Chapter 7: Solutions to Exercises

```
In[1]:= Needs["Graphics`MultipleListPlot`"]
```

à Question 1

```
In[2]:= Clear[δ]
```

Let

$$\delta = \frac{\lambda_1 - \lambda_0}{\lambda_2 - \lambda_1}$$

then we can solve for λ_2

In[3]:= $\texttt{sold = Solve}\left[\delta == \frac{\lambda_1 - \lambda_0}{\lambda_2 - \lambda_1}, \lambda_2\right]$

Out[3]= $\left\{\left\{\lambda_2 \to \frac{-\lambda_0 + \lambda_1 + \delta\,\lambda_1}{\delta}\right\}\right\}$

In[4]:= $\texttt{sold /. }\{\lambda_0 \to 3, \lambda_1 \to 3.4495, \delta \to 4.669\}$

Out[4]= $\{\{\lambda_2 \to 3.54577\}\}$

This approximation compares favourably with the computer programme which provides a value of 3.54409.

à Question 2

We know from the previous question that

$$\lambda_2 = \frac{\lambda_1 - \lambda_0}{\delta} + \lambda_1$$

then λ_3 can be obtained by eliminating λ_2 in the expression:

$$\lambda_3 = \frac{\lambda_2 - \lambda_1}{\delta} + \lambda_2$$

Thus

In[5]:= $\texttt{Eliminate}\left[\left\{\lambda_3 == \frac{\lambda_2 - \lambda_1}{\delta} + \lambda_2, \lambda_2 == \frac{\lambda_1 - \lambda_0}{\delta} + \lambda_1\right\}, \lambda_2\right]$

Out[5]= $\left(1 + \frac{1}{\delta}\right)\lambda_0 == \lambda_1 + \frac{\lambda_1}{\delta} + \delta\,\lambda_1 - \delta\,\lambda_3\ \&\&\ (1+\delta)\,\lambda_0 == \lambda_1 + \delta\,\lambda_1 + \delta^2\,\lambda_1 - \delta^2\,\lambda_3\ \&\&\ \delta \neq 0$

In[6]:= $\texttt{Solve}\left[\left(1 + \frac{1}{\delta}\right)\lambda_0 == \lambda_1 + \frac{\lambda_1}{\delta} + \delta\,\lambda_1 - \delta\,\lambda_3, \lambda_3\right]$

Out[6]= $\left\{\left\{\lambda_3 \to -\frac{\lambda_0 + \delta\,\lambda_0 - \lambda_1 - \delta\,\lambda_1 - \delta^2\,\lambda_1}{\delta^2}\right\}\right\}$

So λ_3 can be expressed:

$$\lambda_3 = (\lambda_1 - \lambda_0)\left(\frac{1}{\delta^2} + \frac{1}{\delta}\right) + \lambda_1$$

Carrying out the same procedure, λ_4 down to λ_k, we obtain:

$$\lambda_4 = (\lambda_1 - \lambda_0)\left(\frac{1}{\delta^3} + \frac{1}{\delta^2} + \frac{1}{\delta}\right) + \lambda_1$$

$$\vdots$$

$$\lambda_k = \frac{(\lambda_1 - \lambda_0)}{\delta}\left(\frac{1}{\delta^{k-2}} + \frac{1}{\delta^{k-3}} + \cdots + 1\right) + \lambda_1$$

Let $1/\delta = r$, then the term in brackets is $(1 + r + r^2 + \cdots + r^{k-3} + r^{k-2})$, whose limit is $1/(1-r)$. Therefore we can take the limit as follows, where λstar denotes the limit.

In[7]:= $\lambda\mathbf{star} = \sum_{k=0}^{\infty} \frac{(\lambda_1 - \lambda_0)}{\delta} \left(\frac{1}{\delta^k}\right) + \lambda_1$

Out[7]= $\lambda_1 + \frac{-\lambda_0 + \lambda_1}{-1 + \delta}$

Furthermore,

In[8]:= **λstar /. {δ -> 4.669, λ₀ -> 3, λ₁ -> 3.4495}**

Out[8]= 3.57201

which is the approximate value at which chaos begins for the logistic equation.

à Question 3

(i)

To show that $x_{n+1} = \lambda x_n(1 - x_n)$ has the same properties as $y_{n+1} = y_n^2 + c$, if

$$c = \frac{\lambda(2-\lambda)}{4} \qquad \text{and} \qquad y_n = \frac{\lambda}{2} - \lambda\, x_n$$

then we substitute these value to obtain:

In[9]:= $\mathbf{Solve}\left[\left(\frac{\lambda}{2} - \lambda\, x_{n+1}\right) == \left(\frac{\lambda}{2} - \lambda\, x_n\right)^2 + \frac{\lambda\,(2-\lambda)}{4},\ x_{n+1}\right]$

Out[9]= $\{\{x_{1+n} \to -\lambda\,(-x_n + x_n^2)\}\}$

Which can readily be expressed $x_{n+1} = \lambda(1 - x_n)$.

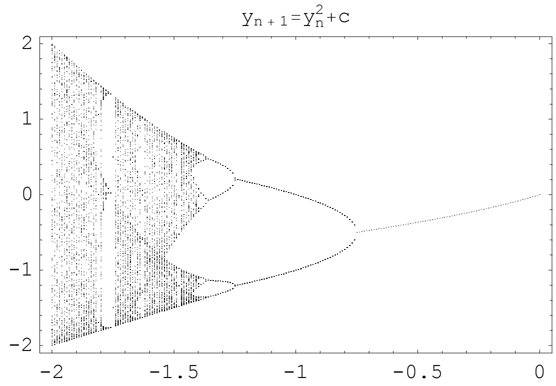
(ii)

In constructing the bifurcation diagram for $y_{n+1} = y_n^2 + c$ we need to know the range for the parameter c. Given $0 < \lambda < 4$ then $-2 < c < 0$. The resulting bifurcation diagram is then derived as follows.

In[10]:= $\mathbf{f[x\_] := x^2 + c}$

In[11]:= **Clear[c]**

```
In[12]:= ListPlot[
           Flatten[Table[
             Transpose[ {
               Table[c, {129}],
               NestList[f, Nest[f, 0.5, 500], 128]
             }],
           {c, -2, 0, 0.01}
         ], 1],
         PlotStyle -> PointSize[0.001],
       Axes -> False, Frame -> True, PlotLabel -> "y_{n+1}=y_n^2+c"];
```



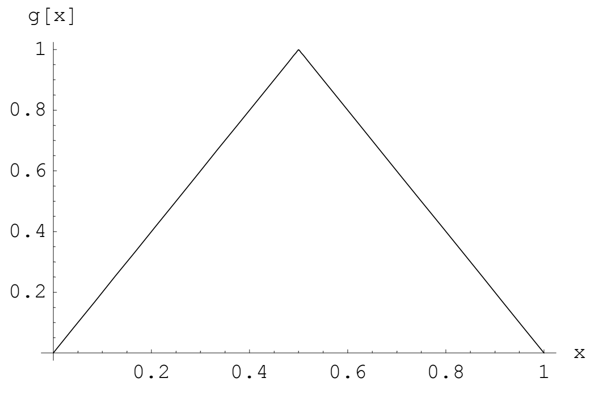
This has the same bifurcation diagram as $x_{n+1} = \lambda\, x_n(1 - x_n)$.

à Question 4

First we plot the tent function.

```
In[13]:= g[x_] := If[0 <= x <= 1 / 2, 2 x, 2 (1 - x)]
```

```
In[14]:= Plot[g[x], {x, 0, 1}, AxesLabel -> {"x", "g[x]"}]
```



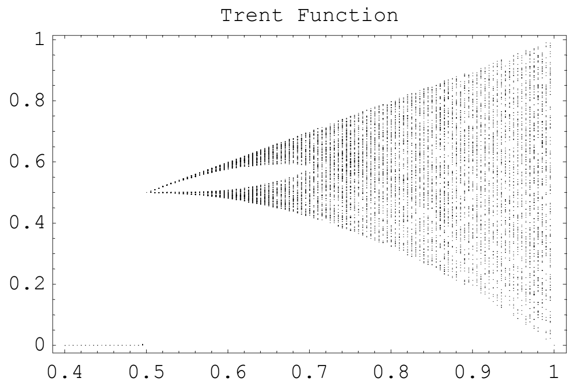

```
Out[14]= - Graphics -
```

which illustrates why it is referred to as the "tend function." The bifurcation diagram is derived as follows.

```
In[15]:= f[x_] := If[0 <= x <= 1 / 2, 2 a x, 2 a (1 - x)]
```

```
In[16]:= Clear[a]
```

```
In[17]:= ListPlot[
           Flatten[Table[
             Transpose[ {
               Table[a, {129}],
               NestList[f, Nest[f, 0.5, 500], 128]
             }],
           {a, 0.4, 1, 0.005}
         ], 1],
         PlotStyle -> PointSize[0.001],
        Axes -> False, Frame -> True, PlotLabel -> "Trent Function"];
```



à Question 5

Given

$$x'(t) = 4x - \frac{\lambda x}{1+4x^2}$$

then in equilibrium

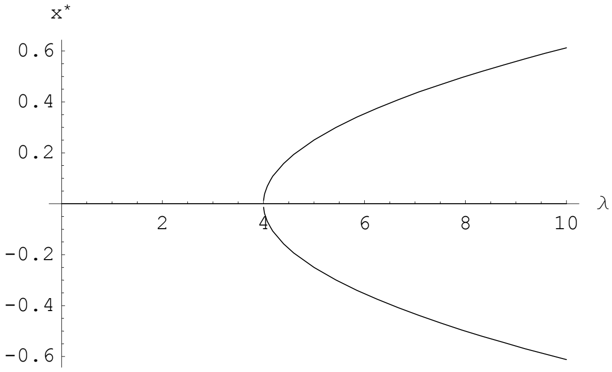
$$4x = \frac{\lambda x}{1+4x^2}$$

In[18]:= **Solve**$\left[4 x == \frac{\lambda x}{1 + 4 x^2}, x\right]$

Out[18]= $\left\{\{x \to 0\}, \left\{x \to -\frac{1}{4}\sqrt{-4+\lambda}\right\}, \left\{x \to \frac{\sqrt{-4+\lambda}}{4}\right\}\right\}$

```
In[19]:= Plot[{0, -1/4 Sqrt[-4 + λ], Sqrt[-4 + λ]/4}, {λ, 0, 10}, AxesLabel → {"λ", "x*"}];
```

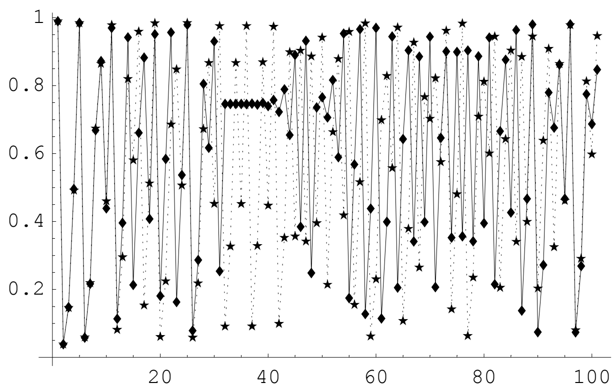
```
Plot::plnr : -1/4 Sqrt[-4 + λ] is not a machine-size real number at λ = 4.1666666666666667`*^-7.

Plot::plnr : -1/4 Sqrt[-4 + λ] is not a machine-size real number at λ = 0.40566991572915795`.

Plot::plnr : -1/4 Sqrt[-4 + λ] is not a machine-size real number at λ = 0.8480879985937368`.

General::stop : Further output of Plot::plnr will be suppressed during this calculation.
```



What this illustrates is a (subcritical) pitchfork bifurcation.

à Question 6

```
In[20]:= f[x_] := 3.94 x (1 - x)

In[21]:= f[x]

Out[21]= 3.94 (1 - x) x

In[22]:= data1 = NestList[f, 0.99, 100];

In[23]:= data2 = NestList[f, 0.9901, 100];

In[24]:= MultipleListPlot[data1, data2, PlotJoined → True];
```



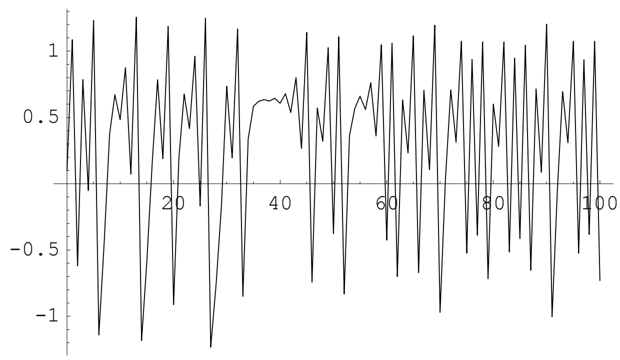
As can be seen from the graph, with just a minor change in the initial condition the two series soon begin to differ.

à Question 7

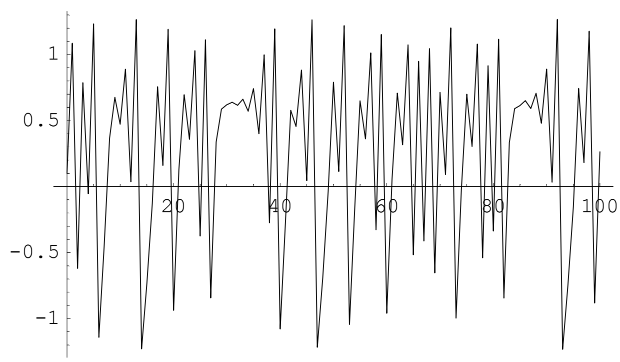
```
In[25]:= Clear[x1, y1]
```

Although this question is for a spreadsheet, we shall deal with it within *Mathematica*.

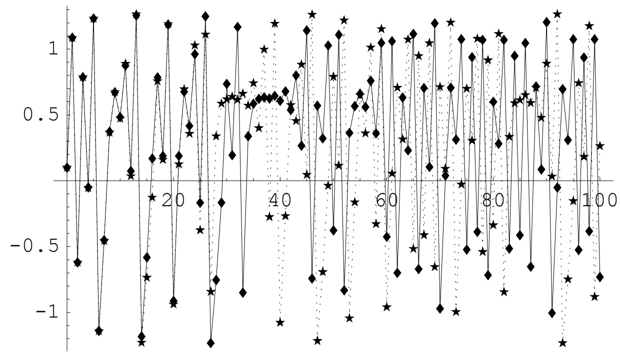
```
In[26]:= {a = 1.4, b = 0.3}

Out[26]= {1.4, 0.3}

In[27]:= {x1[0] = 0.1, y1[0] = 0.1};
         x1[t_] := x1[t] = 1 + y1[t - 1] - a x1[t - 1]^2
         y1[t_] := y1[t] = b x1[t - 1]

In[30]:= data1 = Table[{t, x1[t]}, {t, 0, 100}];

In[31]:= ListPlot[data1, PlotJoined → True];
```




```
In[32]:= {a = 1.4, b = 0.3}

Out[32]= {1.4, 0.3}

In[33]:= {x2[0] = 0.101, y2[0] = 0.1};
         x2[t_] := x2[t] = 1 + y2[t - 1] - a x2[t - 1]^2
         y2[t_] := y2[t] = b x2[t - 1]

In[36]:= data2 = Table[{t, x2[t]}, {t, 0, 100}];

In[37]:= ListPlot[data2, PlotJoined → True];
```



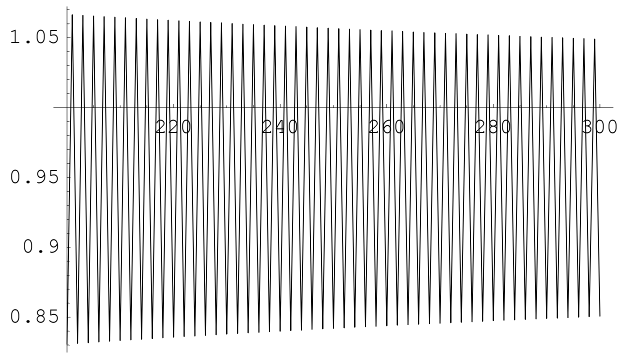
```
In[38]:= MultipleListPlot[data1, data2, PlotJoined → True];
```



The plot of the variable *x* indicates that the two series begin to converge on each other. In other words, after the initial iterations, the Henon map is not sensitive to initial conditions.

à Question 8

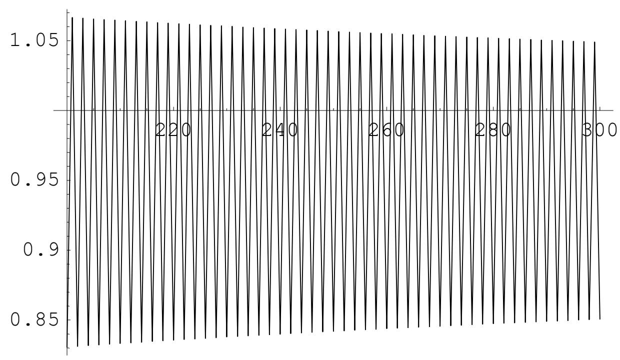
```
In[39]:= {a = 0.3675, b = 0.3}

Out[39]= {0.3675, 0.3}

In[40]:= {x[0] = 0.1, y[0] = 0.1};
         x[t_] := x[t] = 1 + y[t - 1] - a x[t - 1]^2
         y[t_] := y[t] = b x[t - 1]

In[43]:= datax1 = Table[{t, x[t]}, {t, 200, 300}];

   General::spell1 :
    Possible spelling error: new symbol name "datax1" is similar to existing symbol "data1".

In[44]:= ListPlot[datax1, PlotJoined → True];
```



The plot shows a clear two-cycle.

```
In[45]:= {a = 0.9125, b = 0.3}

Out[45]= {0.9125, 0.3}

In[46]:= {x[0] = 0.1, y[0] = 0.1};
         x[t_] := x[t] = 1 + y[t - 1] - a x[t - 1]^2
         y[t_] := y[t] = b x[t - 1]
```

```
In[49]:= datax2 = Table[{t, x[t]}, {t, 200, 300}];

General::spell1 :
 Possible spelling error: new symbol name "datax2" is similar to existing symbol "data2".

In[50]:= ListPlot[datax2, PlotJoined → True];
```



This only just illustrats that a four-cycle results.

à Question 9

The system under investigation is

$x'(t) = -y - z$

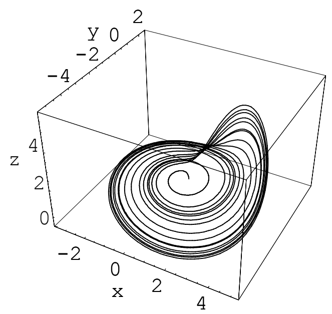
$y'(t) = x + 0.398\,y$

$z'(t) = 2 + z(x - 4)$

$(x(0), y(0), z(0)) = (0.1, 0.1, 0.1)$

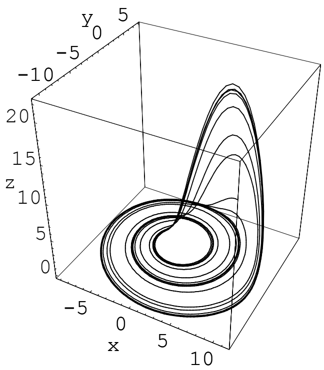
■ (i)

```
In[51]:= Clear[x, y, z]

In[52]:= sol91 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.398 y[t],
           z'[t] == 2 + z[t] (x[t] - 4), x[0] == 0.1, y[0] == 0.1, z[0] == 0.1},
          {x, y, z}, {t, 0, 100}, MaxSteps → 3000]

Out[52]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
          y → InterpolatingFunction[{{0., 100.}}, <>],
          z → InterpolatingFunction[{{0., 100.}}, <>]}}

In[53]:= plot91 = ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol91], {t, 0, 100},
           PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```



■ (ii)

```
In[54]:= Clear[x, y, z]

In[55]:= sol92 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.2 y[t],
           z'[t] == 0.2 + z[t] (x[t] - 5.7), x[0] == 5, y[0] == 5, z[0] == 5},
          {x, y, z}, {t, 0, 100}, MaxSteps → 3000]

Out[55]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
          y → InterpolatingFunction[{{0., 100.}}, <>],
          z → InterpolatingFunction[{{0., 100.}}, <>]}}

In[56]:= plot92 = ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol92], {t, 0, 100},
           PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```



A similar chaotic fold also occurs.

à Question 10

The system under investigation is

$x'(t) = -y - z$

$y'(t) = x + 0.2\,y$

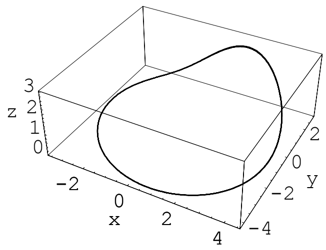
$z'(t) = 0.2 + z(x - c)$

In each plot we ignore the initial part of the trajectory and plot only for $t = 50$ to 100. In each case we assume the initial point is $(x(0), y(0), z(0)) = (1, 1, 1)$.

■ (i) $c = 2.3$

```
In[57]:= Clear[x, y, z]

In[58]:= sol101 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.2 y[t],
           z'[t] == 0.2 + z[t] (x[t] - 2.3), x[0] == 1, y[0] == 1, z[0] == 1},
          {x, y, z}, {t, 0, 100}, MaxSteps → 3000]

Out[58]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
          y → InterpolatingFunction[{{0., 100.}}, <>],
          z → InterpolatingFunction[{{0., 100.}}, <>]}}
```

```
In[59]:= plot101 =
           ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol101], {t, 50, 100},
            PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```



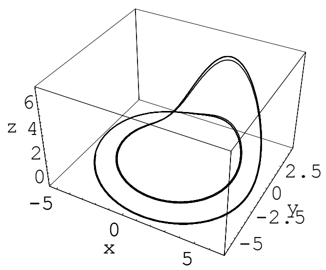
■ (ii) *c* = 3.3

```
In[60]:= Clear[x, y, z]
```

```
In[61]:= sol102 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.2 y[t],
            z'[t] == 0.2 + z[t] (x[t] - 3.3), x[0] == 1, y[0] == 1, z[0] == 1},
           {x, y, z}, {t, 0, 100}, MaxSteps → 3000]
```

```
Out[61]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
           y → InterpolatingFunction[{{0., 100.}}, <>],
           z → InterpolatingFunction[{{0., 100.}}, <>]}}
```

```
In[62]:= plot102 =
           ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol102], {t, 50, 100},
            PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```



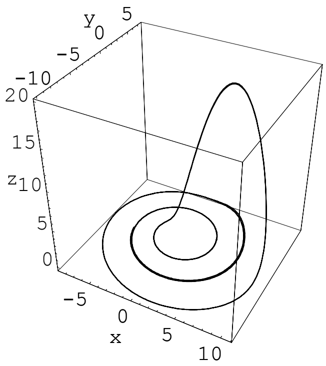
■ (iii) *c* = 5.3

```
In[63]:= Clear[x, y, z]
```

```
In[64]:= sol103 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.2 y[t],
            z'[t] == 0.2 + z[t] (x[t] - 5.3), x[0] == 1, y[0] == 1, z[0] == 1},
           {x, y, z}, {t, 0, 100}, MaxSteps → 3000]
```

```
Out[64]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
           y → InterpolatingFunction[{{0., 100.}}, <>],
           z → InterpolatingFunction[{{0., 100.}}, <>]}}
```

```
In[65]:= plot103 =
           ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol103], {t, 50, 100},
            PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```



■ (iv) $c = 6.3$

```
In[66]:= Clear[x, y, z]
```

```
In[67]:= sol104 = NDSolve[{x'[t] == -y[t] - z[t], y'[t] == x[t] + 0.2 y[t],
            z'[t] == 0.2 + z[t] (x[t] - 6.3), x[0] == 1, y[0] == 1, z[0] == 1},
           {x, y, z}, {t, 0, 100}, MaxSteps → 3000]
```

```
Out[67]= {{x → InterpolatingFunction[{{0., 100.}}, <>],
           y → InterpolatingFunction[{{0., 100.}}, <>],
           z → InterpolatingFunction[{{0., 100.}}, <>]}}
```

```
In[68]:= plot104 =
           ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. sol104], {t, 50, 100},
            PlotPoints → 3000, PlotRange -> All, AxesLabel → {"x", "y", "z"}];
```