Group Number - 12
Pardheev - 21110097
Himasagar - 21110158
Keerthi - 21110176
Sudharshan - 21110218
Sai Ram - 21110241

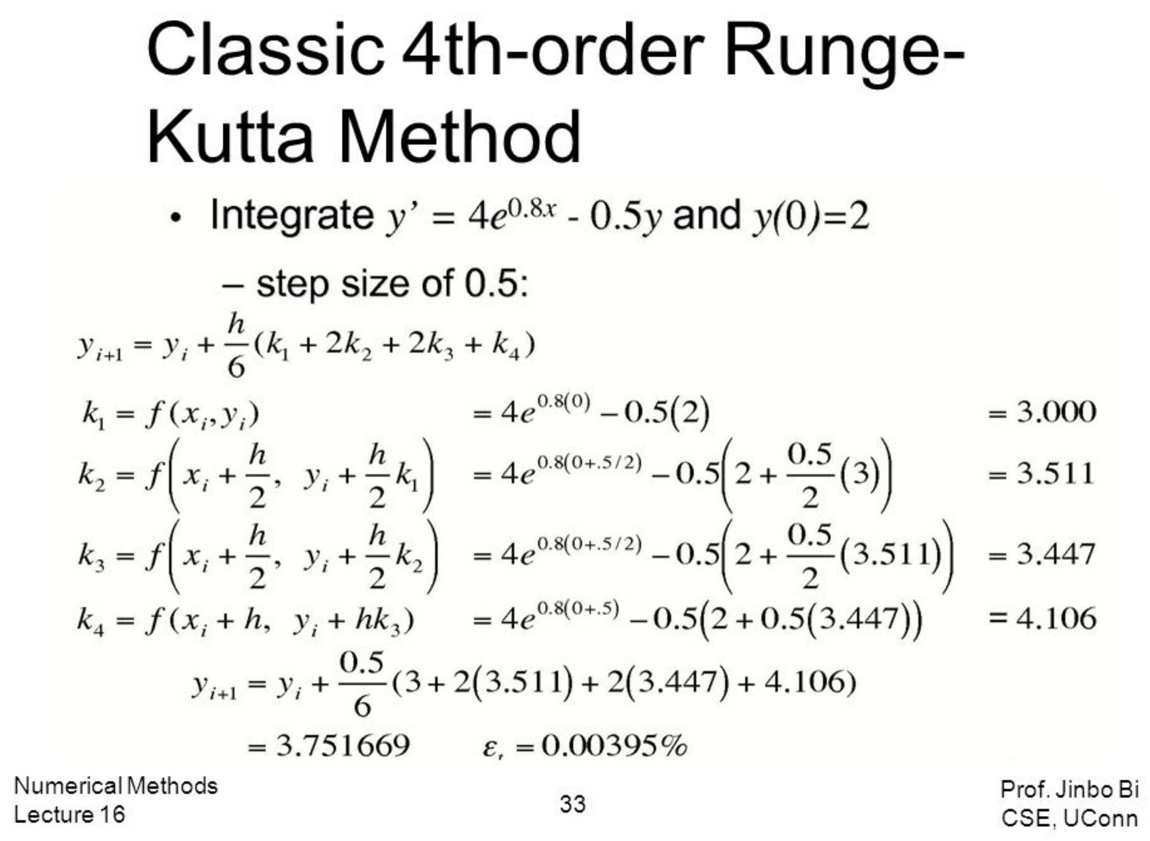
NUMERICAL ANALYSIS OF 3-BODY DYNAMICS IN THE PRESENCE OF GRAVITATIONAL INTERACTIONS

The three-body issue is a mathematical model that represents the motion of three-point mass particles in the presence of reciprocal gravitational interactions. This is a typical problem in astrodynamics that covers a wide range of scenarios. The motion of the Moon around the Earth under the influence of the Sun is an example of such a circumstance. Three bodies move in space due to reciprocal gravitational interactions described by Newton's theory of gravity in the three-body problem. To solve this difficulty, the bodies' future and previous movements must be uniquely identified based merely on their positions.

Positions and velocities are currently being displayed. In general, body movements occur in three dimensions (3D), with no limits on their masses or beginning circumstances. As a result, this is known as the universal three-body issue. The intricacy of the problem does not appear to be clear at first look, especially given that the two-body problem has well-known closed form solutions presented in terms of elementary functions.
Adding one more body complicates the problem and makes it impossible to achieve comparable sorts of solutions. Many physicists, astronomers, and mathematicians have tried and failed in the past to develop closed form solutions to the three-body issue. Such solutions do not exist since the movements of the three bodies are in general unexpected, making the three-body issue one of the most difficult problems in scientific history.

Kinetic Energy(K.E) =½(mv^2)
Potential Energy(P.E)=-Gm1m2/r

Runge Kutta :
The Runge-Kutta method is a reliable and popular approach for resolving differential equation initial-value issues. Without the necessity for high order derivatives of functions, the Runge-Kutta method can be utilized to build high order accurate numerical methods by functions alone.



Classic 4th-order Runge-Kutta Method

- Integrate $y' = 4e^{0.8x} - 0.5y$ and $y(0)=2$
  - step size of 0.5:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(x_i, y_i) \qquad = 4e^{0.8(0)} - 0.5(2) \qquad = 3.000$$

$$k_2 = f\left(x_i + \frac{h}{2},\ y_i + \frac{h}{2}k_1\right) \qquad = 4e^{0.8(0+.5/2)} - 0.5\left(2 + \frac{0.5}{2}(3)\right) \qquad = 3.511$$

$$k_3 = f\left(x_i + \frac{h}{2},\ y_i + \frac{h}{2}k_2\right) \qquad = 4e^{0.8(0+.5/2)} - 0.5\left(2 + \frac{0.5}{2}(3.511)\right) \qquad = 3.447$$

$$k_4 = f(x_i + h,\ y_i + hk_3) \qquad = 4e^{0.8(0+.5)} - 0.5(2 + 0.5(3.447)) \qquad = 4.106$$

$$y_{i+1} = y_i + \frac{0.5}{6}(3 + 2(3.511) + 2(3.447) + 4.106)$$

$$= 3.751669 \qquad \varepsilon_t = 0.00395\%$$

Numerical Methods
Lecture 16

33

Prof. Jinbo Bi
CSE, UConn

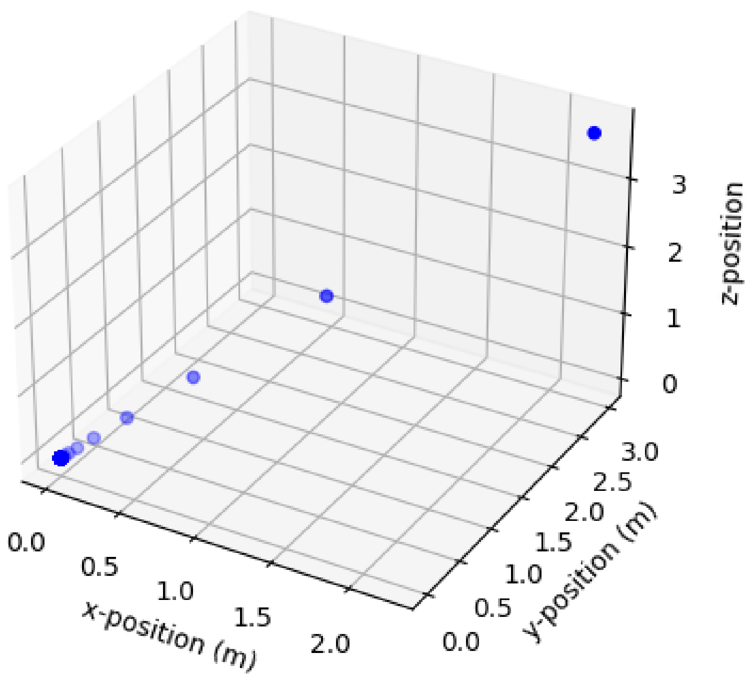
[1]

The 4th order Runge-Kutta follows as shown in the above example we have to calculate k1, k2, k3, k4 and finally k=⅙(k1+2k2+2k3+k4) and from that we have to update the $y_{i+1}=y_i+k$ and from there we the further y values.
We written the code as :

```
def RK4(t,r,v,h,planet,ro,vo):
    k11 = dr_dt(t,r,v,planet,ro,vo)
    k21 = dv_dt(t,r,v,planet,ro,vo)

    k12 = dr_dt(t + 0.5*h,r + 0.5*h*k11,v + 0.5*h*k21,planet,ro,vo)
    k22 = dv_dt(t + 0.5*h,r + 0.5*h*k11,v + 0.5*h*k21,planet,ro,vo)

    k13 = dr_dt(t + 0.5*h,r + 0.5*h*k12,v + 0.5*h*k22,planet,ro,vo)
    k23 = dv_dt(t + 0.5*h,r + 0.5*h*k12,v + 0.5*h*k22,planet,ro,vo)

    k14 = dr_dt(t + h,r + h*k13,v + h*k23,planet,ro,vo)
    k24 = dv_dt(t + h,r + h*k13,v + h*k23,planet,ro,vo)

    y0 = r + h * (k11 + 2.*k12 + 2.*k13 + k14) / 6.
    y1 = v + h * (k21 + 2.*k22 + 2.*k23 + k24) / 6.
    z = np.zeros([2,3])
    z = [y0, y1]
    return z
```

About the Code:

First, we established the force functions that result from one particle acting on another. Later, the force function, the force that ultimately acts on the particle, was defined. Next, the dv_dt and dr_dt functions, which represent the particle's instantaneous acceleration and velocity, were defined. Later, we defined the particle's kinetic and potential energies. The angular momentum function was then defined, the runge-Kutta method was then implemented, all terms were then normalized, and finally the code for accepting inputs was written. Subsequently, we attempted to implement the trajectories by iterating through all of the functions, storing them, and then plotting the trajectory by plotting the positions of the functions. We got the graph as



Problems we faced:

This problem is too simple to implement. For example, if we put one of the elements at rest, the other two will produce a two-body system that is too simple to implement. In addition, we implemented the code for vectors with three components, which includes all of the i, j, and k terms, even though it would be simpler to do so for vectors with only two components.

References:

[1] D.Young, *Today's class Ordinary Differential Equations Runge-Kutta methods*. 2017. Available:https://slideplayer.com/slide/9538872/