A Note on Fourth-Order Time Stepping for

Stiff PDE via Spectral Method

Mohammadreza Askaripour Lahiji\textsuperscript{1,2}, Zainal Abdul Aziz\textsuperscript{1,2,*},

Mahdi Ghanbari\textsuperscript{2} and Hassan Panj Mini\textsuperscript{3}

\textsuperscript{1}UTM Centre for Industrial and Applied Mathematics and
\textsuperscript{2}Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi
Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia
\textsuperscript{3}Department of Mathematics, Islamic Azad University,
Astaneh Ashrafieh Branch, Iran

\*Corresponding author: Zainal Abdul Aziz, zainalabdaziz@gmail.com

Copyright © 2013 Mohammadreza Askaripour Lahiji et al. This is an open access article
distributed under the Creative Commons Attribution License, which permits unrestricted use,
distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this note it is illustrated that the Exponential Time Differencing (ETD) scheme
needs the least steps to achieve a given accuracy, offers a speedy method in
calculation time, and has exceptional stability properties in solving a stiff type
problem. Nonetheless, the celebrated and well established method like Runge-
Kutta is still being applied as the basis of many efficient codes. However, the stiff
type problems seem cannot be solved efficiently via some of these methods. This
note overcomes such stiff type problem via the exponential method. Furthermore,
the exponential time differencing Runge-Kutta 4 method (ETDRK4) is used to
solve the diagonal example of a well known nonlinear partial differential equation
(PDE) in the form of Burgers’ equation. In addition, we use Fourier transformation
for solving Burgers’ equation.

Mathematics Subject Classification: 65M70, 65Z05

Keywords: Exponential methods; Integration factor methods; Exponential time
differencing methods; Runge-Kutta method, Burgers’ equation
1. Introduction

It is found that several time-dependent partial differential equations (PDEs) combine low-order nonlinear terms with higher-order linear terms. Examples are as in the following equations of Allen-Cahn, Burgers, Cahn-Hilliard, Fisher-KPP, Fitzhugh-Nagumo, Gray-Scott, Hodgkin-Huxley, Kuramoto-Sivashinsky, Navier-Stokes and nonlinear Schrödinger. It is most appropriate to apply high-order approximations in space and time for finding accurate numerical solutions of such problems. The majority of calculations have been constrained to second order in time due to the difficulties established by the combination of stiffness and nonlinearity.

Cox and Matthews [1] presented a clear derivation of the explicit Exact Linear Part (ELP) schemes of arbitrary order referring to the above-mentioned methods as the Exponential Time Differencing (ETD) methods (e.g., Holland, [2], Petropoulos, [3]). After that Tokman [4] studied on these formulas leading to a class of exponential propagation techniques known as Exponential Propagation Iterative (EPI) schemes. Reforming integral form of a solution to a nonlinear autonomous system of ODEs as an expansion in terms of products of matrix and vector functions, Wright [5] considered these schemes in order to improve the ETD schemes.

The basics of the formula of ETD schemes are in integrating the linear parts of the differential equation precisely, and approximating the nonlinear terms by a polynomial, which is then integrated exactly. Lawson [6] presented a similar approach for the first time which is currently being used in the Integrating Factor (IF) schemes. In the approach of IF schemes (e.g., Berland et al., [7], Kassam, [8], Berland et al., [9]), both sides of an ODE are multiplied by an appropriate integrating factor, and a differential equation is obtained in which change variables are changed so that the linear part could be solved exactly.

Applications of ETD methods in solving stiff systems are extensive. Moreover, (e.g., Kassam and Trefethen, [10], Krogstad, [11]) in 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. Extensive application of the ETD methods has been made according to related work in many simulations of stiff problems (e.g., Klein, [12]). Aziz et al. [13], [14] studied the exponential time differencing Runge-Kutta 4 method (ETDRK4) for solving the diagonal example of Korteweg-de Vries (KdV) and Kuramoto-Sivashinsky (K-S) equations (e.g., Hyman & Niclanenko [15], Niclanenko et al. [16]) with Fourier transformation, and to
implement by the integration factor method. The paper is organized as follows: In section 1, we introduced the subject. In section 2, we carried out the execution on a diagonal example in Burgers’ equation, and together with fast Fourier Transform (FFT). In section 3, some results and discussion are furnished and finally in section 4, a brief conclusion is given.

2. A diagonal example: Burgers’ Equation

Let us consider Burgers’ equation, which is a fundamental nonlinear partial differential equation from fluid mechanics. It occurs in various areas of applied mathematics, such as modeling of gas dynamics and traffic flow. It is named after Johannes Martinus Burgers (1895–1981).

For a given velocity, \( u \) and viscosity coefficient \( j \), the general form of Burgers’ equation (also known as viscous Burgers' equation, whiles for \( j = 0 \) we have the inviscid Burgers’ equation) is given by

\[
    u_t - ju_{xx} + uu_x = 0 \quad x \in [0,1], \ t \in [0,1]
\]  

(1)

with the initial and Dirichlet boundary conditions prescribed using

\[
    u(x, 0) = (\sin(2\pi x))^2 (1 - x)^{\frac{3}{2}}
\]  

(2)

where \( N = 500 \), \( j = 0.0003 \) (for viscous Burgers’ equation) and \( j = 0 \) (for inviscid Burgers’ equation), \( r = 0.03 \) (the roots of unity in Matlab codes).

As a result of the periodic boundary condition, the problem can be reduced to a diagonal form by Fourier transformation.

In solving the problem, we can write

\[
    u_t - ju_{xx} + \left( \frac{1}{2} u^2 \right)_x = 0.
\]  

(3)

In the above equation, we apply the Fast Fourier transform (FFT)

\[
    \hat{u}_t + jk^2 \hat{u} + \frac{1}{2} ik \hat{u}^2 = 0
\]  

(4)

where \( i = \sqrt{-1} \).

The equation (4) is multiplied by \( e^{ik^2 t} \), i.e.

\[
    e^{ik^2 t} \hat{u}_t + e^{ik^2 t} e^{jk^2 t} \hat{u} + \frac{1}{2} ik e^{ik^2 t} \hat{u}^2 = 0.
\]  

(5)

If we define the change of variable

\[
    \hat{U} = e^{ik^2 t} \hat{u}
\]  

(6)

with

\[
    \hat{U}_t = jk^2 e^{ik^2 t} \hat{u} + e^{ik^2 t} \hat{u}_t,
\]  

(7)

and substituting (7) in (5), we have

\[
    \hat{U}_t + \frac{1}{2} ik e^{ik^2 t} \hat{u}^2 = 0.
\]  

(8)
Working in Fourier space (applying FFT), the numerical algorithm discretizing can be obtained by
\[
\hat{u}_t + \frac{i}{2} e^{jk^2t} kF\left((F^{-1}(e^{-jk^2t}\hat{u}))^2\right) = 0.
\] (9)

For time stepping, we use the ETDRK4 with \( t = 150 \), the ETDRK4 is given as follows;

\[
a_n = u_n e^{hL/2} + (e^{hL/2} - 1)N(u_{n}, t_{n})/L,
\] (10)

\[
b_n = u_n e^{hL/2} + (e^{hL/2} - 1)N(a_{n}, t_{n} + h/2)/L,
\] (11)

\[
c_n = a_n e^{hL/2} + (e^{hL/2} - 1)\left(2N(b_n, t_n + h/2) - N(u_n, t_n)\right)/L,
\] (12)

\[
u_{n+1} = a_n e^{hL} + \\
\{ \phi_1 N(u_n, t_n) + 2\phi_2 N(a_n, t_n + h/2) + N(b_n, t_n + h/2) \} / (L^3 h^2),
\] (13)

where

\[
\phi_1 = (L^2 h^2 - 3Lh + 4)e^{hL} - Lh - 4,
\] (14)

\[
\phi_2 = (Lh - 2)e^{hL} + Lh + 2,
\] (15)

\[
\phi_3 = (-Lh + 4)e^{hL} - L^2 h^2 - 3Lh - 4.
\] (16)

The numerical programme (Matlab codes) is implemented as follows
(for \( j = 0.0003 \) and \( j = 0 \)):

```matlab
clear
close all
cle
N = 500;
dt = .4/N^2;
x=linspace(0,1,N);
u=(sin(2*pi*x)).^2.*((1-x).^1.5);
v= fft(u);

% precomputed various ETDRK4 scalar quantities:
k = [0:N/2-1 0 -N/2+1:-1];
j=0.0003
jk2 = (j^*k).^2;
L=jk2 ;
h = input('inter step h=');
E = exp(dt*jk2/2); E2 = E.^2;
r=0.03;
LR1= h*L;
LR2= r ; LR= LR1+LR2;
Q = h*real(mean((exp(LR/2)-1)./LR ,2));
```
Time stepping for stiff PDE via spectral method

\[
f_1 = h * \text{real(mean}(( -4-LR+\exp(LR).*(-4-3*LR^2))/.LR^3 ,2)) ;
\]
\[
f_2 = h * \text{real(mean}(( 2+LR+\exp(LR).*(-2+LR))/.LR^3 ,2)) ;
\]
\[
f_3 = h * \text{real(mean}(( -4-3*LR^2+\exp(LR).*(-4-LR))/.LR^3 ,2)) ;
\]

% Main time-stepping loop:

uu = u; tt = 0;
g = -.5i*dt*k ;

% Solve PDE and plot results:
tmax = 0.006 ; nplt = floor ( (tmax/25) /dt ) ;
nmax = round(tmax/dt) ;
for n = 1:nmax
    t = n*dt;
    Nv = g.*fft(real(ifft(v)).^2);
    a = E2.*v + Q.*Nv;
    Na = g.*fft(real(ifft(a)).^2);
    b = E2.*v + Q.*Na;
    Nb = g.*fft(real(ifft(b)).^2);
    c = E2.*a + Q.*(2*Nb-Nv);
    Nc = g.*fft(real(ifft(c)).^2);
    v = E.*v + Nv.*f1 + 2*(Na+Nb).*f2 + Nc.*f3;
    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);
xlabel nn, ylable mm

3. Results and Discussion

The computational time required for running the above programme is less than one second, which is fast as compared to the conventional Runge-Kutta 4. Even though there exists certain unusual sensitivity of this Burgers’ equation to
perturbations (refer to [8], [9]), the above implementation of the codes computes accurately in less than one second. This is possible since the ETDRK4 is $A$-stable and thus has exceptional stability properties in solving this stiff type problem. Computational results are depicted in figures 1 and 2, which show the solution graphs of the inviscid and viscous Burgers’ equation respectively.

**Fig.1.** Time evolution for the inviscid Burgers equation ($j = 0$). The $x$ axis runs from $x = -3$ to $x = 3$, and the $t$-axis runs from $t = 0$ to $t = 150$. 
4. Conclusion

This note overcomes a stiff type problem via the exponential method. We have utilized effectively the exponential time differencing Runge-Kutta 4 method (ETDRK4) to solve the diagonal example of Burgers’ equation (inviscid and viscous forms) with Fourier's transformation. By implementing the Matlab codes, we have successfully solved numerically the Burgers equation. In future publication, we hope to employ these techniques to more complicated non-diagonal case, for example the Fisher equation, which is a well known equation from the research areas in heat & mass transfer, population dynamics and ecology.

Acknowledgement
Reza is thankful to UTM for International Doctoral Fund (IDF). This research is partially funded by MOHE FRGS Vote no. 78675 and UTM RUG Vot. No.05J13.
References


Received: January 4, 2013