

```

In[3141]:= Remove[x, y, z, sigma, r, b, s, t]
r = 340;
b = 5;
sigma = 16;
s = NDSolve[{x'[t] == sigma * (y[t] - x[t]), y'[t] == r * x[t] - y[t] - x[t] * z[t],
z'[t] == x[t] * y[t] - b * z[t], z[0] == x[0] == y[0] == 0.1},
{x, y, z}, {t, 0, 1000}, MaxStepSize -> 0.0001,
Method -> {"TimeIntegration" -> {"ExplicitRungeKutta", "DifferenceOrder" -> 8}}];

x'[t] = sigma * (y - x);
y'[t] = r * x - y - x * z;
z'[t] = x * y - b * z;

J = {{D[x'[t], x], D[x'[t], y], D[x'[t], z]},
{D[y'[t], x], D[y'[t], y], D[y'[t], z]}, {D[z'[t], x], D[z'[t], y], D[z'[t], z]}};
lambda = {0, 0, 0};
dt = 0.0001;
Q = IdentityMatrix[3]

```

```

Out[3152]= {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}

```

```
In[3171]:= nIterations = 50000;
list1 = {};
list2 = {};
list3 = {};
time = {};
For[timeStep = 1, timeStep < nIterations, timeStep++,
  t = timeStep * dt;
  x = x[t] /. s[[1]];
  y = y[t] /. s[[1]];
  z = z[t] /. s[[1]];
  M = IdentityMatrix[3] + J * dt;
  {Q, R} = QRDecomposition[Dot[M, Transpose[Q]]];

  AppendTo[list1, lambda[[1]] / (nIterations * dt)];
  AppendTo[list2, lambda[[2]] / (nIterations * dt)];
  AppendTo[list3, lambda[[3]] / (nIterations * dt)];
  AppendTo[time, t];

  For[i = 1, i < 4, i++, lambda[[i]] += Log[Abs[R[[i]][[i]]]]]

]
lambda = lambda / (nIterations * dt)
Total[lambda]
Show[ListLogLinearPlot[{time, list1}], ListLogLinearPlot[{time, list2}],
  ListLogLinearPlot[{time, list3}], PlotRange -> All]
```

Out[3172]= {}

Out[3173]= {}

Out[3174]= {}

Out[3175]= {}

Out[3177]= {16.7601, -18.3441, -29.1692}

Out[3178]= -30.7533

