International Review of Fuzzy Mathematics, Volume 13, No. 2, (December 2018)

Harpreet Kaur and B. P. Garg

MATLAB PROGRAMMING FOR 1^{st}, 2^{nd} & 3^{rd} ORDER ORDINARY DIFFERENTIAL EQUATION OF R-K 4^{th} ORDER METHOD

Abstract: *In this paper, the MATLAB programming is used to provide a solutions to the well-known ordinary differential equations of different order such as Basset-Boussinesq-Ossen equation (1st order) , Rayleigh's equation (2nd order) & Blasius equation (3rd order) by R-K 4th order method. The results obtained by this approach are illustrated and show that Matlab programming is powerful forth is type of equations.*

Keywords: *Basset-Boussinesq-Ossen equation, Blasius Equation, Ordinary Differential Equation, Rayleigh Differential Equation, R-K 4^{th} Order Method.*

1. INTRODUCTION

Numerical solution have always played and still play an important role in properly understanding the qualitative features of many phenomena and processes in various field of science. It is clear that the type of current problems is to find the solution of ordinary differential equation of 1^{st}, 2^{nd} and 3^{rd} order. So for this, we can apply numerical methods like trapezoidal method, Euler's method (1^{st} order Runge-Kutta method), and mid-point method. Trapezoidal method is generally used for typical problems. Mid-point method is the modification of Euler's method. Thus the mid-point method is as a suitable numerical technique in present problem which is also called R-K 4^{th} order method (numerical method) [1]. In numerical analysis, the Runge-Kutta methods are a family of implicit and explicit iterative methods. These methods were developed around 1900 by German mathematicians C. Runge and M. W. Kutta. [2].

MATLAB is the fourth-generation programing language and numerical analysis environment. MATLAB is used in many fields such as computational, image, signal processing, communication, control system for industry, smart grid design etc. It is the powerful technique to solve the problems in mathematics[3].

2. PROBLEMS STSTEMENT

2.1. 1st Order Differential Equation (Basset-Boussinesq-Ossen equation)

In Fluid Dynamics, the Basset-Boussinesq-Ossen equation describes the motion of a small particle in unsteady flow at low Reynolds numbers. The equation is named after Alfred Barnard Basset, Joseph Valentin Boussinesq and Carl Wilhelm Ossen. This equation is also known as force balance equation or BBO equation [4]. Consider a rigid body, non-spherical particle with Sphericity φ, equivalent volume diameter D, mass m and particle density ρ_s is falling in an infinite extent of incompressible Newtonian fluid of density ρ and viscosity μ, u represents the velocity of the non-spherical particle at any instant time t, and g is the acceleration due to gravity [5]. Thus, the force balance equation (BBO) of particle motion is given by

$$mu'(t) = mg\left(1 - \frac{\rho}{\rho_s}\right) - \frac{1}{8}\pi D^2 \rho C_D u^2(t) - \frac{1}{12}\pi D3\rho u'(t) \tag{1}$$

where $u'(t)$ 1st order derivative.

Where $C_D = \dfrac{30}{\text{Re}} + 67.289 e^{(-5.03\varphi)}$ is the drag coefficient and $\text{Re} = \dfrac{\rho D u}{\mu}$ is Reynolds numbers Eq. (1) could be re-written as follows:

$$\left(m + \frac{1}{12}\pi D^3 \rho\right) u'(t) = mg\left(1 - \frac{\rho}{\rho_s}\right) - \frac{1}{8}\pi D^2 \rho C_D u^2(t) \tag{1a}$$

$$au'(t) + bu(t) + cu^2(t) - d = 0 \tag{1b}$$

Where $a = \left(m + \dfrac{1}{12}\pi D^3 \rho\right)$, $b = 3.75\pi D\mu$, $c = \dfrac{67.289 e^{(-5.03\varphi)}}{8}\pi D^2 \rho, d = mg\left(1 - \dfrac{\rho}{\rho_s}\right)$

i.e a, b, c, d are constant.

Take $a = b = c = d = 1$

So that $u'(t) = 1 - u^2(t) - u(t)$, is a 1st order differential equation with initial condition $u(0) = 0$

f is a function of time and velocity

$$\text{i.e. } f(t, u) = 1 - u^2 - u,\ u(0) = 0 \tag{1c}$$

Matlab programming for R-K 4th order method of equ. (1c) (1st order differential equation)

```
f = @(t, u)(1–u^2–u);
t = 0;
u = 0;
h = 0.25;
t = 0: h: 3;
for i = 1:(length(t)-1)
    k1 = f(t(i), u(i));
    k2 = f(t(i) + 0.5* h,u(i) + 0.5*h*k1);
    k3 = f(t(i) + 0.5* h,u(i) + 0.5*h*k2);
    k4 = f(t(i) + h,u(i) + h*k3);
    u(i + 1) = u(i) + 1/6*(k1+2*k2+2*k3+k4)*h;
end
u(:)
plot(t,u,'-+');
```

2.2. 2nd Order Differential Equation (Rayleigh's Differential Equation)

In Fluid Dynamic, Rayleigh's equation is a linear ordinary differential equation to study the hydrodynamic stability of a parallel, incompressible and inviscid shear flow. In 1880, the equation was introduced by the British mathematician Lord Rayleigh [6].

The equation Rayleigh's is given by

$$u''(t) - \mu\left(1 - \frac{1}{3}\left(u'(t)\right)^2\right)u'(t) + u(t) = 0,\ \mu > 0 \tag{2}$$

Take $\mu = 1$

Eq. (2) could be re-written as follows:

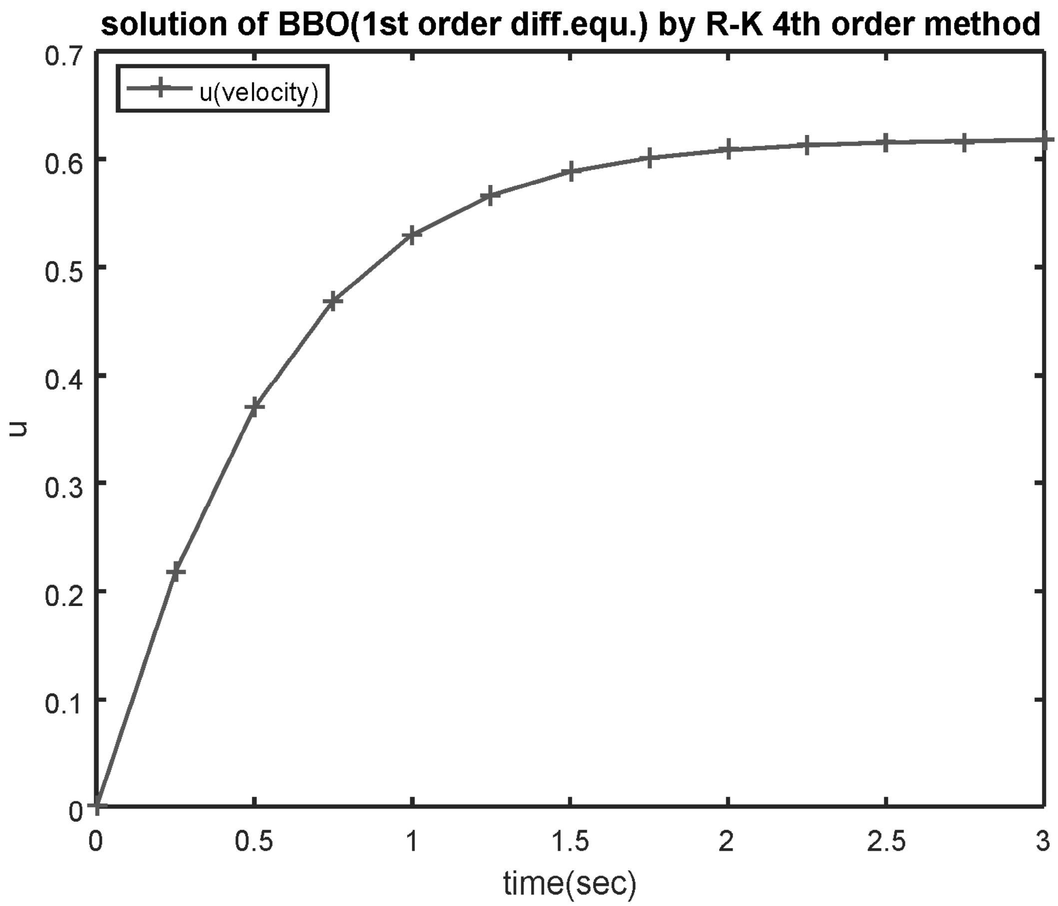

Figure 1: Solution of Basset-Boussinesq-Ossen equation by R-K 4th order method

$$u''(t) - \left(1 - \frac{1}{3}\left(u'(t)\right)^2\right)u'(t) + u(t) = 0$$ With initial condition $u'(0) = 0$, $u(0) = 1$

(2a)

Where $u''(t)$ 2nd order derivative and $u'(t)$ 1st order derivative.

Let $u = F$, $u' = F' = G, u'' = F'' = G'$

$$u''(t) = \left(1 - \frac{1}{3}\left(u'(t)\right)^2\right)u'(t) - u(t) \;\; i.e \;\; G' = \left(1 - \frac{1}{3}G^2\right)G - F \tag{2b}$$

Matlab programming for R-K 4th order method of equ. (2a) (2nd order differential equ.)

```
F1=@(t,F,G)  (G);
F2=@(t,F,G)  ((1-1/3*G^2)*G-F);
t = 0;
F = 1;
G = 0;
h = 0.25;
t = 0:h:3 ;
for i=1:(length(t)-1);
    t(i+1)=t(i)+h;
    k1=F1(t(i), F(i),  G(i));
    l1=F2(t(i),  F(i),  G(i));
    k2=F1(t(i)+0.5*h,F(i)+0.5*h*k1, G(i)+0.5*h*l1);
    l2=F2(t(i)+0.5*h,F(i)+0.5*h*k1, G(i)+0.5*h*l1);
    k3=F1(t(i)+0.5*h,F(i)+0.5*h*k2, G(i)+0.5*h*l2);
    l3=F2(t(i)+0.5*h,F(i)+0.5*h*k2, G(i)+0.5*h*l2);
    k4=F1(t(i)+h,F(i)+h*k3 , G(i)+h*l3 );
    l4=F2(t(i)+h,F(i)+h*k3 , G(i)+h*l3 );
    F(i+1)=F(i)+((k1+2*k2+2*k3+k4)*h)/6;
    G(i+1)=G(i)+((l1+2*l2+2*l3+l4)*h)/6;
end
F(:)
G(:)
Plot(t,F,'--',t,G,'-+');
```

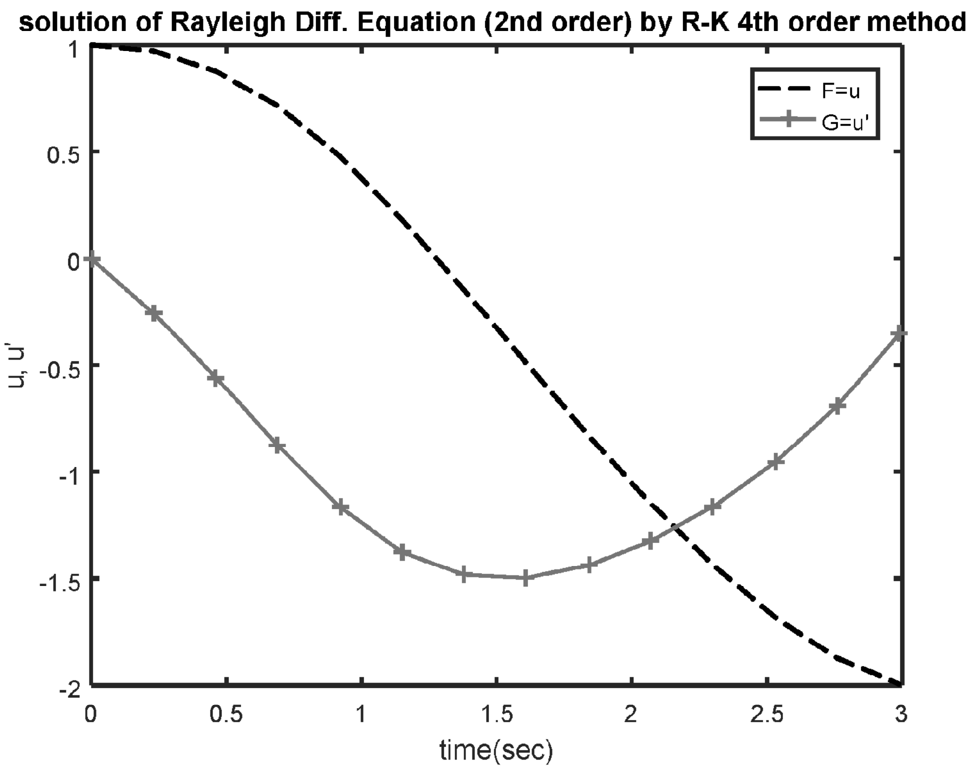

Figure 2: Solution of Rayleigh differential equation by R-K 4th order method

2.3. 3rd Order Differential Equation (Blasius's Equation)

In Fluid Dynamics, a Blasius equation is 3rd order ordinary differential equation describes by Paul Richard Heinrich Blasius to the steady two- dimensional laminar flow that forms on a semi-infinite plate which is held parallel to a constant unidirectional flow. Falkner and Skan later generalized Blasius's solution to wedge flow (Falkner-Skan boundary layer), i.e. flows in which the plate is not parallel to the flow [7].

We consider the generalized Blasius equation is

$$u'''(t) + \alpha u(t)\, u''(t) = 0, \qquad 0 \le t < \infty, \tag{3}$$

Where $\alpha = 1$ or $\alpha = 1/2$, with boundary conditions

$$u(0) = 0,\ \ u'(0) = \delta,\ u'(\infty) = 1 \tag{3a}$$

For special case of $\alpha = 1/2$ and $\delta = 0$, the Blasius equation is

$$u'''(t) + \frac{1}{2}u(t)u''(t) = 0 \text{ with } u(0) = u'(0) = 0,\ u'(\infty) = 1 \quad (3b)$$

where $u(t)$ is the free stream velocity function, t stands for thickness of boundary layer and $u'''(t)$ denotes the 3rd order derivative and $u''(t)$ 2nd order derivative.

Take $u''(0)$=0.332[8]

Let $u = F$, $u' = F' = G$, $u'' = F'' = G' = H$, $u''' = H'$

So $$u'''(t) = -\frac{1}{2}u(t)u''(t)\; i.e. H' = -\frac{1}{2}FH \quad (3c)$$

Matlab programming for R-K 4th order method of equ. (3b) (3rdorder differential equ.)

```
F1=@(t,F,G,H)        (G);
F2=@(t,F,G,H)        (H);
F3=@(t,F,G,H)(-1/2*F*H);
t=0;
F=0;
G=0;
H=0.332;
h=0.25;
t= 0:h:3;
for i=1:(length(t)-1);
    t(i+1)=t(i)+h;
    k1=F1(t(i) ,F(i) , G(i) , H(i) );
    l1=F2(t(i) ,F(i) , G(i) , H(i) );
    m1=F3(t(i) ,F(i) , G(i) , H(i) );
    k2=F1(t(i)+0.5*h,F(i)+0.5*h*k1, G(i)+0.5*h*l1, H(i)+0.5*h*m1);
    l2=F2(t(i)+0.5*h,F(i)+0.5*h*k1, G(i)+0.5*h*l1, H(i)+0.5*h*m1);
    m2=F3(t(i)+0.5*h,F(i)+0.5*h*k1, G(i)+0.5*h*l1, H(i)+0.5*h*m1);
    k3=F1(t(i)+0.5*h,F(i)+0.5*h*k2, G(i)+0.5*h*l2, H(i)+0.5*h*m2);
```

```
    l3=F2(t(i)+0.5*h,F(i)+0.5*h*k2, G(i)+0.5*h*l2, H(i)+0.5*h*m2);
    m3=F3(t(i)+0.5*h,F(i)+0.5*h*k2, G(i)+0.5*h*l2, H(i)+0.5*h*m2);
    k4=F1(t(i)+h ,F(i)+h*k3 , G(i)+h*l3 , H(i)+h*m3 );
    l4=F2(t(i)+h ,F(i)+h*k3 , G(i)+h*l3 , H(i)+h*m3 );
    m4=F3(t(i)+h ,F(i)+h*k3 , G(i)+h*l3 , H(i)+h*m3 );
    F(i+1)=F(i)+((k1+2*k2+2*k3+k4)*h)/6;
    G(i+1)=G(i)+((l1+2*l2+2*l3+l4)*h)/6;
    H(i+1)=H(i)+((m1+2*m2+2*m3+m4)*h)/6;
end
F(:)
G(:)
H(:)
plot(t,F,'--',t,G,'-+',t,H,'-o');
```

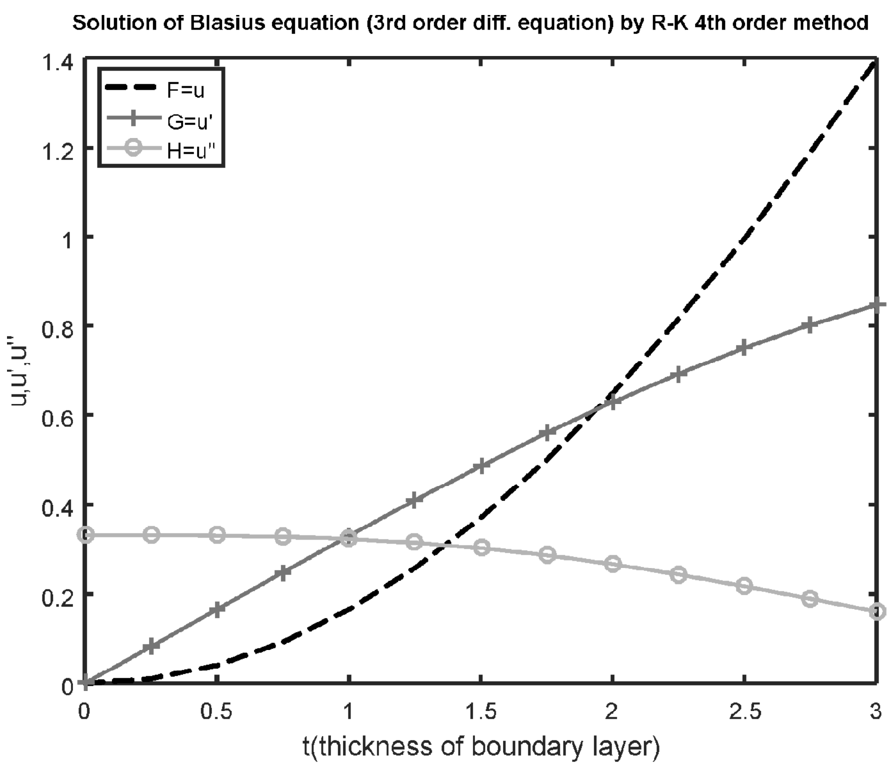

Figure 3: Solution of Blasius equation by R-K 4th order method

3. CONCLUSION

In this article, a new MATLAB programming is developed for solving linear and nonlinear ordinary differential equation of 1st, 2nd and 3rd order with initial and boundary conditions. In figs 1,2 and 3, u (vertically) denotes the velocity w.r.t. time t (horizontally) shows the solution of ordinary differential equations of different orders with help of MATLAB programing. From above discussion, it is clear that the MATLAB is suitable technique for solving the mathematical problems. Also the future scope of this research is that, the programming is generalized for higher order (more than 3rd order) ordinary differential equation and helpful in other differential equation of same orders.

ACKNOWLEDGMENT

The authors would like to thank to the Department of Mathematics at IK Gujral Punjab Technical University, Kapurthala (Punjab) where the author worked as a visiting scholar.

REFERENCES

[1] H. Kaur, B.P.Garg, Acceleration Motion of a Single Vertically Falling Non-Spherical Particle in Incompressible Newtonian Fluid by Different Methods, *IJMTT* 52(7) (2017), pp. 452-457.

[2] *https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods*

[3] *https://whatis.techtarget.com/definition/MATLAB*

[4] *https://en.wikipedia.org/wiki/Basset%E2%80%93Boussinesq%E2%80%93Oseen_equation.*

[5] Hessmeddin Yaghoobi, Mohesen Torabi, Novel solution for acceleration motion of vertically falling non-spherical particle by VIM-Pade' approximan, Powder Technology, (2012), pp 215-216.

[6] *https://en.wikipedia.org/wiki/Rayleigh%27s_equation_(fluid_dynamics)*

[7] *https://en.wikipedia.org/wiki/Blasius_boundary_layer*

[8] Yucheng Liu, Sree N. Kurra, Solution of Blasius Equation by Varinational Iteration Applied Mathematics 1(1), (2011), pp 24-27.

Harpreet Kaur

Research scholar of I.K. Gujral Punjab Technical University, Kapurthala, Punjab (INDIA)

B. P. Garg

Research supervisor of I.K. Gujral Punjab Technical University, Kapurthala, Punjab (INDIA)

E-mail: harpreet2610@yahoo.com