EE1103 - Numerical Methods
QUIZ 2

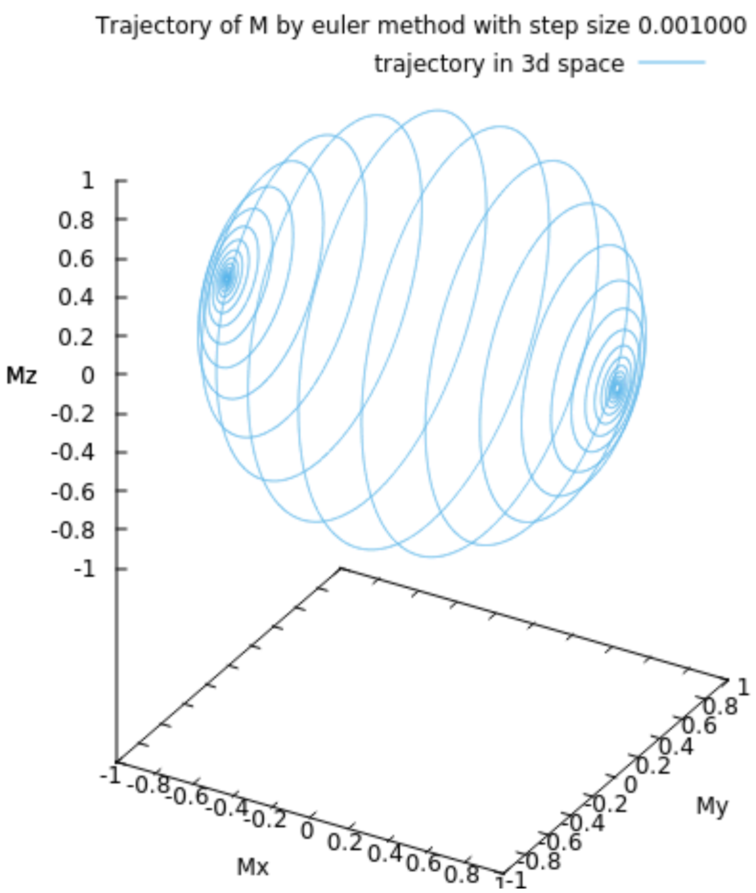
By:
Ch Sarvani (EE18B120)
Abhijeet Ajithkumar (EE18B121)
Akilesh Kannan (EE18B122)

Q1. To plot the trajectories of M for various stepsizes using Euler's method, and to determine the maximum stepsize.

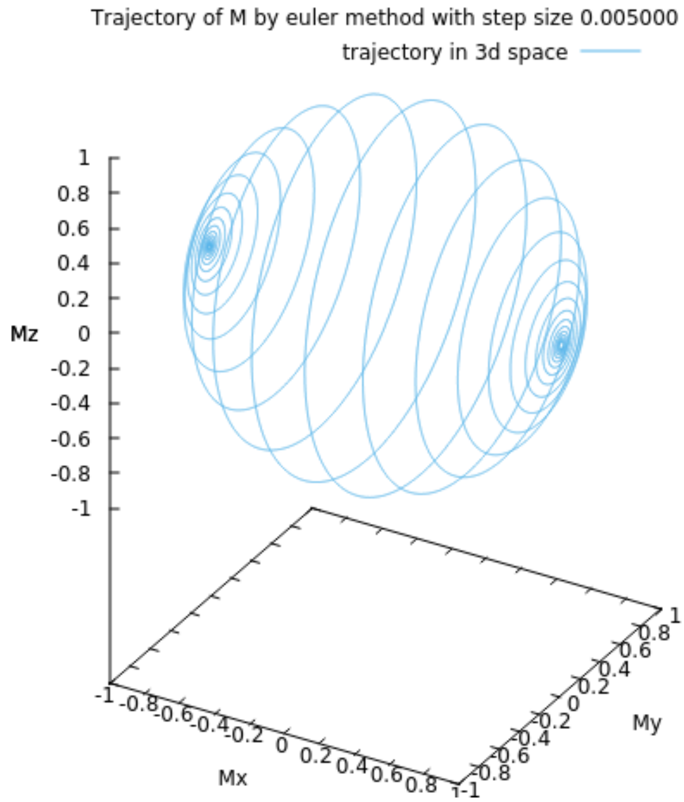
The Euler's or the forward difference method is a simple method of solving ordinary differential equations (O.D.E.). It uses a constant stepsize. The iterative formula is given below:
$y_{n+1} - y_n = hf(x,y)$, where f(x,y) is the derivative of y wrt x.

The plots that we obtained for various time stepsizes are shown below.

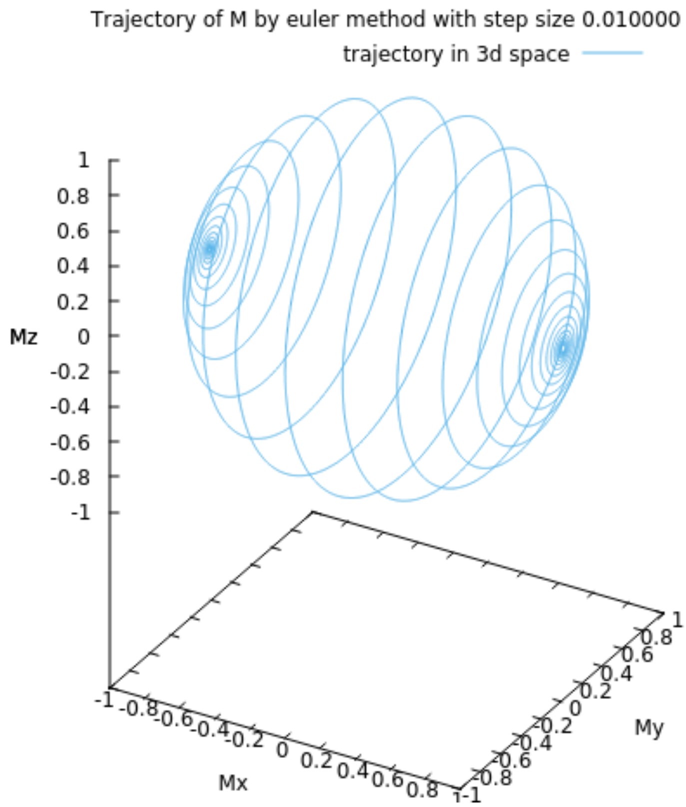
The actual trajectory (stepsize=0.001)



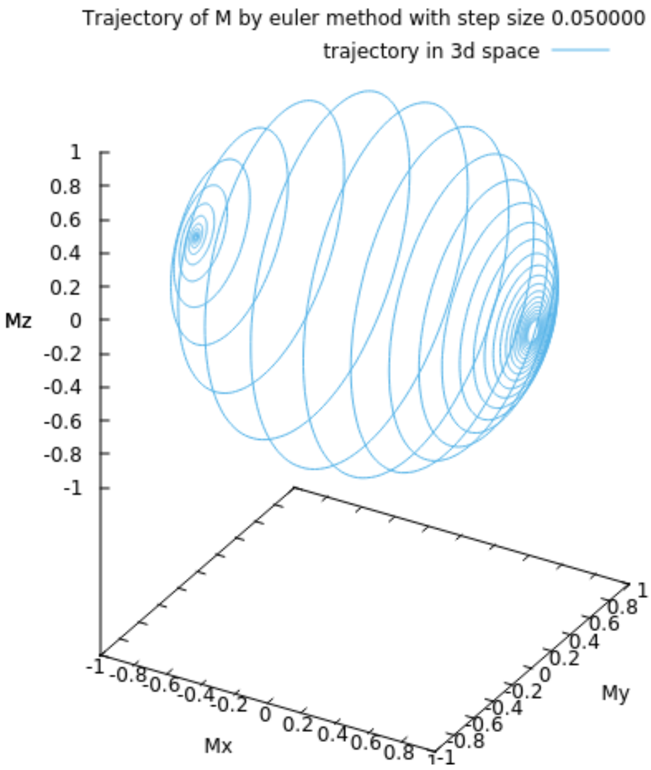
For stepsize 0.005,



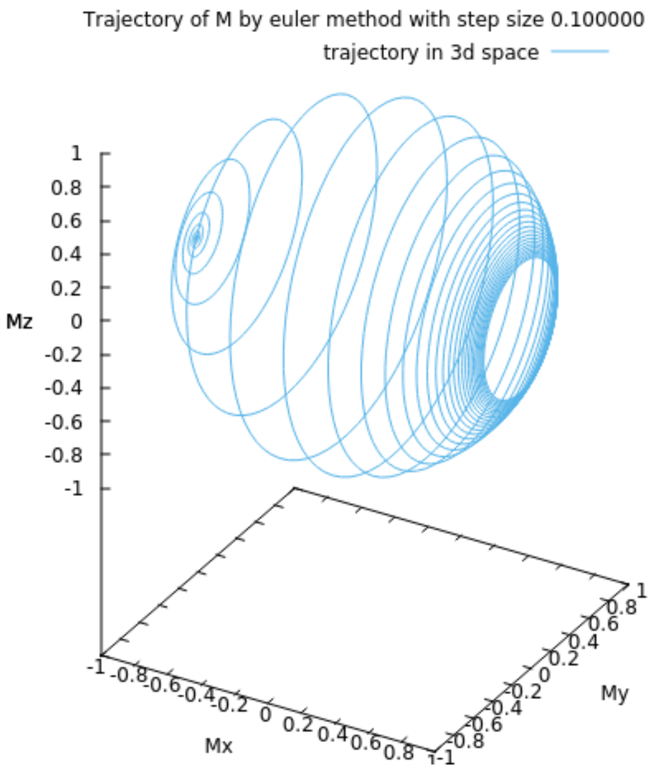
For stepsize=0.01,



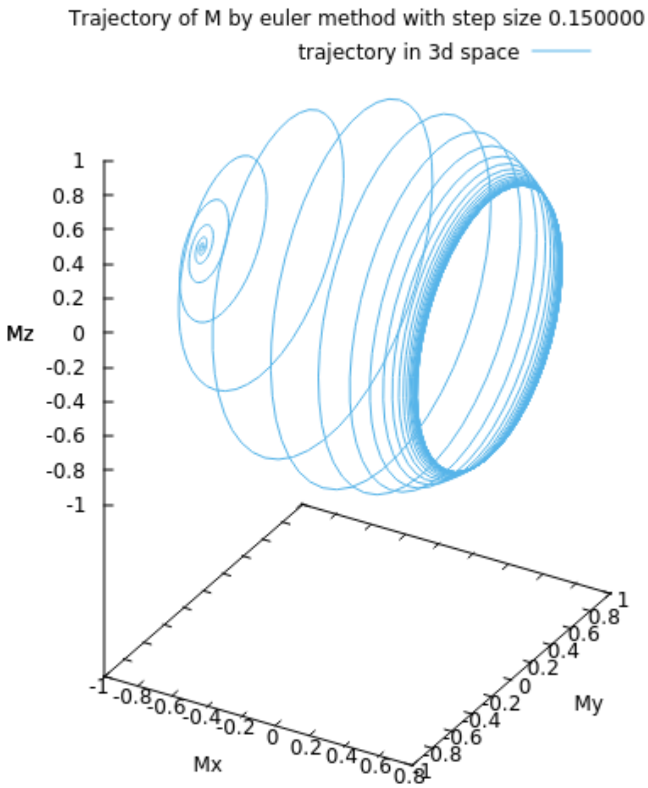
For stepsize=0.05,



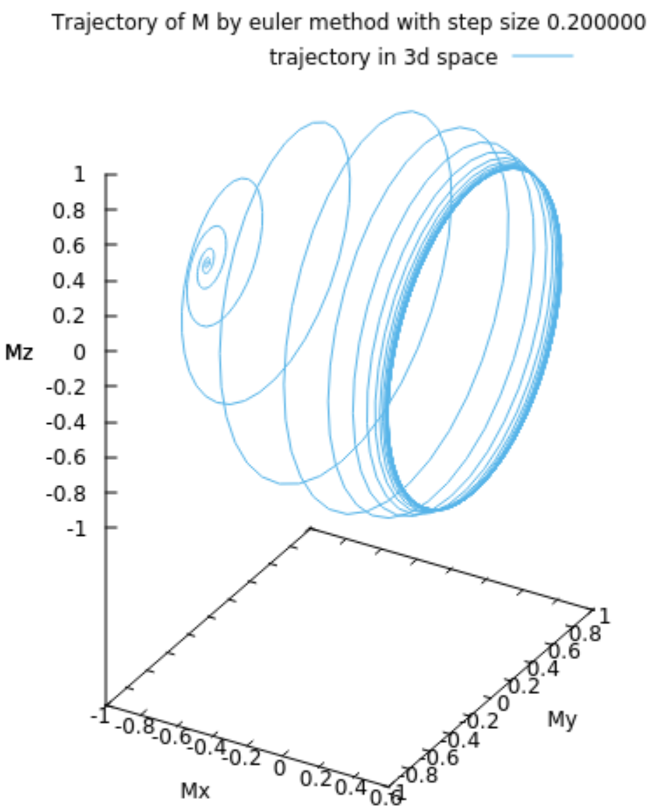
For stepsize=0.1,



For stepsize=0.15,



For stepsize=0.2,



Although the error value for 0.2 is not that significant, the plot begins to distort at around 0.1 itself. The maximum allowable stepsize is between 0.1 to 0.2. The exact value of the maximum stepsize can vary according to considerations of error value and plot distortion.

2. To plot the trajectories of M in 3D space as determined by the fourth order Runge-Kutta method and to hence determine the maximum affordable stepsize.

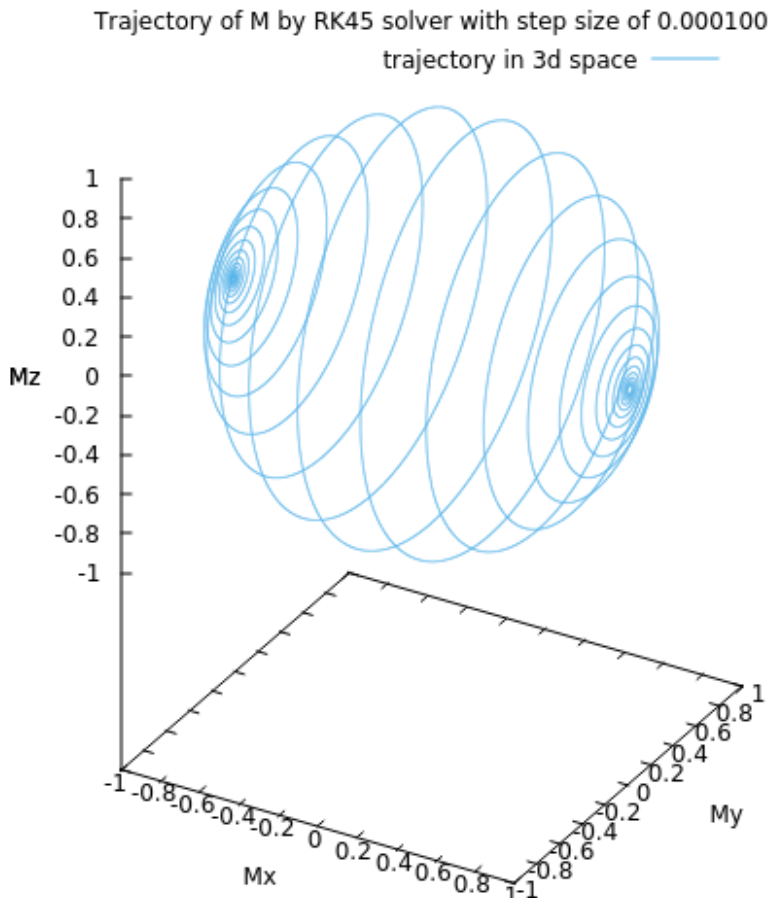
The Runge Kutta solver is an iterative form of the Euler method. The most popular fourth order method has an error of the fifth order.

If we denote $f(x)$ as the derivative of the required function, which we are trying to find, then the Runge-Kutta (RK4) method can be expressed as:

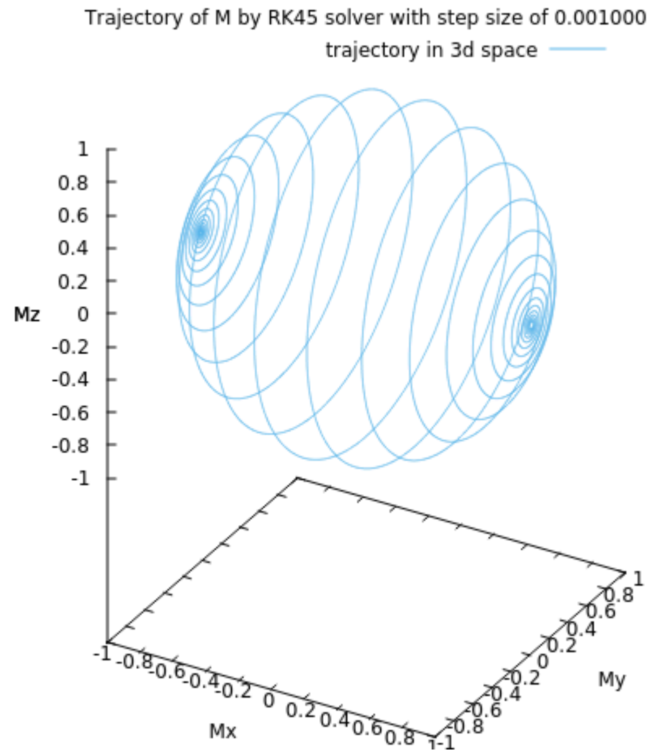
$$k_1 = hf(x_n)$$
$$k_2 = hf(x_n + k_1/2)$$
$$k_3 = hf(x_n + k_2/2)$$
$$k_4 = hf(x_n + k_3)$$

$$y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$$

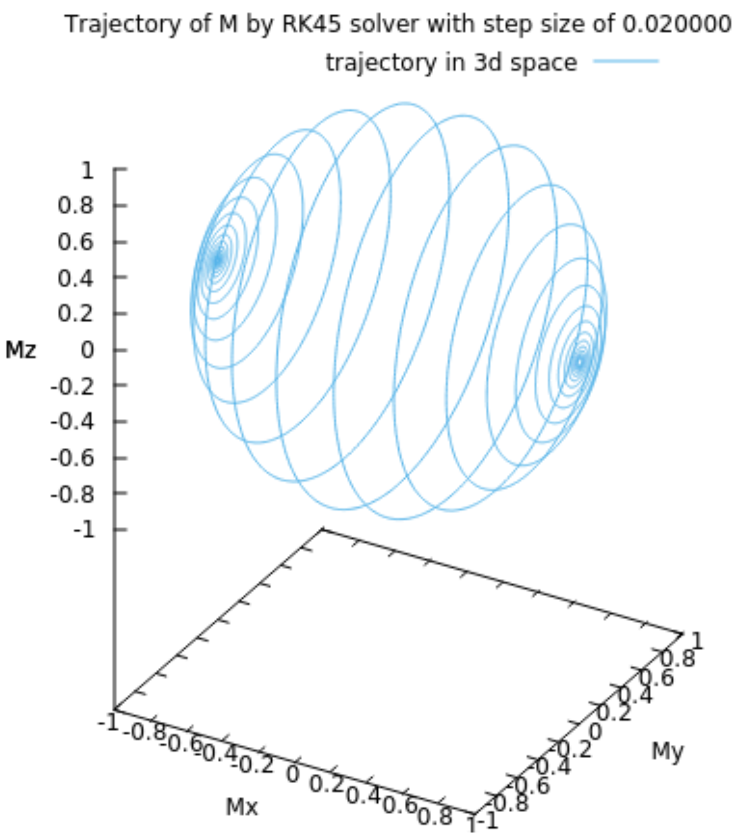
For stepsize=0.0001,



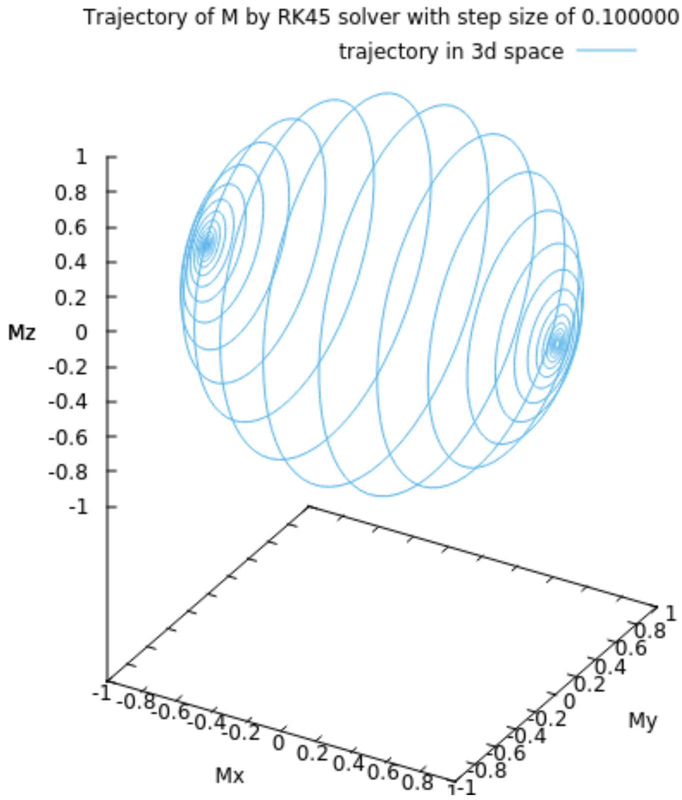
For stepsize=0.001



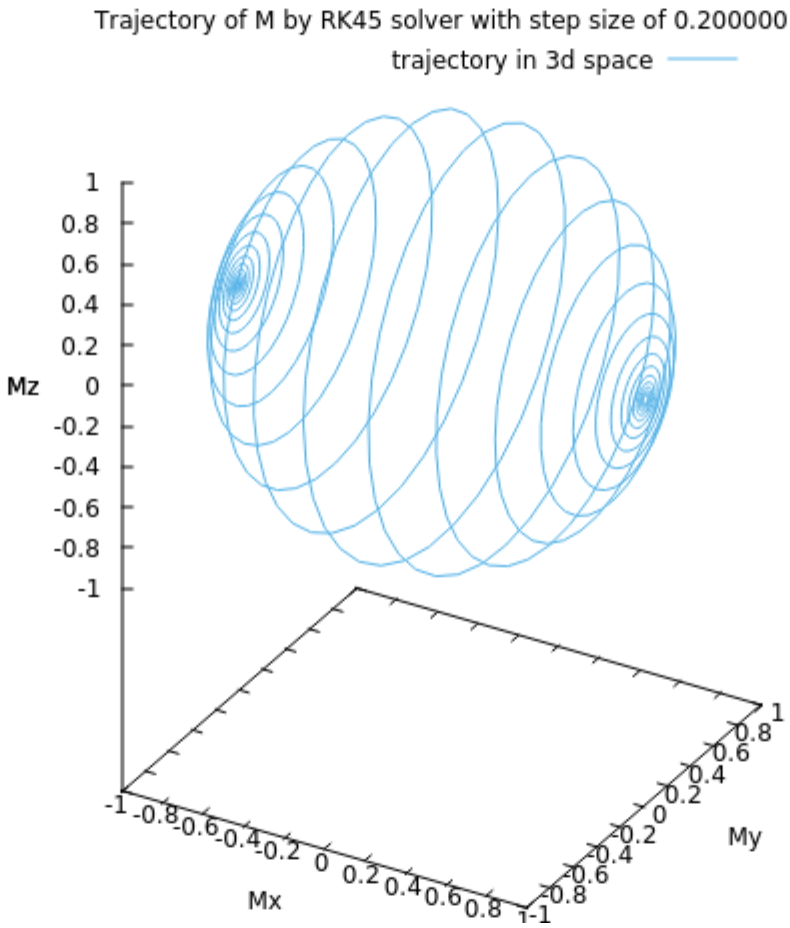
For stepsize=0.02,



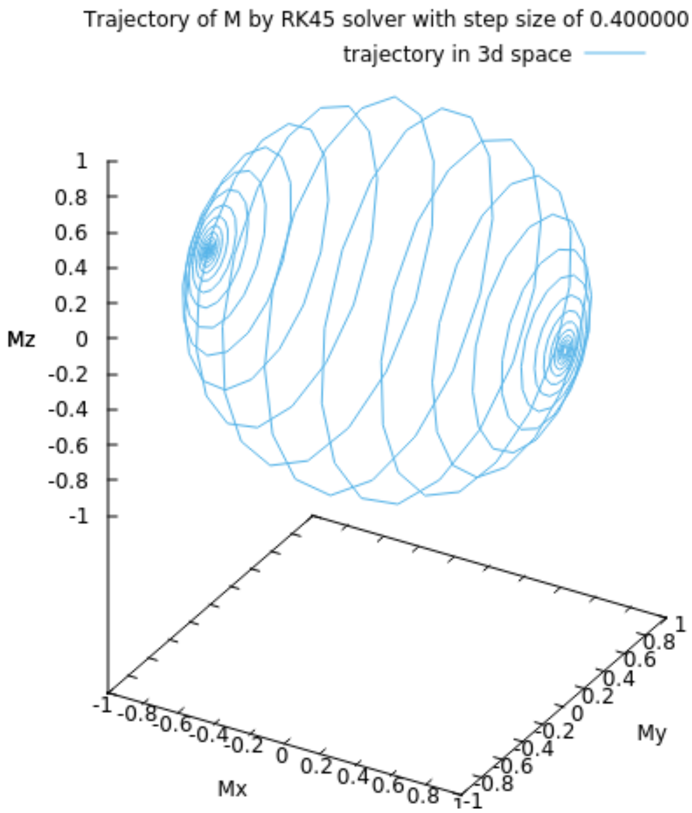
For stepsize=0.1,



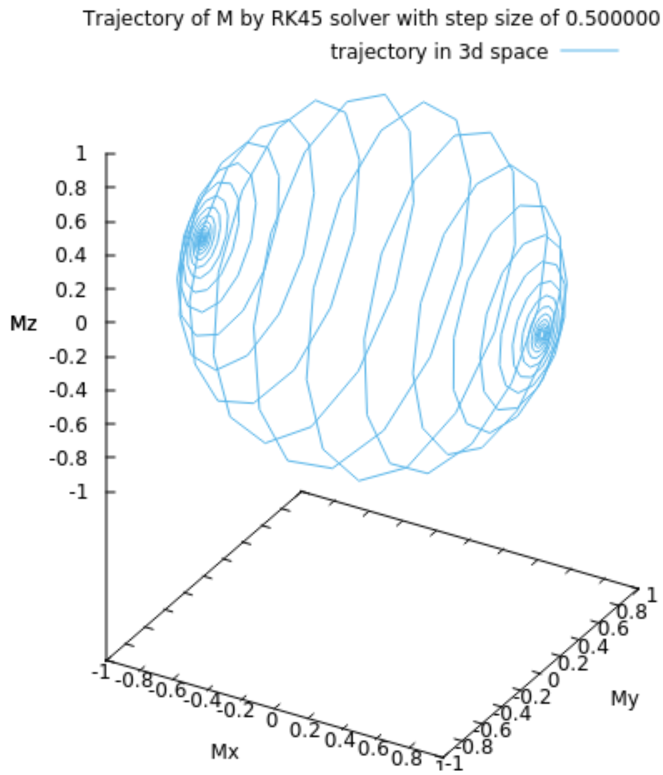
For stepsize=0.2,



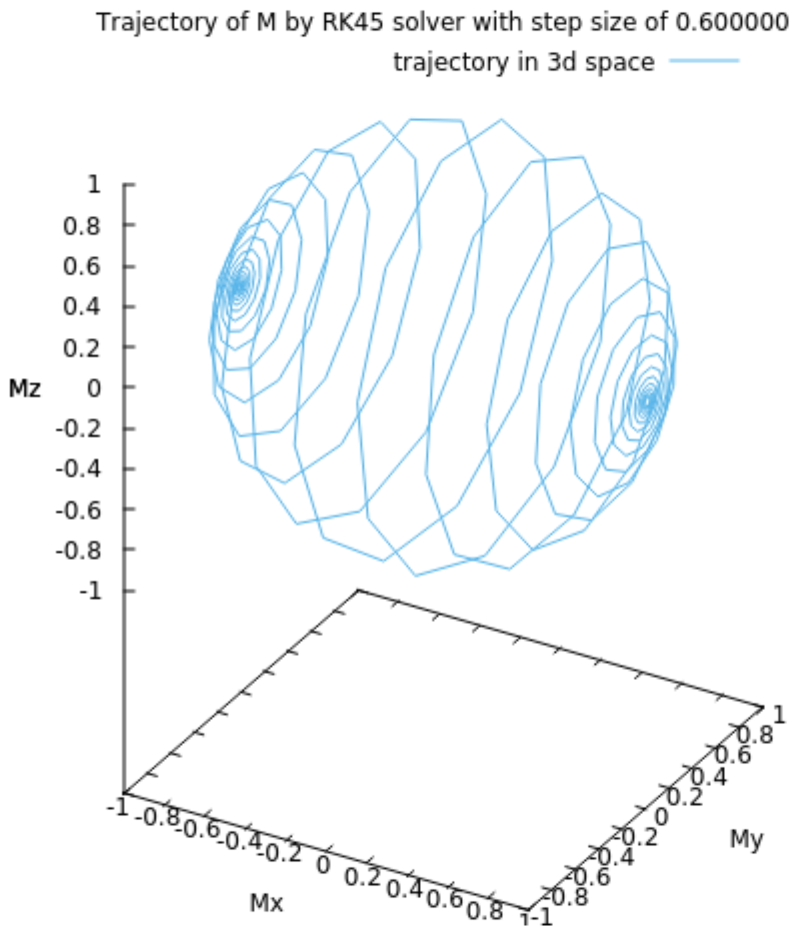
For stepsize=0.4,



For stepsize=0.5,



For stepsize=0.6,



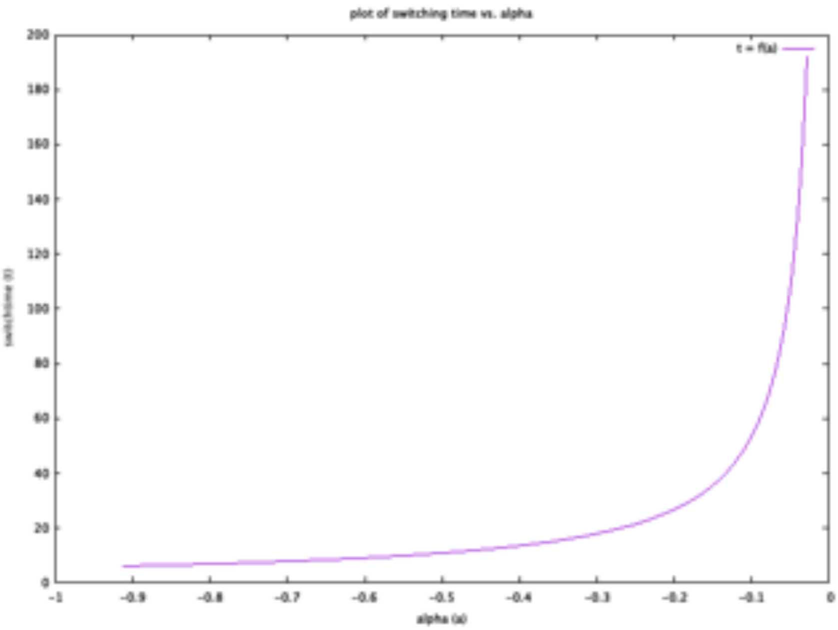
In case of the RK45 solver, the graphs retain their spherical shape (unlike the Euler method). However, from stepsizes 0.4 and above, the spheres become remarkably sharp. Taking that into consideration, the maximum stepsize is around 0.4.

*<u>3. To vary alpha on a log scale from -10^{-4} to -1, and to measure and plot the time taken for the x-component of **M** to switch signs.</u>*

Tabulation of α and switching time

-0.030200	175.7630000000
-0.033113	159.7320000000
-0.036308	145.9550000000
-0.039811	133.1670000000
-0.043652	121.2940000000
-0.047863	110.5170000000
-0.052481	101.1120000000
-0.057544	91.9560000000
-0.063096	83.8760000000
-0.069183	76.6290000000
-0.075858	69.9330000000
-0.083176	63.7680000000
-0.091201	58.0950000000
-0.100000	52.8940000000
-0.109648	48.2830000000
-0.120226	44.1500000000
-0.131826	40.1330000000
-0.144544	36.6910000000
-0.158489	33.4040000000
-0.173780	30.5240000000
-0.190546	27.7600000000
-0.208930	25.4050000000
-0.229087	23.1080000000
-0.251189	21.0680000000
-0.275423	19.2700000000
-0.301995	17.5530000000
-0.331131	15.9710000000
-0.363078	14.5810000000
-0.398107	13.3270000000
-0.436516	12.1590000000
-0.478630	11.0710000000
-0.524807	10.0810000000
-0.575440	9.1910000000
-0.630957	8.3890000000
-0.691831	7.6610000000
-0.758578	6.9950000000
-0.831764	6.3820000000

Plot of switching time vs α

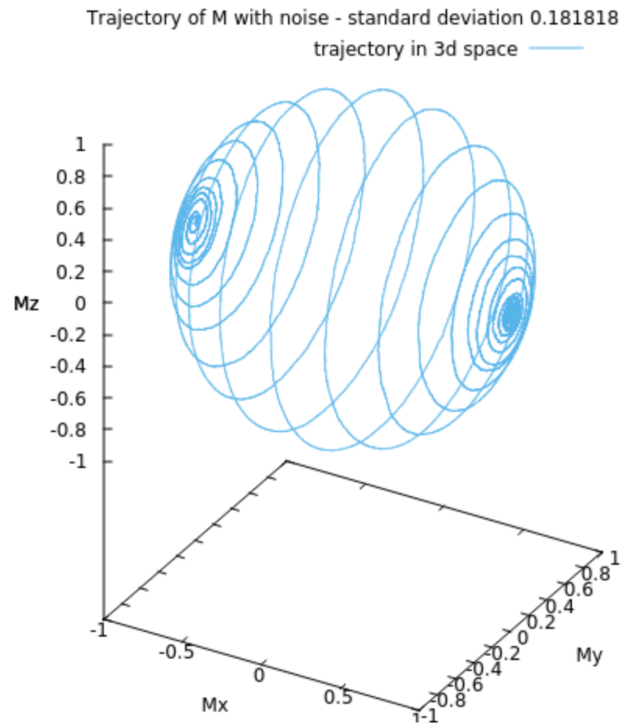


Clearly, the second term of the Landau-Lifshitz equation is the one that is responsible for switching $\mathbf{M_x}$. At α exactly equal to 0, $\mathbf{M_x}$ will take infinite amount of time to change signs. It follows that at values of α close to 0, the switching time is very high. As α increases in magnitude, it takes lesser and lesser time to switch. Since the derivative depends linearly on α, no local maxima or minima exist for this curve. Hence it is hyperbolic.

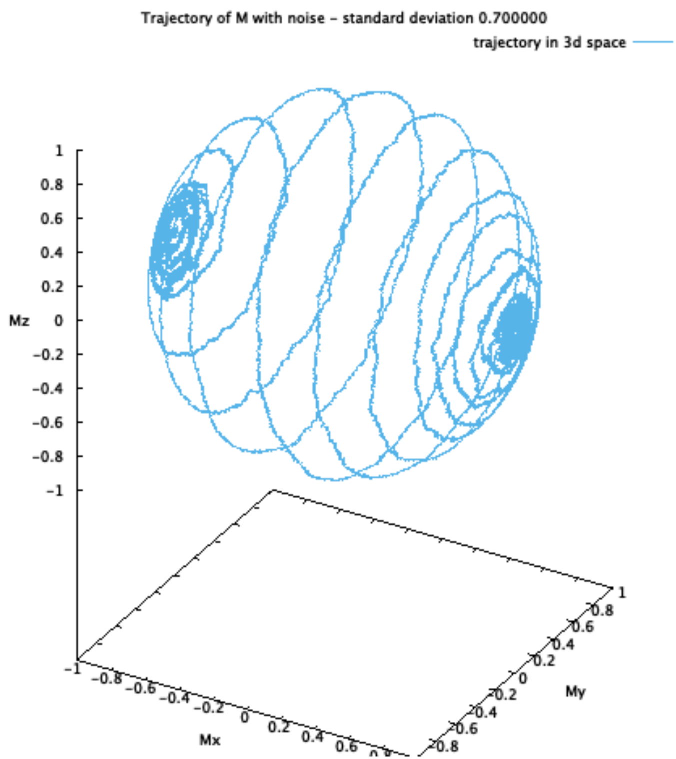
4. To add a random kick in some direction by adding a random numbers to each component of M after every iteration. The random numbers are obtained from a distribution with mean 0 and some standard deviation. The correlation of the simulation between this simulation and the noise-free simulation is also determined and plotted for various values of standard deviation.

Some of the plots that we could generate on varying the standard deviation are shown below:

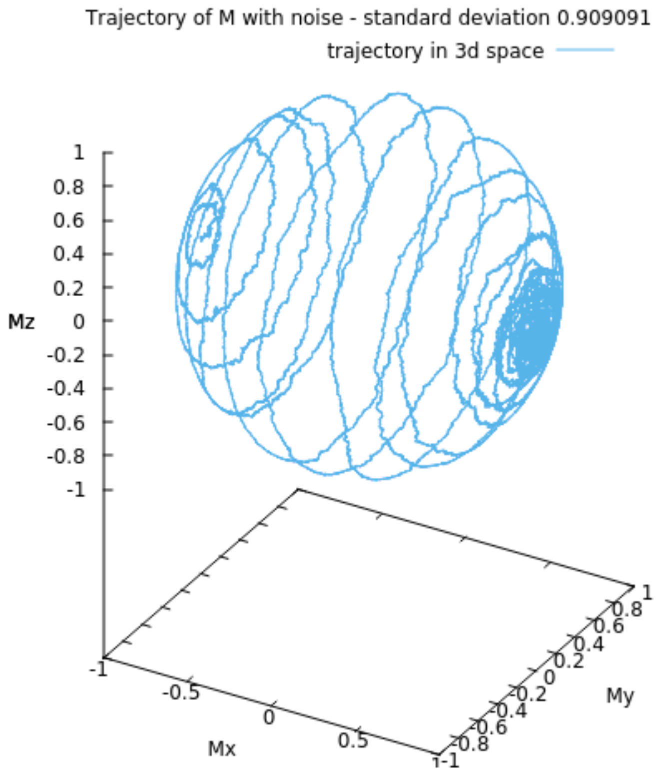
For standard deviation=0.181818,



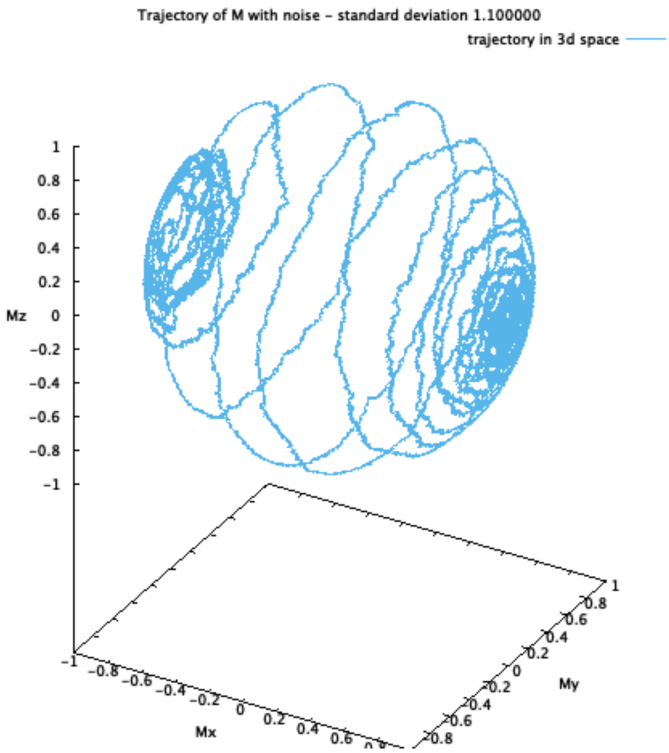
For standard deviation = 0.7,



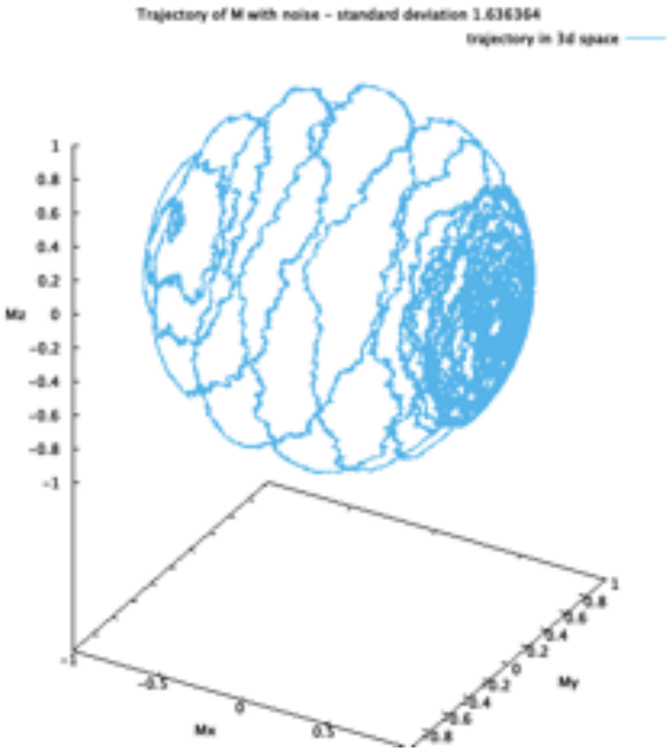
For standard deviation=0.909091



For standard deviation = 1.1,

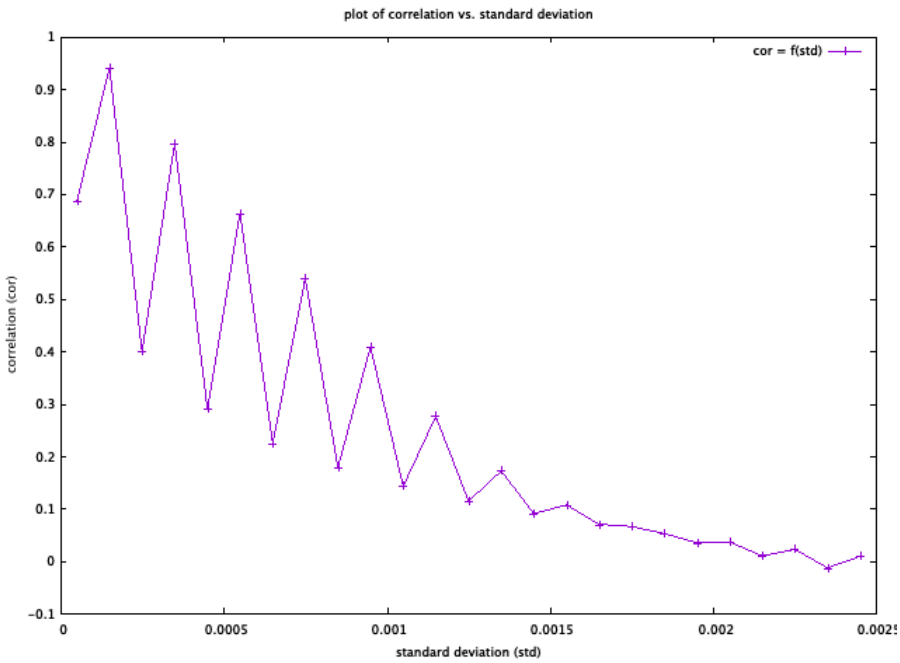


For standard deviation=1.63,



As we can see from the above plots, on increasing the standard deviation, the amount of noise introduced increases and so, the sphere gets more and more distorted. To find out how close the real-life simulation is to the ideal situation, we do a correlation between the two.

In this problem, we have generated a plot of correlation ratio vs. standard deviation of noise introduced to the ideal data. The plot is shown below.



Here, we have defined correlation ratio to be the ratio of cross-correlation of $\mathbf{M}_{noisy}$ and $\mathbf{M}_{ideal}$ to auto-correlation of $\mathbf{M}_{ideal}$.

Cross-correlation of 2 discrete functions, f and g is defined as the sum of product of corresponding values of f and g, i.e., $cross(f, g) = \Sigma(f(x) * g(x))$.

Auto-correlation of a discrete function f is defined as the cross-correlation with itself, i.e., $auto(f) = \Sigma f(x)^2$.

We can see that the correlation ratio goes to zero on increasing standard deviation. This is in accordance with what we expect - the correlation has to be very less on introducing greater amount of noise, as the plots do not resemble the one we obtained initially, without noise.