**MATH10232: COURSEWORK ASSIGNMENT**

The Lorenz equations are the three, coupled ordinary differential equations:

where ρ, σ and β are all real constants greater than or equal to zero.

**1) Find the fixed points of the system. What is the difference between the two cases ρ < 1 and ρ > 1?**

Fixed points appear when the differential equations are all equal to 0.

1st Case:

Fixed Point 1 = (0,0,0)

2nd Case:

🡪

🡪 🡪

🡪 🡪

Fixed Point 2 = , ,

When ρ < 1, the system will have only one real fixed point (at the origin) because the constant is positive, but the bracket is negative, resulting in: .

On the other hand, ρ > 1 will have three fixed points due to

**2) Create a mylorenz program**

The code below is from the mylorenz.m function which I made.

%Matlab Coursework -- Lorenz Function

function ydot = mylorenz(rho, y)

ydot(1) = 10\*(y(2) - y(1)); %(1a) with sigma = 10

ydot(2) = (rho\*y(1)) - y(2) - (y(1)\*y(3)); %(1b) with rho as an argument

ydot(3) = (y(1)\*y(2)) - (8/3\*y(3)); %(1c) with beta = 8/3

end

**3) Use Euler’s method to integrate the Lorenz equations**

function euler(rho)

%rho is an argument

y(1) = 0; %y1 initial value

y(2) = 1; %y2 initial value

y(3) = 0; %y3 initial value

h = 0.001; %h is the stepsize

t = [0:h:30];

for n=1:30000

%makes the n+1th row and all its columns = to the values on the right

y(n+1,:) = y(n,:) + h\*mylorenz(rho,y(n,:)); %uses mylorenz function

end;

y(30000,:) %outputs the last row's (30000) columns.

The code below is from the euler.m function which I made

Case A:

>> euler(0)

ans =

1.0e-13 \*

0.8639 0.7775 0.0000

Case B:

>> euler(15)

ans =

-6.1104 -6.1108 13.9995

Case C:

>> euler(30)

ans =

10.4435 6.5621 35.1653

**4) Time trace graphs**

**Code:**

function euler(rho)

%rho is an argument

y(1) = 0; %y1 initial value

y(2) = 1; %y2 initial value

y(3) = 0; %y3 initial value

h = 0.001; %h is the stepsize

t = [0:h:30];

for n=1:30000

%makes the n+1th row and all its columns = to the values on the right

y(n+1,:) = y(n,:) + h\*mylorenz(rho,y(n,:)); %uses mylorenz function

end;

y(30000,:) %outputs the last row's (30000) columns.

plot(t,y(:,1));

hold on

plot(t,y(:,2));

plot(t,y(:,3));

xlabel('Time (t)');

ylabel('Lorenz Equations');

Case A: 🡪 See graph 4a)

As t 🡪 , The system approaches the fixed point (0,0,0) as can be seen by the graph.

Case B: 🡪 See graph 4b)

As t 🡪 , The system approaches the fixed point:

, , , , 14

Case C: 🡪 See graph 4c)

As t 🡪 , the system does, at first, appear to approach a fixed point. However, the graph has become very chaotic and looks like it may reach , +, 29 , but it does not reach that point.

**5) Time trace graphs**

**Code:**

function euler(rho)

%rho is an argument

y(1) = 0; %y1 initial value

y(2) = 1; %y2 initial value

y(3) = 0; %y3 initial value

h = 0.001; %h is the stepsize

t = [0:h:30];

for n=1:30000

%makes the n+1th row and all its columns = to the values on the right

y(n+1,:) = y(n,:) + h\*mylorenz(rho,y(n,:)); %uses mylorenz function

end;

y(30000,:) %outputs the last row's (30000) columns.

plot3(y(:,1),y(:,2),y(:,3));

xlabel('y1');

ylabel('y2');

zlabel('y3');

Case A: 🡪 See graph 5a)

Motion: The graph appears to move in a steady and stable motion following a curved path and forming a sing loop as it approaches one of the fixed points.

Case B: 🡪 See graph 5b)

Motion: The graph, at first makes a lasso like motion and appears to maintain a steady spiral down to the fixed point , , 14

Case C: 🡪 See graph 5c)

Motion: The graph now follows a more chaotic motion after it seemingly converges to a fixed point. However, it does not reach it and it is repelled back (diverges) to form a similar disc like shape with the same spiral motion.