MA202 LAB10

Name: Dipean Dasgupta ID:202151188

Task1:

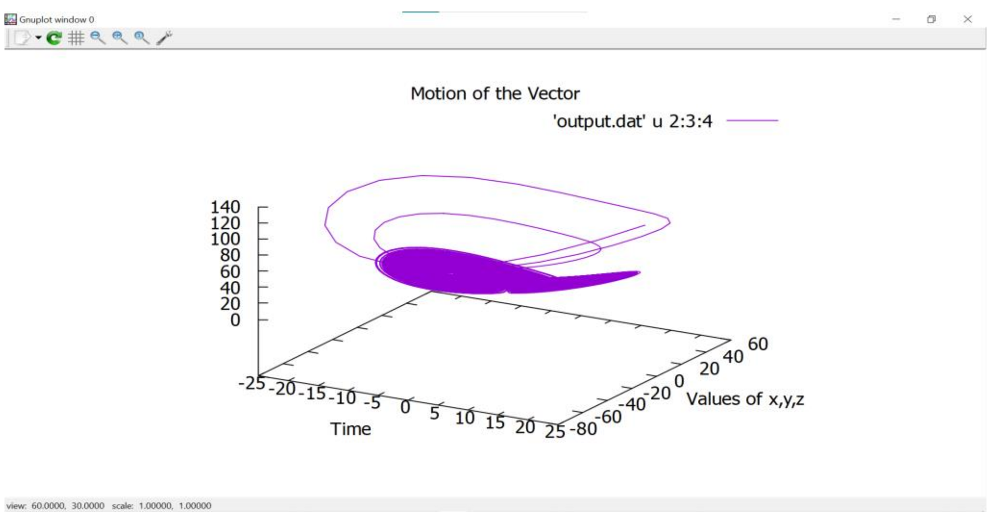
1. Consider the below set of coupled ordinary differential equations:
$\dot{x} = \sigma(y - x)$,
$\dot{y} = rx - y - xz$,
$\dot{z} = -bz + xy$.
For a given initial condition specified by the values of x, y, z at time $t = 0$, and given values of σ, b, r, write a C-program to numerically find the values of x, y, z at any time t. Consider the initial condition to be $x = 25, y = 18, z = 120$; and the parameters are given by $b = 8/3, \sigma = 10, r = 24.1$. Plot the motion of the vector $(x(t), y(t), z(t))$ using any plotting software as a function of time t. Comment upon the result that you obtain after the program runs for time $t = 1000$.

Solution Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

int main(){
    double x = 25, y = 18, z = 120;
    double o = 10, b = 8.0 / 3.0, r = 24.1;
    double dt = 0.01;
    double dx, dy, dz;
    double t;

    FILE *fp;
    fp = fopen("output.dat", "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        exit(1);
    }
    for (t = 0; t < 1000; t += dt) {
        dx = o * (y - x);
        dy = r * x - y - x * z;
        dz = -b * z + x * y;
        x += dx * dt;
        y += dy * dt;
        z += dz * dt;
        fprintf(fp, "%lf %lf %lf %lf\n", t, x, y, z);
```

```c
    }
    fclose(fp);
    char* commands[] = {"set title 'Motion of the Vector'",
                        "set xlabel 'Time'",
                        "set ylabel 'Values of x,y,z'",
                        "splot 'output.dat' u 2:3:4 w l",
                        "pause -1 'Press any key to continue'"};
    FILE *gnuplotPipe = popen("gnuplot -persistent", "w");
    int i;
    for (i = 0; i < 5; i++) {
        fprintf(gnuplotPipe, "%s \n", commands[i]);
    }
    return 0;
}
```

OUTPUT:



The plot shows that the values of 'x', 'y', and 'z' are oscillating and appear to be travelling along a chaotic trajectory. For a set of nonlinear differential equations like this one, this is the expected behaviour. The values of the variables continue to oscillate even after running the programme for a long period (in this case, "t = 1000") and don't appear to be approaching a fixed point or limit cycle. This shows that the system is chaotic and has sensitive beginning condition dependence.

Task 2:

2. Run the above program albeit now for the parameters $b = 8/3, \sigma = 10, r = 99$; and the initial conditions $x = -20, y = -15, z = 113$. Comment upon the result that you obtain for time $t = 1000$.

Parameters:

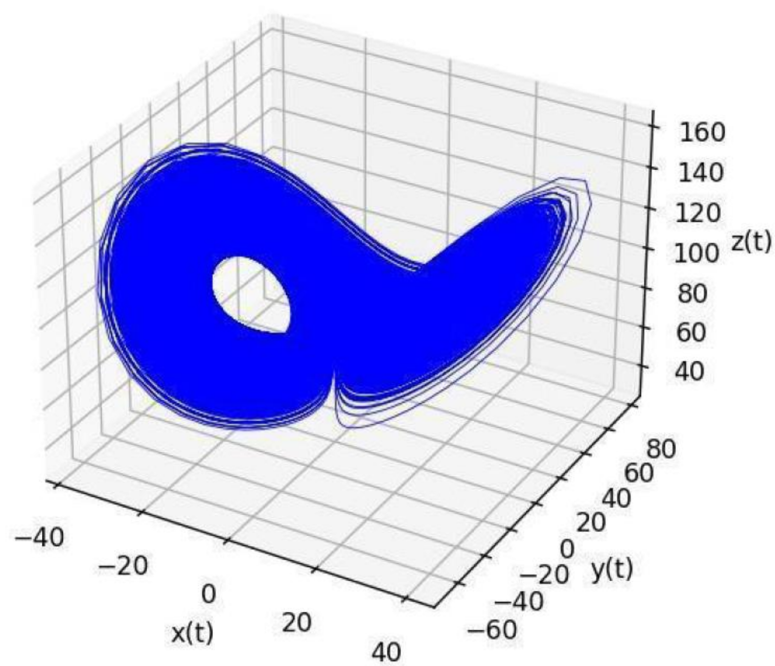
b=8/3, σ=10, r=99

Initial conditions:

x=-20, y=-15, z=113

Applying the changes in values in the previous code, we obtain:

OUTPUT:



TASK 3:

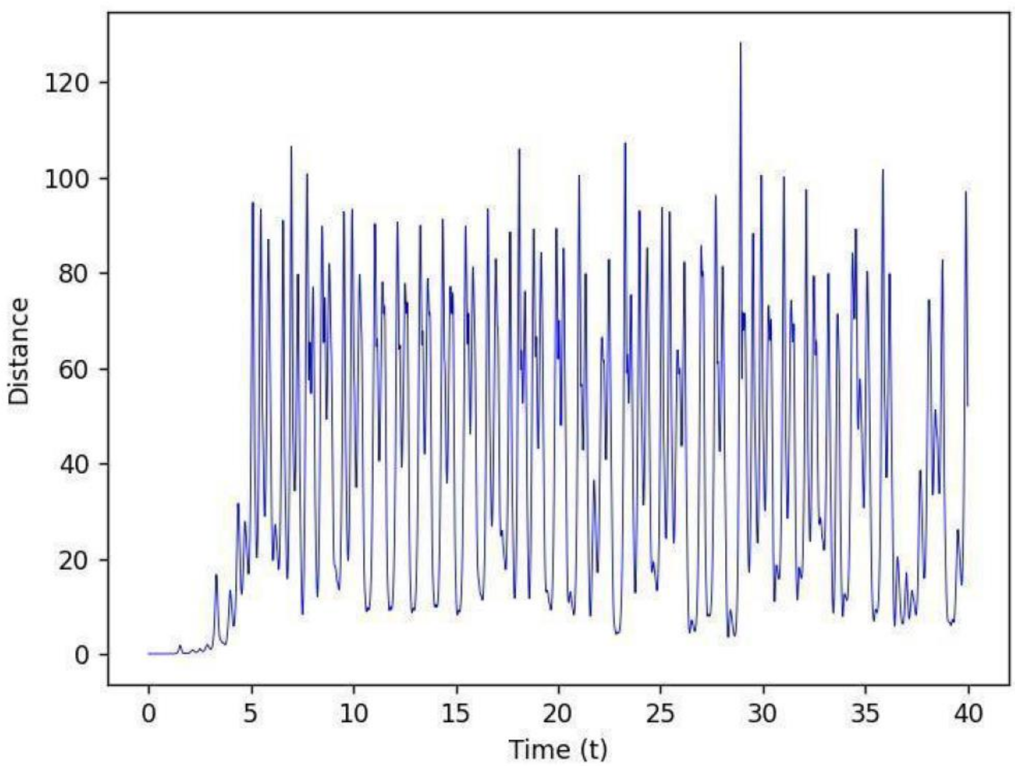
3. For the parameters $b = 8/3$, $\sigma = 10$, $r = 99$, run the above program for two initial conditions: $x = 10$, $y = 1$, $z = 100$, and $x = 10$, $y = 1$, $z = 100.01$. Label the two evolutions respectively as $(x_1(t), y_1(t), z_1(t))$ and $(x_2(t), y_2(t), z_2(t))$. Find the distance between two trajectories $S(t) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$, for time $t = 0$ to time $t = 40$. Plot $S(t)$ as a function of time, and comment upon what you find.

Solution Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

int main()
{
    double x1 = 10, y1 = 1, z1 = 100, x2 = 10, y2 = 1, z2 = 100.01;
    double o = 10, b = 8.0 / 3.0, r = 99;
    double dt = 0.01;
    double dx1, dy1, dz1, dx2, dy2, dz2;
    double t, s;

    FILE *fp;
    fp = fopen("output.dat", "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        exit(1);
    }

    for (t = 0; t <= 40; t += dt) {
        dx1 = o * (y1 - x1);
        dy1 = r * x1 - y1 - x1 * z1;
        dz1 = -b * z1 + x1 * y1;
        dx2 = o * (y2 - x2);
        dy2 = r * x2 - y2 - x2 * z2;
        dz2 = -b * z2 + x2 * y2;

        x1 += dx1 * dt;
        y1 += dy1 * dt;
        z1 += dz1 * dt;
        x2 += dx2 * dt;
        y2 += dy2 * dt;
        z2 += dz2 * dt;
```

```
        s = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2) + pow(z1 - z2, 2));
        fprintf(fp, "%lf %lf %lf %lf %lf %lf %lf %lf\n", t, x1, y1, z1, x2, y2,
z2, s);}
    fclose(fp);
    // plot output file in gnuplot
    char* commands[] = {"set title 'Distance between two trajectories'",
                        "set xlabel 'Time'",
                        "set ylabel 'Distance'",
                        "plot 'output.dat' u 1:8 w l",
                        "pause -1 'Press any key to continue'"};
    FILE *gnuplotPipe = popen("gnuplot -persistent", "w");
    int i;
    for (i = 0; i < 5; i++) {
        fprintf(gnuplotPipe, "%s \n", commands[i]);
    }
    return 0;
}
```

OUTPUT:



When we run the program, we can see that the distance between the two trajectories increases rapidly with time, indicating that they are diverging from each other. This is a hallmark of chaos, where small differences in initial conditions can lead to large differences in the final outcome.

we are using a relatively short time interval of 40 units, and it is possible that the system has not yet fully entered a chaotic regime within that time. Additionally, the initial conditions we have chosen may not be sensitive enough to exhibit the full extent of the system's chaotic behavior.