Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, 1997.
[14] T Roska and L Kek. *Analogic CNN Program Library*. Version 6.1, Budapest, Hungary, 1994.
[15] T Roska. *CNNM Users Guide*. Version 5.3x, Budapest, Hungary, 1994.
[16] JA Nossek, G Seiler, T Roska and LO Chua. Cellular neural networks: theory and circuit design. *Int. J. Circ. Theor. Appl*. 1992; **20**, 533-53.
[17] S Wang, FL Korris and FD Chung. Applying the improved fuzzy cellular neural network IFCNN to white blood cell detection. *Neurocomputing* 2007; **70**, 1348-59.
[18] M Laiho, A Paasio, J Flak and KAI Halonen. Template design for cellular nonlinear networks with 1-bit weights. *IEEE Trans. Circ. Syst. Fund. Theor. Appl*. 2008; **55**, 904-13.
[19] T Yang, LB Yang, CW Wu and LO Chua. Fuzzy cellular neural networks: theory. *In*: Proceedings of the IEEE International Workshop on Cellular Neural Networks and Applications, Seville, 1996, p. 181-6.
[20] T Yang and LB Yang. The global stability of fuzzy cellular neural networks. *IEEE Trans. Circ. Syst. Fund. Theor. Appl*. 1996; **43**, 880-3.
[21] T Yang and LB Yang. Fuzzy cellular neural network: A new paradigm for image processing. *Int. J. Circ. Theor. Appl*. 1997; **25**, 469-81.
[22] T Yang and LB Yang. Application of fuzzy cellular neural networks to euclidean distance transformation. *IEEE Trans. Circ. Syst. Fund. Theor. Appl*. 1997; **44**, 242-246.
[23] S Wang, D Fu, M Xu and D Hu. Applying advanced fuzzy cellular neural network AFCNN to segmentation of serial CT liver images. *Lect. Notes Comput. Sci.* 2005; **3612**, 1128-31.
[24] S Wang, D Fu, M Xu and D Hu. Advanced fuzzy cellular neural network: Application to CT liver images. *Artif. Intell. Med*. 2007; **39**, 65-77.
[25] RC Gonzalez, RE Woods and SL Eddin. *Digital Image Processing using MATLAB*. Pearson Education Asia, Upper Saddle River, New Jersey, 2009.
[26] S Senthilkumar. 2009, New Embedded Runge-Kutta Fourth order Four Stage Algorithms for Raster and Time-Multiplexing Cellular Neural Networks Simulation. Ph.D. Thesis, Department of Mathematics, National Institute of Technology, Tiruchirappalli, Tamilnadu, India.
[27] S Senthilkumar. Raster simulation using advanced fuzzy cellular non-linear network. *Int. J. Autonom. Adapt. Comm. Syst.* 2010; **3**, 464-78.
[28] M Itoh and LO Chua. Designing CNN genes. *Int. J. Bifurcat. Chaos Appl. Sci. Eng*. 2003; **13**, 2739-824.
[29] ALADDIN CNN Software Library for ACE4K Chip (Templates and Algorithms). Budapest, Analogic Computers Ltd., 2000.