International Journal of Applied Mathematical Research, *4 (3) (2015) 437-441*
www.sciencepubco.com/index.php/IJAMR
©Science Publishing Corporation
doi: 10.14419/ijamr.v4i3.4591
Research Paper

An efficient numerical technique for the solution of nonlinear heat equation via spectral method

Mohammadreza Askaripour Lahiji [1]*, Mahdi Ghanbari [2], Hassan Panj Mini [3]

[1] *Department of Mathematics, Islamic Azad University, Astaneh Ashrafieh Branch, Astaneh Ashrafieh, Iran*
[2] *Department of Mathematics, Islamic Azad University, Khorramabad Branch, Khorramabad, Iran*
[3] *Technical and Vocational University Lahijan (Shaid Rajaei) Branch, Lahijan, Iran*
**Corresponding author E-mail: maskarepor@yahoo.com*

Abstract

Nonlinear wave equations are more difficult to study mathematically, and no general analytical method exists for their solution. It is found that the Exponential Time Differencing (ETD) scheme requires the steps to achieve a given accuracy, offers a speedy method in calculation time, and has exceptional stability properties in solving a nonlinear equation. This article solves the diagonal example of nonlinear heat equation via the exponential time difference Runge-Kutta 4 methods (ETDRK4). Implementation of the method is proposed by short Matlab programs.

Keywords: *Exponential Methods; Integration Factor Methods; Exponential Time Differencing Methods; Runge-Kutta Method.*

1. Introduction

It is found that several time-dependent partial differential equations (PDEs) combine low-order nonlinear terms with higher-order linear terms. It is most appropriate to apply high-order approximations in space and time in order to find accurate numerical solutions of such problems.
Cox and Matthews presented a clear derivation of the explicit ELP schemes of arbitrary order referring to the ELP methods as the Exponential Time Differencing (ETD) methods [1 - 3]. After that Tokman studied a class of exponential propagation techniques known as Exponential Propagation Iterative (EPI) schemes [4]. In order to make better the ETD schemes, Wright deliberated on these schemes and thus reforming the solution in integral form of a nonlinear autonomous system of ODEs to an extension in terms of matrix and vector functions products [5].
The basic formula of the ETD schemes, the linear part of the differential equation is integrated, and applied a suitable approximate for the nonlinear terms. Lawson [6] presented a similar approach for the first time that is currently used in the integrating factor (IF) schemes. In the approach of IF schemes [7 - 9], both sides of an ODE were multiplied by an integrating factor, and made a change of variable that the linear part can be solved exactly.
Applications of exponential time difference methods are in solving stiff systems. Moreover, comparing various fourth-order methods, including the ETD methods and their results revealed that the best choice was the ETD4RK method for solving various one-dimensional diffusion-type problems [10], [11]. An extensive application of the ETD methods has been made according to related work in many simulations of stiff problems [12]. Aziz et al. [13], [14] studied on the exponential time difference Runge-Kutta 4 method (ETDRK4) for solving the diagonal example of Korteweg-de Vries (KdV) and Kuramoto-Sivashinsky (K-S) equations [15], [16] with Fourier's transformation, and to be implement by the integration factor method. Other papers on this subject include [17 - 24].
The present paper is organized as follows: In section 1, we introduce the subject. In section 2, we carry out the execution on a diagonal example of nonlinear heat equation, together with fast Fourier transform (FFT).In section 3, we compare the exact and numerical solution. In section 4, a brief conclusion is given.

2. A diagonal example

Let us consider a nonlinear heat equation

$$u_t = u_{xx} + u^5 \,,\ x \in [0,2\pi] \tag{1}$$

With the initial condition given using

$$u(x,0) = 1 + \cos(\pi x). \tag{2}$$

In the above equation, we use the fast Fourier transform (FFT)

$$\hat{u}_t - i^2k^2\hat{u} + \widehat{u^5} = 0$$

Where $i = \sqrt{-1}$.The equation (1) is multiplied by e^{k^2t}, i.e.

$$e^{k^2t}\,\hat{u}_t + e^{k^2t}k^2\hat{u} + \ e^{k^2t}\,\widehat{u^2} = 0$$

If we define the change of variable

$$\widehat{U} = e^{k^2t}\hat{u} \tag{3}$$

With

$$\widehat{U}_t = k^2e^{k^2t}\hat{u} + e^{k^2t}\,\hat{u}_t \tag{4}$$

And substituting (3) in (4), we have

$$\widehat{U}_t + e^{k^2t}\,\widehat{u^5} = 0$$

Working in Fourier space (applying FFT), the numerical algorithm discretizing can be obtained by

$$\widehat{U}_t + e^{k^2t}\,F\big((F^{-1}\big(e^{-k^2t}\widehat{U}\big))^5\big) = 0$$

Where F is the Fourier transformed operator.
For time stepping, we use the ETDRK4 with $t = 150$, the ETDRK4 is given by

$$a_n = u_ne^{hL/2} + \big(e^{hL/2} - I\big)N(u_n,t_n)/L, \tag{5}$$

$$b_n = u_ne^{hL/2} + \big(e^{hL/2} - I\big)N(a_n,t_n + h/2\,)/L \tag{6}$$

$$c_n = a_ne^{hL/2} + \big(e^{hL/2} - I\big)\,(2N(b_n,t_n + h/2\,) - N(u_n,t_n))\,/L \tag{7}$$

$$u_{n+1} = a_ne^{hL} + \tag{8}$$

$$\left\{ \begin{array}{l} \emptyset_1N(u_n,t_n) + 2\emptyset_2(N(a_n,t_n + h/2\,) + N(b_n,t_n + h/2\,)) \\ \qquad\qquad +\emptyset_3\,N(c_n,t_n + h)\end{array}\right\} \Big/ (L^3h^2),$$

Where

$$\emptyset_1 = (L^2h^2 - 3Lh + 4)e^{hL} - Lh - 4\,, \tag{9}$$

$$\emptyset_2 = (Lh - 2)e^{hL} + Lh + 2\,, \tag{10}$$

$$\emptyset_3 = (-Lh + 4)e^{hL} - L^2h^2 - 3Lh - 4 \tag{11}$$

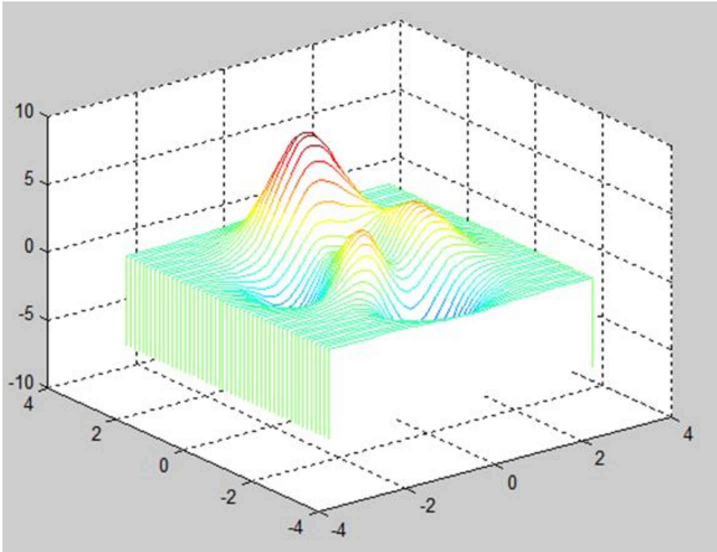

Fig. 1: Time Evolution for Nonlinear Equation. The **X** Axis Runs from $\mathbf{X = 0}$ to $\mathbf{X = 2\pi}$ and the **t**-Axis Runs from **t=0** to **t=150**.

3. Comparison with the numerical solution and the exact solution

The nonlinear heat equation (1) and initial condition (2) are considered .The results of the equation (2) is plotted in Figure (2)

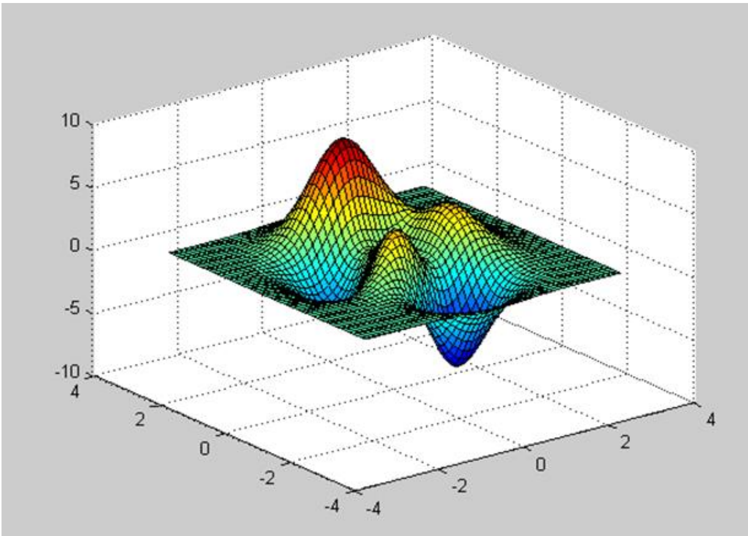

Fig. 2: The Evolution of Exact Solution Nonlinear Heat Equation

As a result, Figure 2 shows that the results of ETDRK4 numerical solution as shown in Figure 1 are acceptable to the exact solution. In conclusion, the ETDRK4 is able to generate the numerical solution.

4. Conclusion

In this work, we have presented a nonlinear wave equation with the initial condition $\mathrm{u(x,0)} = 1 + \cos(\pi \mathrm{x})$. For integrating the system (1), the ETDRK4 method is used. For solving the equation, we implemented the Matlab software. Figure 1 shows the result created by Matlab code listed (Appendix A). It is observed that ETDRK4 is suitable in every case. For diagonal and non-diagonal problems, it works equally well due to it is accurate and fast, and the ETDRK4 needs the fewest steps to achieve a given accuracy. Moreover, for running the programmes, the computational time required is less than one second, which is faster compared with the conventional Runge-Kutta 4.

References

[1] S. Cox, P. Matthews, *Exponential time differencing for stiff systems*, Journal of Computational Physics. 176(2) (2002) 430-455. http://dx.doi.org/10.1006/jcph.2002.6995.

[2] R. Holland, *Finite-difference time-domain (FDTD) analysis of magnetic diffusion*, Electromagnetic Compatibility, IEEE Transactions on. 36(1) (1994) 32-39.

[3] P.G. Petropoulos, *Analysis of exponential time-differencing for FDTD in lossy dielectrics.* Antennas and Propagation, IEEE Transactions on. 45(6) (1997) 1054-1057. http://dx.doi.org/10.1109/8.585755.

[4] M. Tokman, *Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods.* Journal of Computational Physics. 213(2) (2006) 748-776. http://dx.doi.org/10.1016/j.jcp.2005.08.032.

[5] W. Wright, *A partial history of exponential integrators.* Department of Mathematical Sciences, NTNU, Norway, 2004.

[6] J.D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants.* SIAM Journal on Numerical Analysis. 4(3) (1967) 372-380. http://dx.doi.org/10.1137/0704033.

[7] H. Berland, B. Owren, B. Skaflestad, *Solving the nonlinear Schrodinger equation using exponential integrators,* Modeling, Identification and Control. 27(4) (2006) 201-217. http://dx.doi.org/10.4173/mic.2006.4.1.

[8] A.K. Kassam, *High order time stepping for stiff semi linear partial differential equations*, University of Oxford, 2004.

[9] H. Berland, B. Skaflestad, W.M. Wright, *EXPINT-A MATLAB package for exponential integrators.* ACM Transactions on Mathematical Software (TOMS). 33(1) (2007) 4. http://dx.doi.org/10.1145/1206040.1206044.

[10] A.K. Kassam, L.N. Trefethen, *Fourth-order time-stepping for stiff PDEs.* SIAM Journal on Scientific Computing. 26(4) (2005) 1214-1233. http://dx.doi.org/10.1137/S1064827502410633.

[11] S. Krogstad *Generalized integrating factor methods for stiff PDEs.* Journal of Computational Physics. 203(1) (2005) 72-88. http://dx.doi.org/10.1016/j.jcp.2004.08.006.

[12] C. Klein, *Fourth order time-stepping for low dispersion Korteweg-de Vries and nonlinear Schrödinger equation.* Electronic Transactions on Numerical Analysis. 29 (2008) 116-135.

[13] Z.A. Aziz, N. Yaacob, M. Askaripour, M. Ghanbari, *Fourth-Order Time Stepping for Stiff PDEs via Integrating Factor.* Advanced Science Letters. 19(1) (2013) 170-173. http://dx.doi.org/10.1166/asl.2013.4667.

[14] Z.A. Aziz, M. Askaripour, M. Ghanbari, *a New Review of Exponential Integrator*, USA: CreateSpace 106, 2012.

[15] J.M. Hyman, B. Nicolaenko, *The Kuramoto-Sivashinsky equation: a bridge between PDE's and dynamical systems.* Physica D: Nonlinear Phenomena. 18(1) (1986) 113-126.

[16] B. Nicolaenko, B. Scheurer, R. Temam, *Some global dynamical properties of the Kuramoto-Sivashinsky equations: nonlinear stability and attractors.* Physica D: Nonlinear Phenomena. 16(2) (1985) 155-183. http://dx.doi.org/10.1016/0167-2789(85)90056-9.

[17] M. Askaripour, Z.A. Aziz, M. Ghanbari, and H. Panj mini, *A Note on Fourth-Order Time Stepping for Stiff PDE via Spectral Method.* Applied Mathematical Sciences.7 (38) (2013) 1881-1889.

[18] Z.A. Aziz, N. Yaacob, M. Askaripour, M. Ghanbari, *A review for the time integration of semi-linear stiff problems.* Journal of Basic & Applied Scientific Research. 2(7) (2012) 6441-6448.

[19] M. A. Lahiji, Z. A. Aziz, Numerical Solution for Kawahara Equation by Using Spectral Methods, *IERI Procedia*. *10* (2014) 259-265. http://dx.doi.org/10.1016/j.ieri.2014.09.086.

[20] M. A. Lahiji, Z. A. Aziz,Numerical Solution of the Nonlinear Wave Equation via Fourth-Order Time Stepping, In *Applied Mechanics and Materials*.729 (2015) 213-219. http://dx.doi.org/10.4028/www.scientific.net/AMM.729.213.

[21] Z.A. Aziz, N. Yaacob, M. Askaripour, M. Ghanbari, *A review of the time discretization of semi linear parabolic problems.* Research Journal of Applied Sciences, Engineering and Technology. 4(19) (2012) 3539-3543.

[22] Z.A. Aziz, N. Yaacob, M. Askaripour, M. Ghanbari, *Split-Step Multi-Symplectic Method for Nonlinear Schrödinger Equation,* Research Journal of Applied Sciences, Engineering and Technology. 4(19) (2012) 3858-3864.

[23] Z.A. Aziz, N. Yaacob, M. Askaripour, M. Ghanbari, *A Numerical Approach for Solving a General Nonlinear Wave Equation,* Research Journal of Applied Sciences, Engineering and Technology. 4(19) (2012) 3834-3837.

[24] M. Askaripour, Z.A. Aziz, M. Ghanbari, A. Farzamnia, Efficient Semi-Implicit Schemes for Stiff Systems via Newton's Form, *Journal of Optoelectronics and Biomedical Materials.5* (3) (2013) 43-50.

Appendix A

Matlab code to solve nonlinear heat equation and produce Figure 1

```
clear
clc
% Spatial grid and initial condition:
N = 128;
dt = .4/N^2;
x = 2*pi*(1:N)/N;
u=1+cos(pi*x);
v = fft(u);
% Precompute various ETDRK4 scalar quantities:
h = 1/6; % time step
k = [0:N/2-1 0 -N/2+1:-1]; % wave numbers
L = k.^2 ; % Fourier multipliers
E = exp(-dt*L/2); E2 = E.^2;
M = 16; % no. of points for complex means
r = exp(1i*pi*((1:M)-.5)/M);
LR1= h*L(:,ones(M,1));
LR2= r(:,ones (M,1));
LR= LR1+LR2;
Q = h*real(mean( (exp(LR/2)-1)./LR ,2));
f1 = h*real(mean( (-4-LR+exp(LR).*(4-3*LR+LR.^2))./LR.^3 ,2));
f2 = h*real(mean( (2+LR+exp(LR).*(-2+LR))./LR.^3 ,2));
f3 = h*real(mean( (-4-3*LR-LR.^2+exp(LR).*(4-LR))./LR.^3 ,2));
uu = u; tt = 0;
tmax = 150; nmax = round(tmax/h);
nplt = floor((tmax/100)/h);
g = exp(dt*L);
Nv = g.*fft(real(ifft(v)).^5);
a = E2.*v + Q.*Nv;
Na = g.*fft(real(ifft(a)).^5);
b = E2.*v + Q.*Na;
Nb = g.*fft(real(ifft(b)).^5);
c = E2.*a + Q.*(2*Nb-Nv);
Nc = g.*fft(real(ifft(c)).^5);
v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
for n = 1:nmax
t = n*h;
if mod(n,nplt)==0
u = real(ifft(v));
uu = [uu,u]; tt = [tt,t];
end
end
nn=length(tt);
mm=length(x);
uu2=reshape(uu,mm,nn);
figure
[mm,nn,uu2]=peaks;
waterfall (mm,nn,uu2);
```