3.1 Lorenz model

a)

```
(*Createing system and evaluate the jacobian matrix(1)*)
system = {σ * (y - x), r * x - y - x * z, x * y - b * z};
dots = {x, y, z};
jacobian[x_, y_, z_, σ_, b_, r_] = Grad[system, dots]
```

Out[∘]= {{-σ, σ, 0}, {r - z, -1, -x}, {y, x, -b}}

```
(*putting in the values, x, y and z should be zero since
 we want the fixed points*)
sol = jacobian[0, 0, 0, 10, 8 / 3, 28]
```

Out[∘]= $\left\{\{-10, 10, 0\}, \{28, -1, 0\}, \left\{0, 0, -\frac{8}{3}\right\}\right\}$

In[∘]:=
```
(*We see above that there exist three fixed points
 Now its time to investigate how many that are stable*)
eigenVals = Eigenvalues[sol];
N[eigenVals]
```

Out[∘]= {-22.8277, 11.8277, -2.66667}

```
(* We can se above that there is two negative eigenvalues and one positive,
so we have a unstable saddle point *)
```

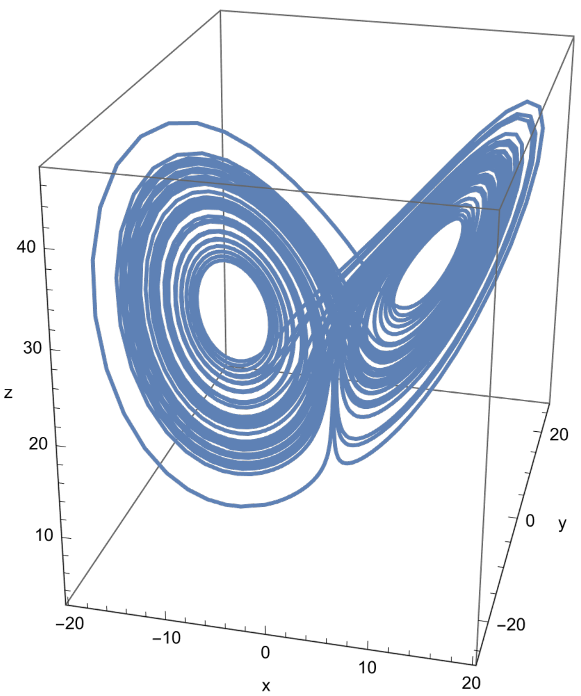
b)

```
(*Setting the values for the constants and start value*)
bVal = 8 / 3;
rVal = 28;
σVal = 10;
startVal = 0.01;
(*Setting up the equations and use NDSolve to solve it numerically*)
eqns = {x'[t] == σVal * (y[t] - x[t]), y'[t] == rVal * x[t] - y[t] - x[t] * z[t],
   z'[t] == x[t] * y[t] - bVal * z[t], x[0] == startVal, y[0] == startVal, z[0] == startVal};
solution = NDSolve[eqns, {x, y, z}, {t, 0, 2000}]
```

Out[ ]=
```
{{x → InterpolatingFunction[ Domain: {{0., 2.00 × 10^3}}  Output: scalar
                             Data not in notebook. Store now ],
  y → InterpolatingFunction[ Domain: {{0., 2.00 × 10^3}}  Output: scalar
                             Data not in notebook. Store now ],
  z → InterpolatingFunction[ Domain: {{0., 2.00 × 10^3}}  Output: scalar
                             Data not in notebook. Store now ]}}
```

```
(*Taking out the solutions for x, y and z and plot it*)
ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. solution],
 {t, 100, 150}, AxesLabel → {"x", "y", "z"}]
```

Out[◦]=



c)

In[2]:=
```
(*Createing system and evaluate the jacobian matrix(1)*)
system = {σ * (y - x), r * x - y - x * z, x * y - b * z};
dots = {x, y, z};
jacobian[x_, y_, z_, σ_, b_, r_] = Grad[system, dots] // MatrixForm
```

Out[4]//MatrixForm=

$$\begin{pmatrix} -\sigma & \sigma & 0 \\ r - z & -1 & -x \\ y & x & -b \end{pmatrix}$$

d)

```
(*Creating the jacobian again and then take the trace of it*)
jacobian[x_, y_, z_, σ_, b_, r_] = Grad[system, dots]
```

In[13]:=
```
{{-σ, σ, 0}, {r - z, -1, -x}, {y, x, -b}}
Tr[jacobian[x, y, z, σ, b, r]]
```

Out[13]= {{-σ, σ, 0}, {r - z, -1, -x}, {y, x, -b}}

Out[14]= $-1 - b - \sigma$