SOLUTION OF THE SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS USING RUNGA - KUTTA METHOD BY A NEW MATHEMATICAL TECHNIQUE

Abdel Radi Abdel Rahman Abdel Gadir Abdel Rahman[1*], Tasabeh Awadallah Abdel Majid Ali[2], Anoud Hassan Elzain Ageeb[3]

[1,2]Department of Mathematics, Faculty of Eduction ,Omdurman IsIamic University ,Omdurman, Sudan*

[3]Department of Mathematics, Gadarif Technologic College, Sudan Technical University, Gadarif, Sudan

Corresponding Author:-

Abstract

This study aims to recognizing the role played by differential and normal equations loop and their methods of solution if they are linear or non-linear, their class, and if they are of constant or variable coefficient .The study also aims to study the concept and development implementation of differential equations and their increasing importance in all scientific fields and their some applications. We followed in this study the mathematical deductive and inductive by using Runga - Kutta method by a new mathematical technique . We found the some following results:.. Runga - Kutta's method in general situation depends upon calculating inclination at the point x_0 and at many other points nearing x_0 then taking the average of these inclinations and multiply it by h then adding the resulting value toY_0 in order to get the result Y_1.$Y_1 = Y_0 + K_{av}$, Calculating relative error in Runga - Kutta's method from second and fourth order by knowing the analytical solution . The high efficiency of using Runga - Kutta's methods in solving the initial value problem comes through the numerical results obtained from the application of the methods and various examples.The new mathematical technique which we used in this study its an easy and accurate method that reduces errors and provide graphic solutions , so we recommended researcher to use it.

Keywords: *Calculation, Second order Ordinary Differential Equations, Runga - Kutta's method.*

1.INTRODUCTION:

On physics, we always are interested in how things move, it could be: in time, in position, or any other variable, the important part is that we always want to know how are the changes, and that's described by a differential equation. Newton's second law of motion, $ma = f$, is maybe one of the first differential equations written. This is a second order equation, since the acceleration is the second time derivative of the particle position function. Second order differential equations are more difficult to solve than first order equations. second order differential equations of a particular type: those that are linear and have constant coefficients. Such equations are used widely in the modelling of physical phenomena, for example, in the analysis of vibrating systems and the analysis of electrical circuits. The Rungekutta method This method was developed by the German scientists Rang and Cotta. One of the advantages of this method is to avoid the derivatives of the higher ranks that we used in Tyler's method, as this method has two types: Rungekutta of the second rank. Rungekutta of the fourth rank. The fourth Runga - Kutta is more popular and widely used due to its precise accuracy. Until now, we've been trying to understand how to give some instructions to the PC using C++, in this presentation, we're going to study this instructions applied to an specific problem, in this case it will be: solve differential equations. MATLAB (Matrix Laboratory): MATLAB is a platform for scientific calculation and high-level programming which uses an interactive environment that allows you to conduct complex calculation tasks more efficiently than with traditional languages, such as C, C++ and FORTRAN. It is the one of the most popular platforms currently used in the sciences and engineering MATLAB is an interactive high-level technical computing environment for algorithm development, data visualization, data analysis and numerical analysis. MATLAB is suitable for solving problems involving technical calculations using optimized algorithms that are incorporated into easy to use commands.

2. Second order Ordinary Differential Equations:

An equation of the form$a\frac{d^2y}{dx^2} + b\frac{dy}{dx} + cy$, where a, b and c are constants, is called a linear second order differential equation with constant coeffi cients. When the right-hand side of the differential equation is zero, it is referred to as a homogeneous differential equation. When the right-hand side is not equal to zero it is referred to as a non-homogeneous differential equation. There are numerous engineering examples of second order differential equations. [13, p473]

Definition(2.1):

A second order linear differential equation for the function y is

$$\acute{y} + a_1(t)\acute{y} + a_0(t)y = b(t) \qquad (2.1)$$

where a_0, a_1, b are given functions on the interval $I \subset \mathbb{R}$ The Eq. (2.1) above :

(i) is homogeneous iff the source $b(t) = 0$ for all $t \in \mathbb{R}$.

(ii) has constant coefficients iffa_1 and a_0 are constants.

(iii) has variable coefficients iff either a_1 ora_0 is not constant.

3. Homogeneous Second Order Linear Ordinary Differential Equations with Constant Coefficients:

A homogeneous 2nd order linear ode with constant coefficients is given by

$$ay''(x) + by'(x) + cy(x) = 0 \qquad (3.1)$$

Here $a \neq 0, b$ and c are given real constants.

$$y"(x) + \frac{b}{a}y'(x) + \frac{c}{a}y(x) = 0$$

we can construct the general solution of (2.2) by finding two linearly independent solutions. Tofind a solution of , let us try

$$y(x) = e^{\lambda x}$$

where λ is a constant.

Differentiating, we obtain

$$y'(x) = \lambda e^{\lambda x}$$
$$y''(x) = \lambda.\lambda e^{\lambda x} = \lambda^2 e^{\lambda x}$$

Substituting into (3.1), we find that

$$a\lambda^2 e^{\lambda x} + b\lambda e^{\lambda x} + ce^{\lambda x} = 0$$
$$e^{\lambda x}[a\lambda^2 + b\lambda + c] = 0$$

The ODE in (3.1) is satisfied for all x if

$$a\lambda^2 + b\lambda + c = 0$$

We consider the following cases.

Case (3.1):$b^2 - 4ac > 0$

Now if $b^2 - 4ac > 0$, then the quadratic equation in λ has two distinct real solutions given by

$$\lambda = \lambda_1 \equiv \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$
$$\lambda = \lambda_2 \equiv \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

and we obtain two solutions of (3.1) as given by

Since $y_2/y_1 = e^{\lambda_2 x}/e^{\lambda_1 x} = e^{[\lambda_2 - \lambda_1]}$ is not a constant as $\lambda_1 \neq \lambda_2$, the two solutions $y_1$and y_2 are linearly independent. If $b^2 - 4ac > 0$ tells us the general solutionof (3.1) is given by

$$y = Ae^{\lambda_1 x} + Be^{\lambda_2 x}$$

where A and B are arbitrary constants and λ_1 and λ_2 are the two distinct real solutions of the quadratic equation$a\lambda^2 + b\lambda + c = 0$

Case $(\mathbf{3.2})$: $b^2 - 4ac = 0$

For this particular case, the quadratic equation $a\lambda^2 + b\lambda + c = 0$ has only one real solution given by $\lambda = \lambda_1 \equiv -b/(2a)$. Hence, in seeking a solution of the form$y = e^{\lambda x}$ we obtain only one solution of (3.1), that is, $y_1 = e^{-bx/(2a)}$. To construct the general solution of (3.1), another solution which is linearly independent to y_1 is needed. To find another solution, we let

$$y = u(x).e^{\lambda_1 x}$$

where $u(x)$ is a function to bedetermined Differentiating, we obtain

$$y' = \lambda_1 u(x).e^{\lambda_1 x} + u'(x).e^{\lambda_1 x}$$

$$y'' = {\lambda_1}^2 u(x).e^{\lambda_1 x} + 2\lambda_1 u'(x).e^{\lambda_1 x} + u''(x)e^{\lambda_1 x}.$$

Substituting into (3.1), we find that

$$a[{\lambda_1}^2 u(x).e^{\lambda_1 x} + 2\lambda_1 u'(x).e^{\lambda_1 x} + u''(x)e^{\lambda_1 x}] + b\left(\lambda_1 u(x).e^{\lambda_1 x} + u'(x).e^{\lambda_1 x}\right) + cu(x)e^{\lambda_1 x} = 0$$

which may be rewritten as

$$\left[\left(a{\lambda_1}^2 + b\lambda_1 + c\right)u(x) + (2\lambda_1 a + b)u'(x) + au''(x)\right]e^{\lambda_1 x} = 0.$$

Since $a{\lambda_1}^2 + b\lambda_1 + c = 0$ and $\lambda_1 = -b/(2a)$ the equationabove reduces to

$$au''(x)e^{\lambda_1 x} = 0.$$

Since $a \neq 0$ and $e^{\lambda_1 x}$we obtain the ordinary differential equations

$$u''(x) = 0$$

A solution of this simple ordinary differential equations is $u(x) = x$. Summarizing, if $b^2 - 4ac = 0$, two particular solutions of (2.2) are given by $y_1 = e^{-bx/(2a)}$and $y_2 = xe^{-bx/(2a)}$

Since $y_2/y_1 = x$ (not a constant), the two solutions above are linearly independent. the required general solution of the ODE is$y = Ae^{-bx/(2a)} + Bx.e^{-bx/(2a)}$

where A and B are arbitrary constants.

Case (3.3): $b^2 - 4ac < 0$

For this case, the quadratic equation$a\lambda^2 + b\lambda + c = 0$does not have any real solutions. It has two distinct complex solutions given by

$$\lambda = \lambda_1 \equiv -\frac{b}{2a} + i\frac{\sqrt{|b^2 - 4ac|}}{2a}$$

$$\lambda = \lambda_2 \equiv -\frac{b}{2a} - i\frac{\sqrt{|b^2 - 4ac|}}{2a}$$

where$i = \sqrt{-1}$.

If we ignore the fact thatλ_1 and$\lambda_2$are complex and proceed as in Case I above, the general solution of (3.1) is given by

$$y = Ae^{\lambda_1 x} + Be^{\lambda_2 x}$$

where A and B are arbitrary constants [25, p55-58]

4.Complementary Function and Particular Integral:

If in the differential equation

$$a\frac{d^2y}{dx^2} + b\frac{dy}{dx} + cy = f(x) \qquad (4.1)$$

the substitution$y = u + v$ is made then:

$$a\frac{d^2(u+v)}{dx^2} + b\frac{d(u+v)}{dx} + c(u+v) = f(x)$$

Rearranging gives:

$$\left(a\frac{d^2u}{dx^2} + b\frac{du}{dx} + cu\right) + \left(a\frac{d^2v}{dx^2} + b\frac{dv}{dx} + cv\right) = f(x)$$

If we let

$$a\frac{d^2v}{dx^2} + b\frac{dv}{dx} + cv = f(x) \qquad (4.2)$$

Then

$$a\frac{d^2u}{dx^2} + b\frac{du}{dx} + cu = 0 \qquad (4.3)$$

The general solution, u. of equation (4.3) will contain two unknown constants, as required for the general solution of equation (4.1). The function u is called the complementary function (C. F.). If the particular solution, v, of equation (4.2) can be determined without containing any unknown constants then $y = u + v$will give the general solution of equation (4.1). The function v is called the particular integral (P. I.). Hence the general solution of equation (4.1) is given by:

$\boldsymbol{y = C.F. + P.I.}$ **[13, p483]**

5. Runga - Kutta's Method:

Fundamentally, all Runge Katta methods are generalizations of the basic Euler formula (5.1) of in that the slope function $f$is replaced by a weighted average of slopes over the interval $x_n \le x \le x_{n+1}$. That is

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2 + \cdots = w_mk_m) \qquad (5.1)$$

Here the weights$w_i, i = 1,2, \dots, m,$are constants that generally satisfy$w_1 + w_2 + \cdots + w_m = 1,$ and each $k_i, i = 1,2, \dots m,$is the function $f$evaluated at a selected point $(x, y)$for which $x_n \le x \le x_{n+1}$.We shall see that the $k_i$are defined recursively. The number m is called the order of the method. Observe that by taking $m = 1, w_1 = 1,$ and $k_1 = f(x_n, y_n),$ we get the familiar Euler formula$y_{n+1} = y_n hf(x_n, y_n)$ Hence Euler's method is said to be a first-order RungeKatta method. The average in (5.1) is not formed willy-nilly, but parameters are chosen so that (5.1) agrees with a Taylor polynomial of degree m. As we saw in the preceding section, if a function$y(x)$ possesses $k + 1$ derivatives that are continuous on an open interval containing $a$and $x$then we can write

$$y(x) = y(a) + y'(a)\frac{x-a}{1!} + y''(a)\frac{(x-a)^2}{2!} + \cdots + y^{(k+1)}(c)\frac{(x-a)^{k+1}}{(k+1)!},$$

where c is some number between a and x.If we replace a by x_n and x by$x_{n+1} = x_n + h,$ then the foregoing formula becomes

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \cdots + \frac{h^{k+1}}{(k+1)!}y^{(k+1)}(c),$$

where c is now some number between x_n andx_{n+1}When $y(x)$ is a solution of $y' = f(x, y)$ in the case $k = 1$ and the remainder $\frac{1}{2}h^2y''(c)$ is small, we see that a Taylor polynomial $y(x_n + 1) = y(x_n) + hy'(x_n)$ of degree one agrees with the approximation formula of Euler's method

$$y_{n+1} = y_n + hy'_n = y_n + hf(x_n, y_n).$$

6.Second Runga - Kutta's Method:

To further illustrate (5.1), weconsider now a second-order RungeKatta procedure. This consists of findingconstants or parameters w_1, w_2, α and β so that the formula

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2), \qquad (5.1)$$

Where $k_1 = f(x_n, y_n)$

$$k_2 = f(x_n + \alpha h, y_n + \beta hk_1),$$

agrees with a Taylor polynomial of degree two. For our purposes it suffices to say that this can be done whenever the constants satisfy

$w_1 + w_2 = 1, w_2\alpha = \frac{1}{2},$ and $w_2\beta = \frac{1}{2}$(6.1)

This is an algebraic system of three equations in four unknowns and has infinitelymanysolutions:

$$w_1 = 1 - w_2, \qquad \alpha = \frac{1}{2w_2}, \quad and\ \beta = \frac{1}{2w_2}, \qquad (6.2)$$

where $w_2 \neq 0$ the choice $w_2 = \frac{1}{2}$,yields $w_1 = \frac{1}{2}, \alpha = 1,and\beta = 1,$so (5.1) becomes

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2),$$

Where

$k_1 = f(x_n, y_n)$ and $k_2 = f(x_n + h, y_n + hk_1).$

Since $x_n + h = x_{n+1}$ and $y_n + hk_1 = y_n + hf(x_n, y_n),$ the foregoing result is rec-ognized to be the improved Euler's method.[2,p345]

7.Fourth-Order Runga - Kutta's Method:

The fourth-order Runge-Kutta method is obtained from the Taylor series along the same lines as the second-order method. Since the derivation is rather long and not very instructive, we skip it. The final form of the integration formula again depends on thechoice of the parameters; that is, there is no unique Runge-Kutta fourth-order formula.The most popular version, which is known simply as the Runga - Kutta's method, entailsthe following sequence of operations:

$$K_1 = hF(x, y)$$
$$K_2 = hF(x + \frac{h}{2}, y + \frac{K_1}{2})$$
$$K_3 = hF(x + \frac{h}{2}, y + \frac{K_2}{2})$$
$$K_4 = hF(x + h, y + K_3)$$

$$y(x+h) = y(x) + \frac{1}{6}(K_1 + K_2 + K_3 + K_4)$$

The main drawback of this method is that it does not lend itself to an estimate of the truncation error. Therefore, we must guess the integration step size h, or determine it by trial and error. In contrast, the so-called adaptive methods can evaluate the truncation error in each integration step and adjust the value of h accordingly (but at a higher cost of computation).[12, p260]

8.Runga-Kutta Approximations for EquationsSubject to Initial Conditions:

The Runga - Kutta's method was introduced to approximate the solution to a first-order differential equation. It can also be used to estimate the solution of a second order differential equation by expressing it as two first-order equations.

For

$$y'' + P(x)y' + Q(x)y = f(x, y, y') \qquad (8.1)$$

constrained by the initial conditions

$$y(x_0) = y_0$$
$$y'(x_0) = y'_0$$

we define the first-order differential equation for $y(x)$

$$y'(x) \equiv w(x) \qquad (8.2)$$

With this definition, the second order differential equation for $y(x)$ given in eq. can be written as a first order equation for $w(x)$:

The initial conditions for $y(x)$ become

$$y(x_0) = y_0$$
$$w(x_0) = w_0 = y'_0$$

The Runga - Kutta's method for solving a second order differential equation involves defining two sets of Runga-Kutta parameters, R and S. With the values of y_0 and w_0 given by initial conditions, we determine the value of R_1. With R_1 we find S_1. Then $S_1 \Rightarrow R_2 \Rightarrow \cdots \Rightarrow R_4 \Rightarrow S_4$

we begin with

$$R_1 = y'(x_0)h = w(x_0)h$$

S_1, the Runga -Kutta parameter for the first-order equation for $w(x)$given is

$$S_1 = G(x_0, y_0, w_0)h = [f(x_0, y_0, w_0) - P(x_0)w_0 - Q(x_0)y_0]h$$

From these, we determine

$$R_2 = \left(w_0 + \frac{1}{2}S_1\right)h$$

and

$$S_2 = G\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}R_1, w_0 + \frac{1}{2}S_1\right)h$$
$$= \left[f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}R_1, w_0 + \frac{1}{2}S_1\right) - P\left(x_0 + \frac{1}{2}h\right)\left(w_0 + \frac{1}{2}S_1\right) - Q\left(x_0 + \frac{1}{2}h\right)\left(y_0 + \frac{1}{2}R_1\right)\right]h$$

Then

$$R_3 = \left(w_0 + \frac{1}{2}S_2\right)h$$

and

$$S_3 = G\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}R_2, w_0 + \frac{1}{2}S_2\right)h$$
$$= \left[f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}R_2, w_0 + \frac{1}{2}S_2\right) - P\left(x_0 + \frac{1}{2}h\right)\left(w_0 + \frac{1}{2}S_2\right) - Q\left(x_0 + \frac{1}{2}h\right)\left(y_0 + \frac{1}{2}R_2\right)\right]h$$

Then

$$R_4 = (w_0 + S_3)h$$

and

$S_4 = G(x_0 + h, y_0 + R_3, w_0 + S_3)h = [f(x_0 + h, y_0 + R_3, w_0 + S_3) - P(x_0 + h)(w_0 + S_3) - Q(x_0 + h)(y_0 + R_3)]h$

From these parameters, we obtain

$$y(x) = y'(x_0 + h) = y_0 + \frac{1}{6}[R_1 + 2R_2 + 2R_3 + R_4]$$

$w(x) = y'(x_0 + h) = w_0 + \frac{1}{6}[S_1 + 2S_2 + 2S_3 + S_4]$ [10, p273-274]

Example (8.1):

Runga -Kutta approximations for an equation subject to initialconditions

We again consider

$$y'' - 2xy' - y = y^2 e^{-x^2}$$

subject to

$$y(0) = y_0 = 1$$
$$y'(0) = w_0 = 0$$

the solution to which is

$$y(x) = e^{x^2}$$

From this solution, we have the exact values

$$y(0.50) = e^{(0.50)^2} = 1.28403$$

and

$$w(0.50) = y'(0.50) = 2(0.50)e^{(0.50)^2} = 1.28403$$

The corresponding pair of first-order equations for ea. 7.6a are

$$y'(x) = w(x)$$

and

$$w' = y^2 e^{-x^2} + 2xw + y$$

With $x_0 = 0$ and $h = 0.5$, the R and S parameters are

$$R_1 = w_0 h = 0$$

$$S_2 = \left[\left(y_0 + \frac{1}{2}R_1\right)^2 e^{-\left(x_0+\frac{1}{2}h\right)^2} + 2\left(x_0 + \frac{1}{2}h\right)\left(w_0 + \frac{1}{2}S_1\right) + \left(y_0 + \frac{1}{2}R_1\right)\right] h = 1.09471$$

$$R_3 = \left(w_0 + \frac{1}{2}S_2\right) h = 0.27368$$

$$S_3 = \left[\left(y_0 + \frac{1}{2}R_2\right)^2 e^{-\left(x_0+\frac{1}{2}h\right)^2} + 2\left(x_0 + \frac{1}{2}h\right)\left(w_0 + \frac{1}{2}S_2\right) + \left(y_0 + \frac{1}{2}R_2\right)\right] h = 1.29381$$

$$R_4 = (w_0 + S_3)h = 0.64691$$

$$S_4 = \left[\left(y_0 + R_3\right)^2 e^{-(x_0+h)^2} + 2(x_0 + h)(w_0 + S_3) + (y_0 + R_3)\right] h = 1.91545$$

Then

$$y(0.50) \simeq y_0 + \frac{1}{6}[R_1 + 2R_2 + 2R_3 + R_4] = 1.28238$$

and

$$w(0.50) \simeq w_0 + \frac{1}{6}[S_1 + 2S_2 + 2S_3 + S_4] = 1.28208$$

To develop a multiple step Runga -Kutta process, we take $h = 0.25$ and apply the above process to find $y(0.25)$ and $w(0.25)$ Then setting $x_0 = 0.25$ and $y_0 = 0.25$ and applying the Runge-Kutta method to find $y(0.25)$and $w(0.25)$ we obtain

$$y(0.25) \simeq 1.28382$$

And

$y(0.25) \simeq 1.28378$ [10, p275]

9.MATLAB(New Mathematical Technique):

is a powerful language for technical Computing The name MATLAB Stands for matrix laboratory because its basic data element is matrix.MATLAB Can be used for math Computations modeling and simulation data analysis and processing visualization and graphic and algorithm develpmcut.MATLAB is widely used in universities and colleges in introduction and advanced courses in mathematics and especially engineering. MATLAB program has tools that. Can be used be solve common problems. [1]

i.Solve Part Differential Equation with MATLAB:

The MATLAB partial differential equation solver *pdepe* initial boundary value problems for systems of Parabolic and elliptic part differential equation in the one space variable and time . there must be at lead on parabolic equation in the system. The *pdepe* solver converts the Pdf to ode using a second order accurate spatial discretization based on a set of nodes specified by the use. [19]

ii.Finding Explicit Solutions:

MATLAB has an extensive library of functions for solving ordinary differential equations. In these notes, we will only consider the most rudimentary.[11,p1]

iii.First Order Equation:

Though MATLAB is primarily a numeric package, it can certainly solve straightforward differential equations symbolically[11] Suppose, for example, that we want to solve the first order differential equation

$$y'(x) = xy.$$

We can use MATLAB's built-in ( ).The input and output for solving this problem in MATLAB is given below.

$$>> y = dsolve('Dy = y^*x','x')$$
$$y = C1^*\exp(1/2^*x\text{^}2)$$

Notice in particular that MATLAB uses capital D to indicate the derivative and requires that the entire equation appear in single quotes. MATLAB takes t to be the independent variable by default, so here must be explicitly specified as the independent variable. Alternatively,if you are going to use the same equation a number of times, you might choose to define it as a variable, $eqn1$

$$>> eqn1 = ' Dy = y^*x'$$
$$eqn1 =$$
$$Dy = y * x$$
$$>> y = dsolve(eqn1,'x')$$
$$y = C1^*\exp(1/2^*x\text{^}2)$$

To solve an initial value problem, say, equation () with $y(1) = 1$, use

$$>> y = dsolve(eqn1,' y(1) = 1','x'$$
$$y = 1/exp\left(\frac{1}{2}\right)^* \exp(1/2^*x\text{^}2)$$

Or

$$>> inits = ' y(1) = 1';$$
$$>> y = dsolve(eqn,inits,'x')$$
$$y = 1/exp\left(\frac{1}{2}\right)^* \exp(1/2^*x\text{^}2)$$

Now we've solved the ODE, suppose we want to plot the solution to get a rough idea of its behavior. We run immediately into two minor difficulties: (1) our expression for$y(x)$ isn't suited for array operations (.*, ./, .^) and (2) as MATLAB returns it, is actually a symbol (a symbolic object). The first of these obstacles is straightforward to fix, using$vectorize($).

For the second, we employ the useful command$eval($), , which evaluates or executes text strings that constitute valid MATLAB commands. Hence, we can use

$$\gg linspace(0,1,20);$$
$$>> z = eval(vectorize(y));$$
$$>> plot(x,z)$$

You may notice a subtle point here, that evaluates strings (character arrays), and , as we have defined it, is a symbolic object. However, vectorize converts symbolic objects into strings.

iv.Second and Higher Order Equations:

Suppose we want to solve and plot the solution to the second order equation

$$y''(x) + 8y'(x) + 2y(x) = \cos(x)\,; \qquad y(0) = 0, y'(0) = 1$$

The following (more or less self-explanatory) MATLAB code suffices:

$$>> eqn2 = ' D2y + 8^*Dy + 2^*y = \cos(x)'\,;$$
$$>> inits2 = ' y(0) = 0, Dy(0) = 1';$$
$$>> y = dsolve(eqn2,inits2,'x')$$
$$y =$$
$$1/65^*\cos(x) + 8/65^*\sin(x) + (-1/130 + 53/1820 * 14\text{^}(1/2))^*\exp((-4 + 14\text{^}(1/2))^*x) - 1/1820 * (53 + 14\text{^}(1/2))^*14\text{^}(1/2)^*\exp(-(4 + 14\text{^}(1/2))^*x)$$
$$>> z = eval(vectorize(y));$$
$$>> plot(x,z)$$

[11,p2-3]

10.Calculation of the Second order Ordinary Differential Equations Using Runga - Kutta's Method by A New Mathematical Technique :

Example(10.1):

Using Runga - Kutta's method solve $y'' = xy'^2 - y^2$ for $x = 0.2$ correct to 4 decimal places initial conditions $y(0) = 1, y'(0) = 0$

Solution:

$$h = 0.2 - 0 = 0.2 \qquad , x_0 = 0, \qquad y_0 = 1, \qquad y'_0 = 0$$

let $y' = z$ then $y'_0 = z_0 = 0$

$$y' = z = f(x,y,z)$$
$$y'' = z' = xz^2 - y^2$$
$$z' = g(x,y,z) = xz^2 - y^2$$
$$K_1 = hf(x_0,y_0,z_0) = 0.2(z_0) = 0.2(0) = 0$$

$$L_1 = hg(x_0, y_0, z_0) = hg(0,1,0) = 0.2((0)(0) - (1^2)) = 0.2(-1) = -0.2$$

$$K_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{K_1}{2}, z_0 + \frac{L_1}{2}\right) = 0.2f(0.1,1,-0.1) = 0.2(-0.1) = -0.02$$

$$L_2 = hg\left(x_0 + \frac{h}{2}, y_0 + \frac{K_1}{2}, z_0 + \frac{L_1}{2}\right) = hg(0.1,1,-0.1) = 0.2[(0.1)(-0.1^2) - 1^2] = -0.1998$$

$$K_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{K_2}{2}, z_0 + \frac{L_2}{2}\right) = hf(0.1,0.99,-0.999) = 0.2(-0.0999) = -0.01998 = -0.02$$

$$L_3 = hg\left(x_0 + \frac{h}{2}, y_0 + \frac{K_2}{2}, z_0 + \frac{L_2}{2}\right) = 0.2g(0.1,0.99,-0.0999) = -0.1958$$

$$K_4 = hF(x_0 + h, y_0 + K_3, z_0 + L_3) = 0.2(-0.1958) = -0.03916 = -0.0392$$

$$L_4 = hg(x_0 + h, y_0 + K_3, z_0 + L_3) = 0.2g(0.2,0.98,-0.1958) = -0.1905$$

$$y(x + h) = y_0(x) + \frac{1}{6}(K_1 + K_2 + K_3 + K_4)$$

$$K = \frac{1}{6}(0 + 2(-0.02) + 2(-0.02) \pm 0.0392) = -0.0199$$

$$y = y_0 + 1 - 0.0199 = 0.98 = 0.2$$

$$L = \frac{1}{6}(L_1 + 2L_2 + 2L_3 + L_4) = \frac{1}{6}(-0.2 = 2(-0.1998) + 2(-0.1958) \pm 0.1905) = -0.1970$$

$$z = z_0 + L = 0 - 0.1970$$

$$y' = z = -0.1970$$

Solution:

```
%y"=x*y'^2-y^2
%let w"=x*w'^2-w^2 and w'=z
Clc
clear all
h=0.2;
t(1)=0;
z(1)=0;
w(1)=1;
f = @(t,w,z) (z);
g = @(t,w,z) (t*z^2-w^2);
forI =1:40
L1 = h*g(t,w,z);
k1 = h*f(t,w,z);
L2 = h*g(t+h/2,w+k1/2,z+L1/2);
k2 = h*f(t+h/2,w+k1/2,z+L1/2);
L3 = h*g(t+h/2,w+k2/2,z+L2/2);
k3 = h*f(t+h/2,w+k2/2,z+L2/2);
L4 = h*g(t+h,w+k3,z+L3);
k4 = h*f(t+h,w+k3,z+L3);

z=z+(L1+2*L2+2*L3+L4)/6;
w=w+(k1+2*k2+2*k3+k4)/6;
t=t+h;
    fprintf('i=%8.0f, t=%8.2f, w=%8.6f, z=%8.6f\n', I, t,w,z)      plot(t,z,'-*r')
hold on
plot(t,w,'-ob')
end
```

Results:

```
i=       3, t=    0.60, w=0.833498, z=-0.508421
i=       4, t=    0.80, w=0.722980, z=-0.587046
i=       5, t=    1.00, w=0.602450, z=-0.609743
i=       6, t=    1.20, w=0.481994, z=-0.588975
i=       7, t=    1.40, w=0.368612, z=-0.541798
i=       8, t=    1.60, w=0.266023, z=-0.483168
i=       9, t=    1.80, w=0.175418, z=-0.423260
i=      10, t=    2.00, w=0.096418, z=-0.367768
```

i= 11, t= 2.20, w=0.027843, z=-0.319251
i= 12, t= 2.40, w=-0.031795, z=-0.278427
i= 13, t= 2.60, w=-0.084024, z=-0.245070
i= 14, t= 2.80, w=-0.130277, z=-0.218549
i= 15, t= 3.00, w=-0.171847, z=-0.198113
i= 16, t= 3.20, w=-0.209877, z=-0.183032
i= 17, t= 3.40, w=-0.245371, z=-0.172651
i= 18, t= 3.60, w=-0.279211, z=-0.166405
i= 19, t= 3.80, w=-0.312175, z=-0.163815
i= 20, t= 4.00, w=-0.344952, z=-0.164472
i= 21, t= 4.20, w=-0.378156, z=-0.168022
i= 22, t= 4.40, w=-0.412332, z=-0.174147
i= 23, t= 4.60, w=-0.447966, z=-0.182564
i= 24, t= 4.80, w=-0.485492, z=-0.193009
i= 25, t= 5.00, w=-0.525289, z=-0.205246
i= 26, t= 5.20, w=-0.567695, z=-0.219065
i= 27, t= 5.40, w=-0.613008, z=-0.234287
i= 28, t= 5.60, w=-0.661494, z=-0.250772
i= 29, t= 5.80, w=-0.713395, z=-0.268421
i= 30, t= 6.00, w=-0.768937, z=-0.287171
i= 31, t= 6.20, w=-0.828336, z=-0.306995
i= 32, t= 6.40, w=-0.891808, z=-0.327892
i= 33, t= 6.60, w=-0.959569, z=-0.349885
i= 34, t= 6.80, w=-1.031841, z=-0.373007
i= 35, t= 7.00, w=-1.108855, z=-0.397305
i= 36, t= 7.20, w=-1.190851, z=-0.422826
i= 37, t= 7.40, w=-1.278079, z=-0.449626
i= 38, t= 7.60, w=-1.370801, z=-0.477760
i= 39, t= 7.80, w=-1.469289, z=-0.507286
i= 40, t= 8.00, w=-1.573828, z=-0.538262

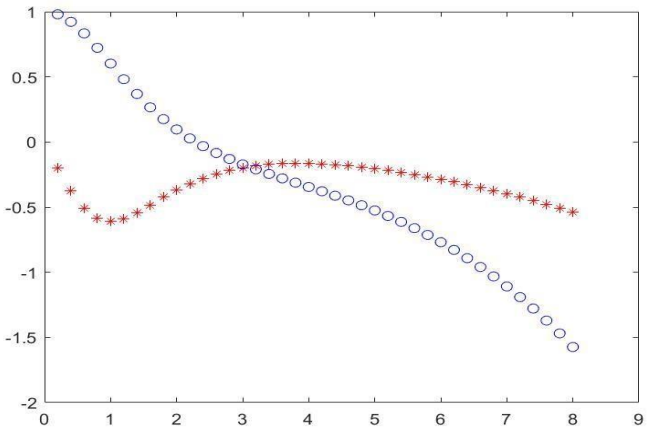

Fig No(10.1): Calculating using Runga - Kutta's by MATLAB

Results :

The possibility of solving the ordinary equations using Runga - Kutta's method by a new mathematical technique , reducing errors , speed and accuracy of the solution .

References:

[1]. Amos Gilat, MATLAB and Introduction with Application, Fourth Edition, johnwibey Sonlue,2011.
[2]. Dennis G. Zill and Michael R. Cullen, Differential Equations with Boundary Value Problems, Brooks/Cole Cengage learning , Seventh edition,USA,2008,p343.
[3]. D. Zill and W. Wright, Differential Equations with Boundary Value Problems, Brooks/Cole ,,Boston,8th edition.
[4]. E. Coddington, An Introduction to Ordinary Differential Equation, prentice Hall,1961.
[5]. E. Zeidler, Nonlinear Functional Analysis and its Applications I, Fixed-Point Theorems. Springer, New York, 1986.
[6]. E. Zeidler, Applied functional analysis, applications to mathematical physics, Springer, New York, 1995.
[7]. G. Simmons, Differential equations with applications and historical notes, McGraw-Hill, New York,2nd edition ,1991.

[8]. G. Thomas, M. Weir, and J. Hass, Thomas' Calculu , Pearson,12th edition.
[9]. G. Watson. A treatise on the theory of Bessel functions. Cambridge University Press, London, 2nd edition,1944.
[10]. Harold Cohen, Numerical Approximation Methods, springer,London,2010,p273-277.
[11]. Howard, Solving ODE in MATLAB, fall, 2007, p1-3.
[12]. JaanKiusalass, Numerical Methods in engineering with MATLAB, Cambridge, NewNork ,2005, p260.
[13]. John Bird, Higher Engineering Mathmatics, Newne is an imprint of Elsevier,sixth edition,USK,2010,p477.
[14]. J.D. Jackson. Classical Electrodynamics. Wiley, New Jersey, 3rd edition,1999.
[15]. J. Stewart, Multivariable Calculus, Cenage Learning, 7th edition.
[16]. Math work, using MATLAB, versin ,6,2001.
[17]. Math work, Getting stated with MATLAB, second printing, may 1997.
[18]. R. Churchill. Operational Mathematics. McGraw-Hill, New york, Second Edition,1958.
[19]. S. Hassani , Mathematical physics. Springer, New York, Corrected second printing, 2000.
[20]. S. Strogatz, Nonlinear Dynamics and Chaos, Perseus Books Publishing, Cambridge, USA, 1994.
[21]. T. Apostol. Calculus. John Wiley & Sons, New York, Volume I, Second edition,1967.
[22]. T. Apostol. Calculus. John Wiley & Sons, New York, Volume II, Second edition,1969.
[23]. W. Boyce and R. DiPrima, Elementary differential equations and boundary value problems, Wiley, New Jersey, 10th edition ,2012.
[24]. W. Rudin, Principles of Mathematical Analysis, McGraw-Hill, New York, NY, 1953.
[25]. W.T. Ang and Y.S,Park , Ordinary differential equations, universal publisher,USA,2008,p55.