FOUCAULT PENDULUM

-Group 2G

Method:

In each method, there will be two vectors, the State vector and the Function vector constantly present. The State vector stores the x and y coordinates along with the x and y velocities, while the Function vector stores the differentiated elements of the State vector.

The equations were:

$x'' - 2\Omega y' + \omega*\omega x=0$

$y'' + 2\Omega x' + \omega*\omega y=0$

(Here prime notation indicates differentiation with respect to time)

State Vector $Y= [\ x, x', y, y'\ ]$

Function Vector $f= [\ x', x'', y', y''\ ]= [x', 2\Omega y'-\omega*\omega x, y', -2\Omega x'-\omega*\omega x\ ]$

So each element correspondence can be taken to be a differential equation-i.e, we are solving 4 different differential equations.

1) Euler's Method:

In Euler's method,

$Y_{n+1} = Y_n + dt*f(Y_n)$ is a step advancement.

2) Heun's Method:

$k1 = f(Y_n)$

$k2 = f(Y_n+dt*k1)$

$Y_{n+1} = Y_n+(dt/2)*(k1+k2)$ describes step advancement.

k1 and k2 are two step vectors

3) Runge-Kutta's Method:

$k1 = f(Y_n)$

$k2 = f(Y_n + (dt/2)*k1)$

$k3 = f(Y_n + (dt/2)*k2)$

$k4 = f(Y_n+dt*k3)$

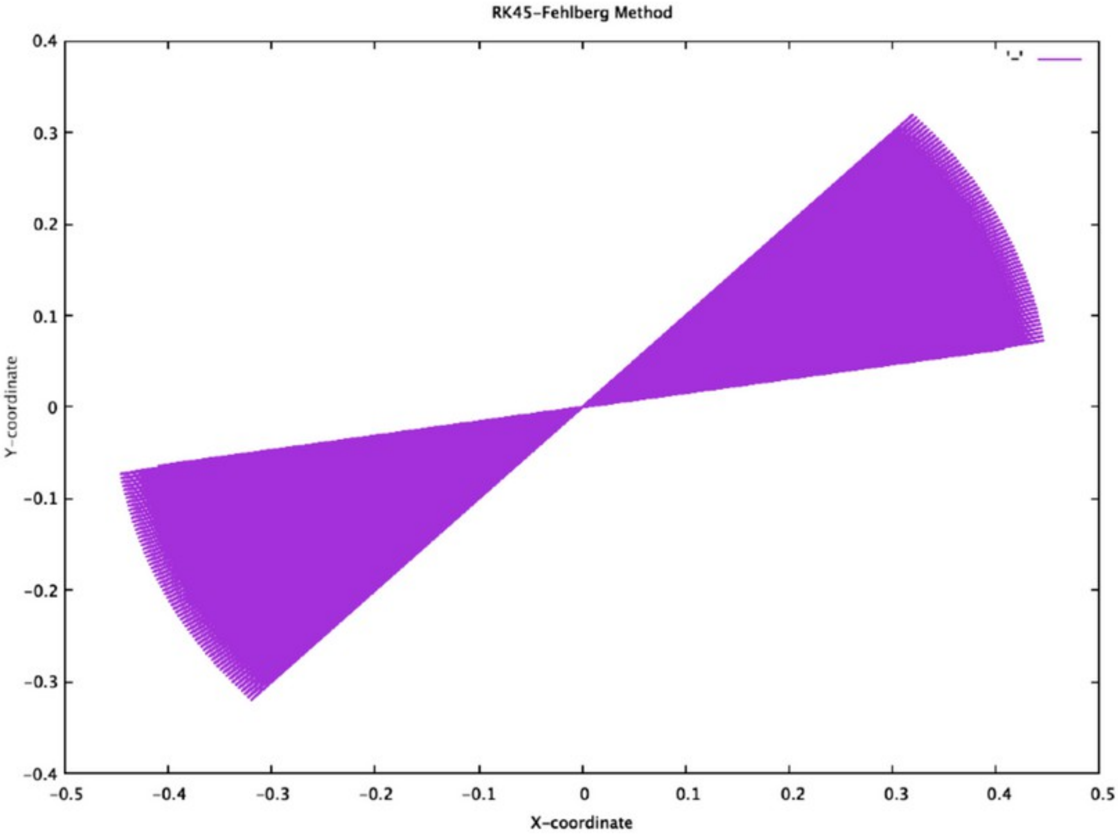
$Y_{n+1} = Y_n + (dt/6)*(k1 +2k2 +2k3 +k4)$

4) RK45 (Fehlberg's) Method:

Here the coefficients used were developed by Cash and Karp which are mentioned in the Chapra textbook.

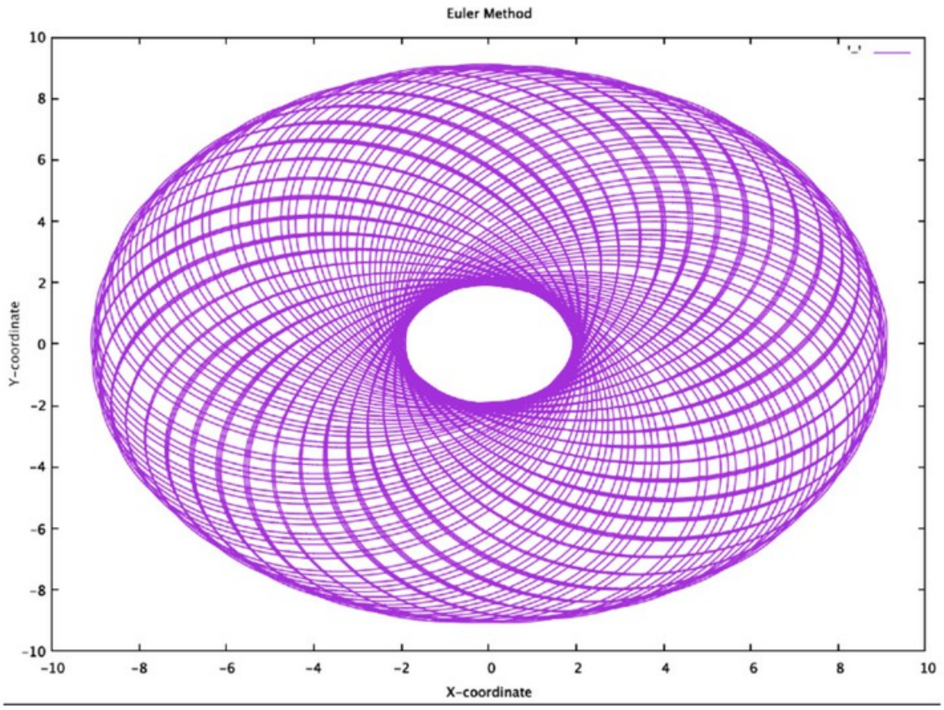
Graphs and Observations:

By taking the Coriolis coefficient to be that of earth, 0.000073, and length
of pendulum to be 1m, we obtain a graph as follows from RK45 method:



So this shows the plane of oscillations changing throughout the day-
although the above graph is only for a few hours. This is a successful simulation of the Foucault pendulum.

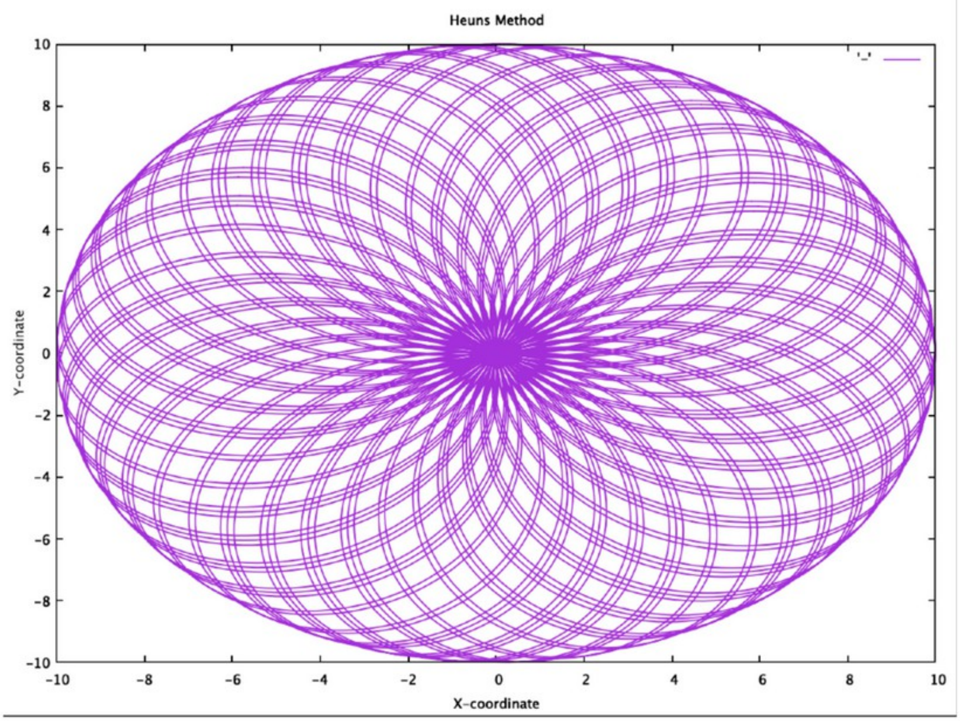
Accuracy Comparisons:
We can make some comparisons between these methods by noticing the
time steps required to obtain a clean graph.
For example,



This graph was obtained by taking value of time step as 0.000411:

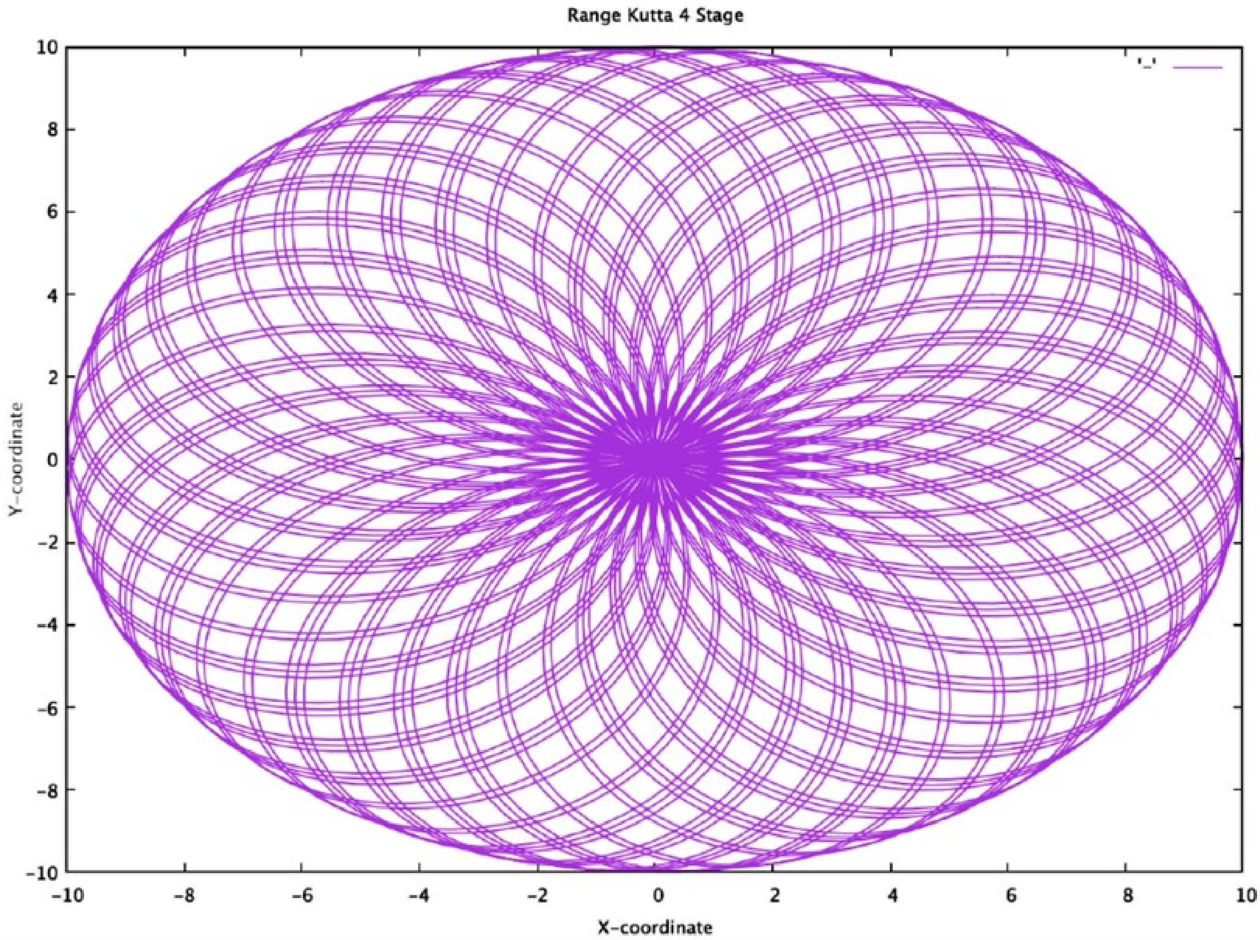
./a.out e 0.1 0.1 1 1 0.000411 6000000

Using Heun’s,



Obtained for a time step of 0.00411, for values:
./a.out h 0.1 0.1 1 1 0.00411 600000
Notice that this graph has more resolution and the centre is also blocked.

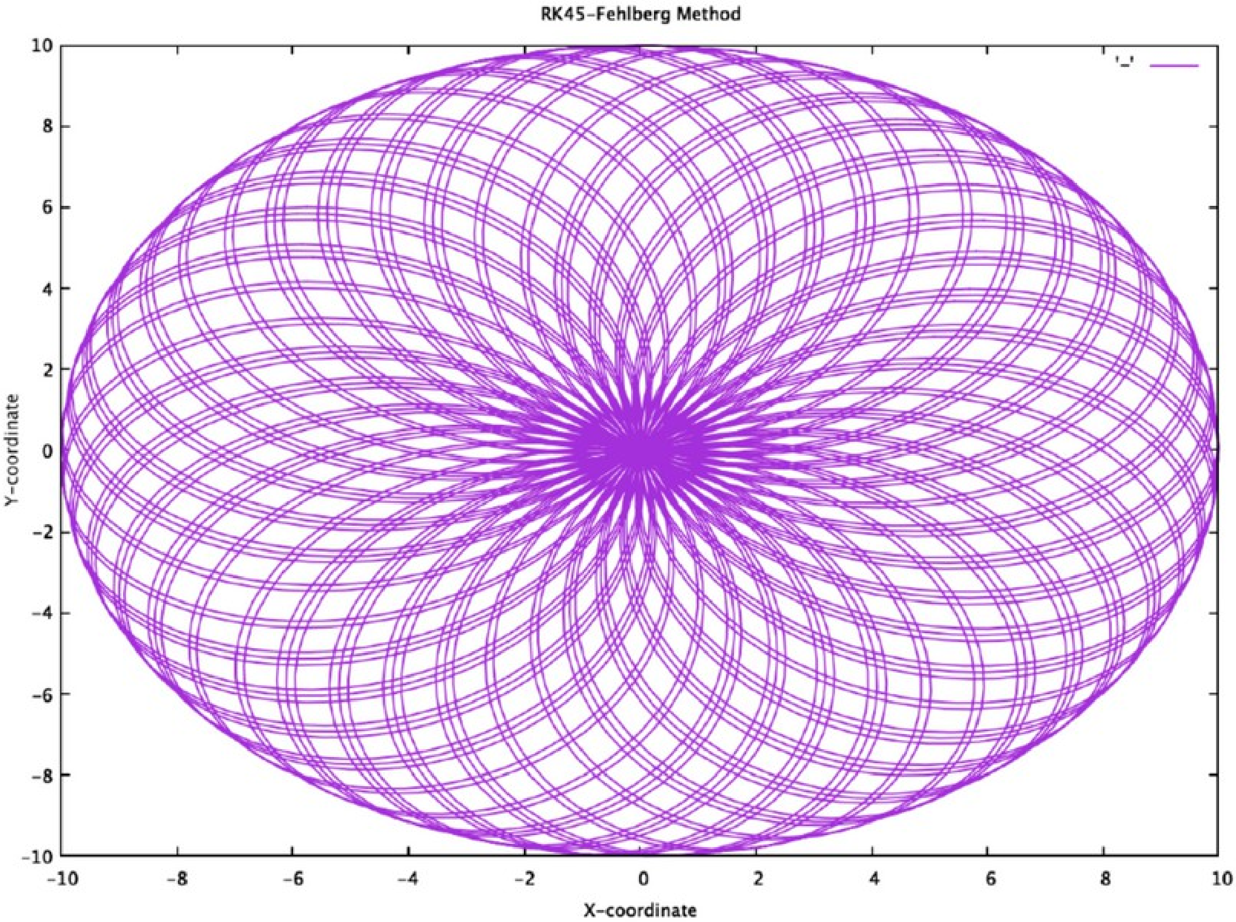
Range Kutta improved on the accuracy even more:



This graph came for a time step of 0.0411, at values:
./a.out r 0.1 0.1 1 1 0.0411 60000

(Here a slight modification to code was made: instead of 1:50 being
printed we printed 1:10)

This same graph was obtained by RK45 method:



Here the step size was largest: 0.411
The values were:
./a.out f 0.1 0.1 1 1 0.411 6000

We see that of all the methods used, RK45 is most capable of handling
large step sizes, while Euler is least capable.

Thank You!