12.9 Differential Equations for Nonlinear Oscillators

QUESTION 1

Consider the forced Duffing equation with periodic forcing in the form

$$\ddot{x} + a\dot{x} - x + x^3 = b\cos t$$

where a and b are constants and dot signifies differentiation with respect to t.

To integrate this system, turn the second-order equation above into two first-order equations. Let

$$x_1 = x, \qquad x_2 = \dot{x}$$

Then the system becomes

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -ax_2 + x_1 - {x_1}^3 + b\cos t$$

Then use the Runge-Kutta 4th order method to integrate the system; the method works as follows:

- Let $\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = f(t, X)$, where $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and have initial condition $X(t_0) = X_0$.
- We will be approximating the value of $X(t_0)$ at $t_0, t_1 = t_0 + h, t_2 = t_0 + 2h$ etc., where h is the time step.

The RK4 method works to estimate $X(t + h)$ using **four** slope evaluations as follows:

1. $k_1 = f(t_n, X_n)$
2. $k_2 = f(t_n + \frac{h}{2}, X_n + \frac{h}{2}k_1)$
3. $k_3 = f(t_n + \frac{h}{2}, X_n + \frac{h}{2}k_2)$
4. $k_4 = f(t_n + \frac{h}{2}, X_n + hk_3)$

Then, combine them like this:

$$X_{n+1} = X_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

The programs used to integrate this system using the RK4 method are **Code 1** and **Code 2** on page 20, labelled as

```
fn_vector(t,Y,a,b)
```

which defines x & $\dot{x}$ in a vector, and

```
duffing_phase_plot(h,t_max,a,b,ICs)
```

which carries out the RK4 method and plots a graph of $x(t)$ against $\dot{x}(t)$.

QUESTION 2

The graphs produced by the program for initial conditions $-2 \leq x(0) \leq 2$ and $-2 \leq \dot{x}(0) \leq 2$, and values $a = -0.2,\ 0,\ 0.2$ and $b = 0$, going up to $t = 25$ for $a = -0.2$ and $t = 60$ for the others, and using time step $h = 0.01$, can be seen below:

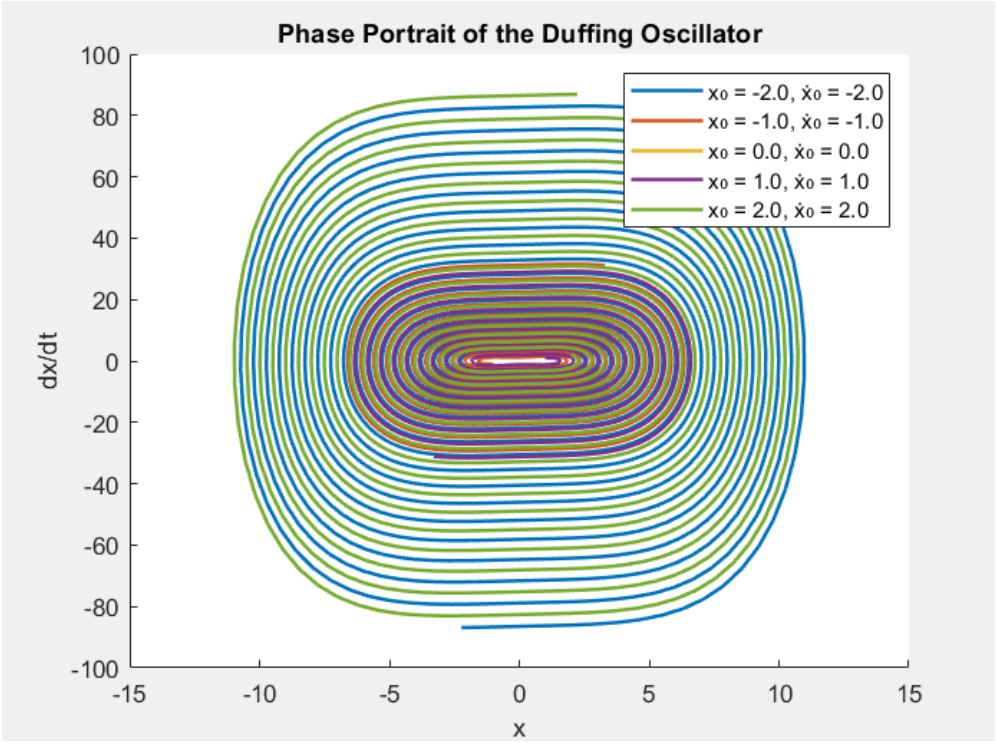

Figure 1: *A plot of the solution to the Duffing equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 0 0; 1 1; 2 2], and values* $a = -0.2$ *and* $b = 0$.

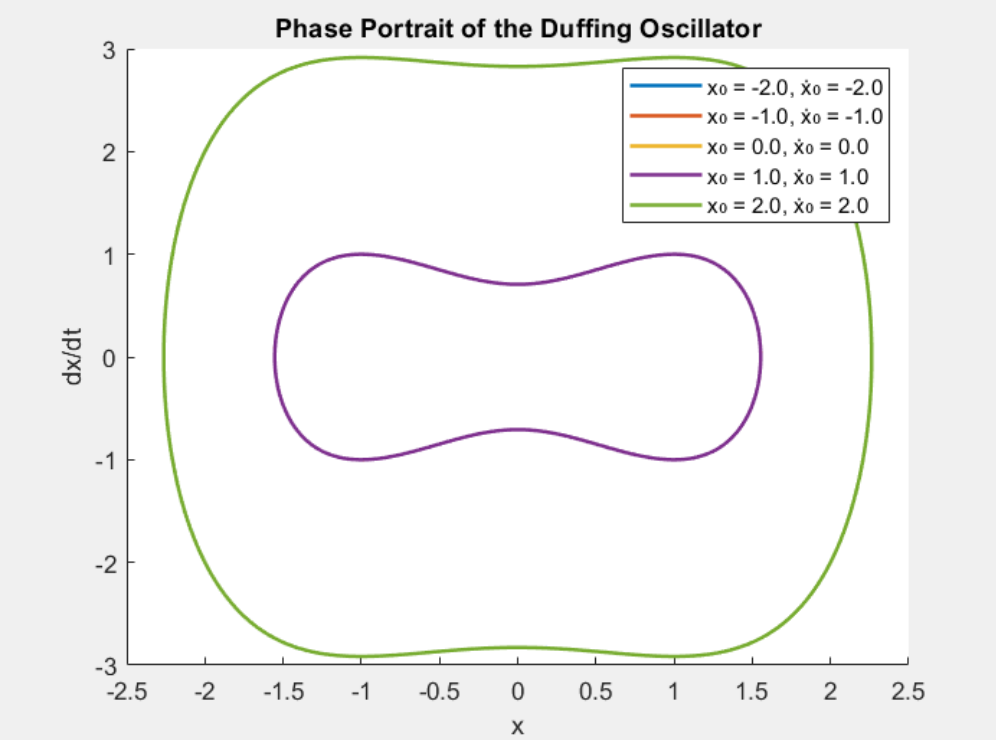

Figure 2: *A plot of the solution to the Duffing equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 0 0; 1 1; 2 2], and values* $a = b = 0$.

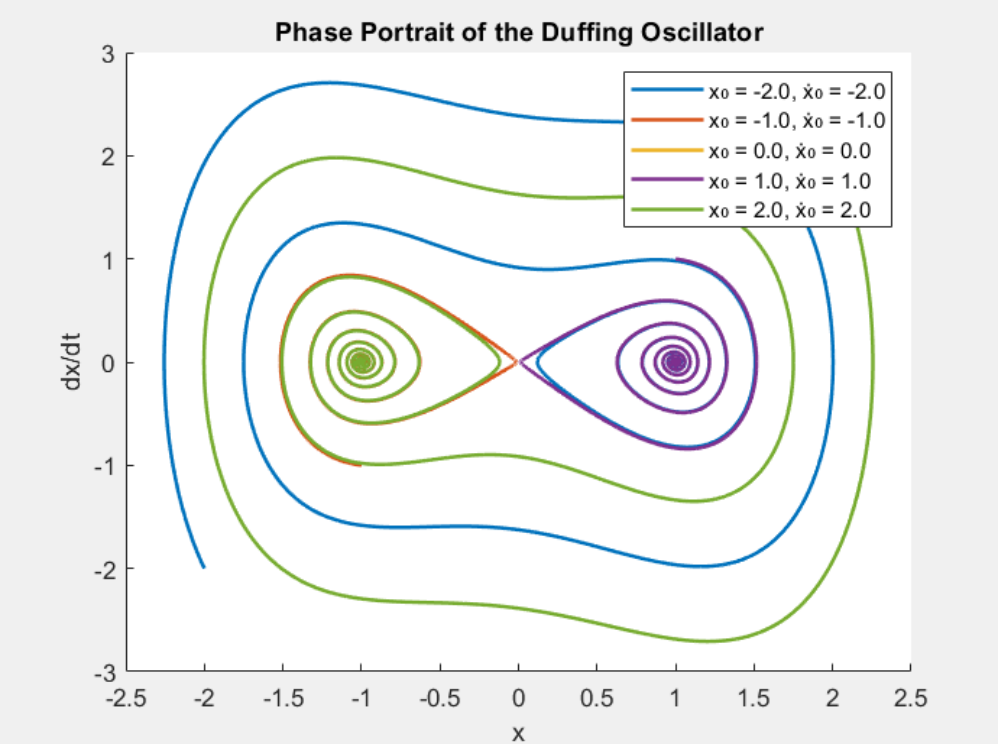

Figure 3: *A plot of the solution to the Duffing equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 0 0; 1 1; 2 2], and values* $a = 0.2$ *and* $b = 0$.

For $a = -0.2, b = 0$, the phase portrait shows spirals that grow outwards with time from the origin, meaning that the system diverges – there's no bound in the motion, and all trajectories move away from equilibrium. However, for $a = 0.2, b = 0$, the phase portrait shows spirals that decay inwards with time towards $(1,0)$, meaning these equilibria are stable. We can also identify a saddle point at the origin.

For $a = b = 0$, we can see that all the orbits are closed – the orbits are centred around 0 like a 'figure 8' shape, meaning the system is periodic.

QUESTION 3

Now set $a = 0.15, b = 0.3$, and experiment with different initial conditions to see the types of solutions it plots. Focus on the solutions from these two initial conditions:

- $\big(x(0), \dot{x}(0)\big) = (1.5,1)$: the phase portrait seems to take the shape of a periodic orbit.
- $\big(x(0), \dot{x}(0)\big) = (-1,-1)$: the phase portrait seems to take the shape of a strange attractor.

A new program will not be necessary for this, as the original program, **Code 2**, is written to take the initial conditions as a parameter, so instead of inputting $ICs = [-2\ -2;\ -1\ -1;\ 0\ 0;\ 1\ 1;\ 2\ 2]$, input $ICs = [-1\ -1;\ 1.5\ 1]$. Furthermore, go up till a significantly larger value of t to see the long-term behaviour effectively: choose $t = 1000$. Keep the same value for the time step: $h = 0.01$.

See the graph below for the solutions obtained:

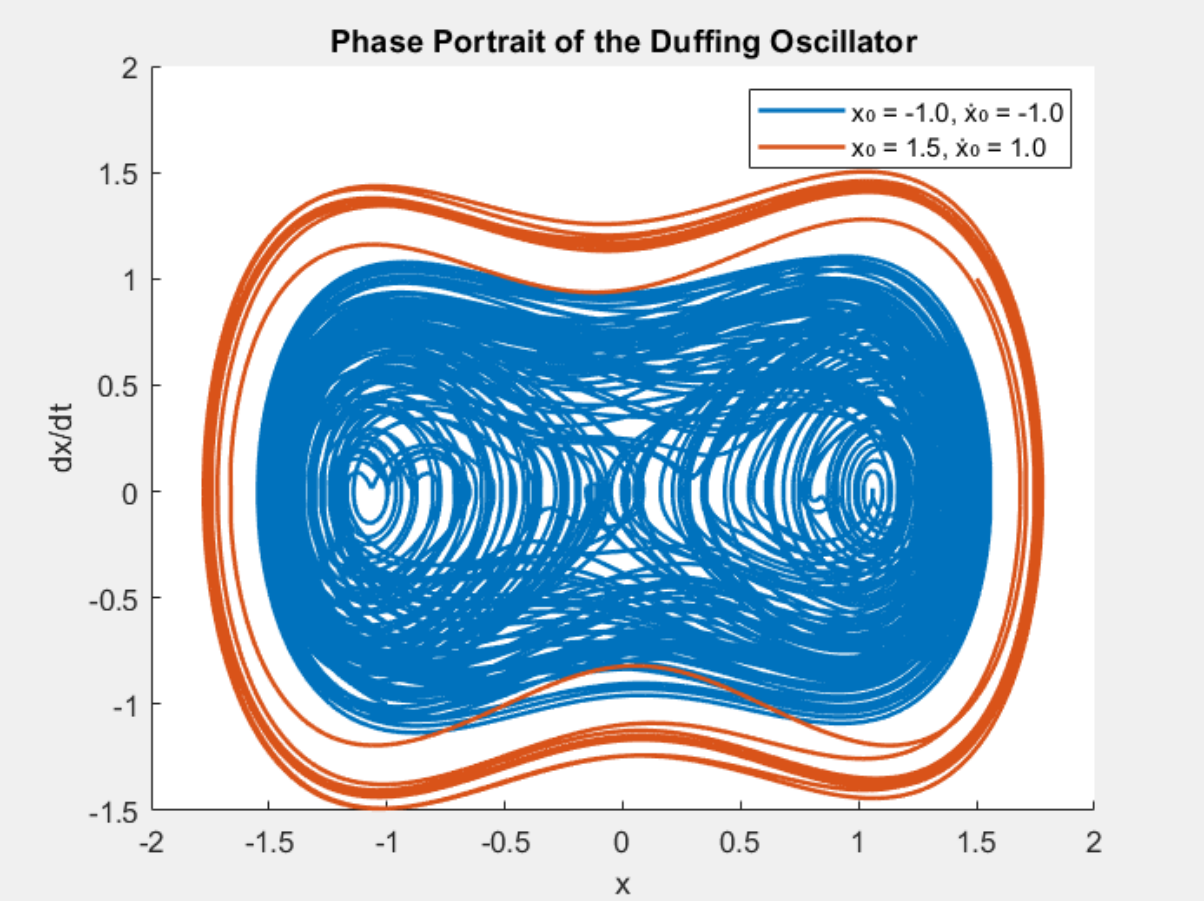

Figure 4: *A plot of the solution to the Duffing equation obtained using the RK4 method, using initial conditions [-1 -1; 1.5 1], and values* $a = 0.15$ *and* $b = 0.3$.

QUESTION 4

This time, aim to repeat the experiment with the same two initial conditions and same parameter values of $a = 0.15, b = 0.3$, but only plot points when $t = 2n\pi$ ($n = 0,1,2, \dots$). To achieve this, turn the vectors of x and $\dot{x}$ values into an array we choose to only add values to if $t = 2n\pi$, and inclusively choose a time step that will go into 2π: choose $h = \pi/100$, as it's also sufficiently small. Furthermore, choose a final value of t that's a multiple of 2π and also large: choose $t = 3200\pi$. Finally, to determine whether to store a point to be plotted or not, implement an 'if' condition to check if $mod(t, 2\pi) < h/4$: don't set it to exactly zero due to floating-point rounding errors.

The program used to integrate this system using the RK4 method is **Code 3** on page 21, labelled as

```
duffing_phase_plot_2(h,t_max,a,b,ICs)
```

This code inclusively makes use of `fn_vector(t,Y,a,b)`.

See the graph below for the solutions obtained:

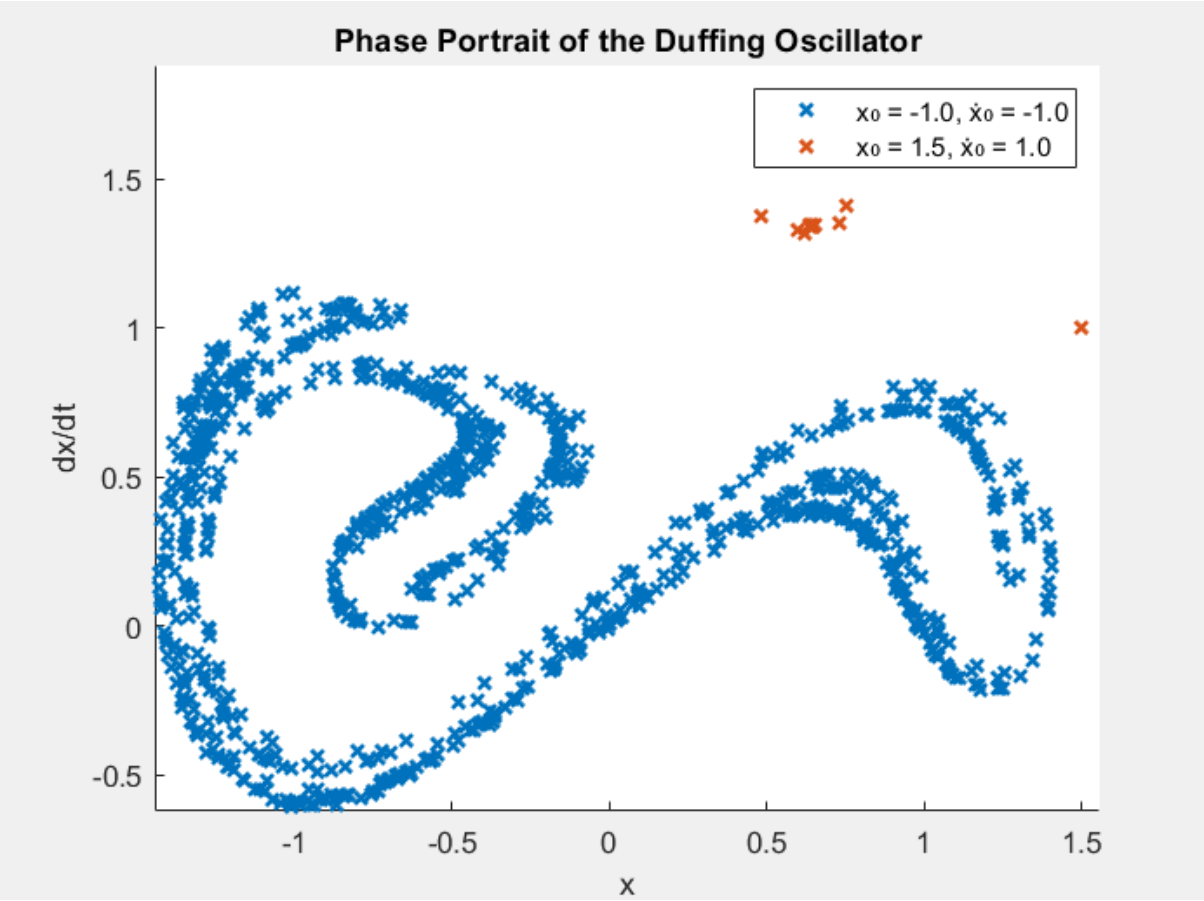

Figure 5: *A plot of the solution to the Duffing equation obtained using the RK4 method, using initial conditions [-1 -1; 1.5 1], values* $a = 0.15$ *and* $b = 0.3$*, and only plotting at time values* $t = 2n\pi$.

As the graph depicts, a lot more of the plotted points come from the initial condition $(-1, -1)$, and very few from $(1.5,1)$ – this aligns with what we would expect, as $(1.5,1)$ is a periodic solution, so presumedly the solution has a period close to 2π, whereas $(-1, -1)$ is the strange attractor, so it is expected to see multiple points generally going inwards.

QUESTION 5

Next, we wish to investigate in detail the behaviour of the system at different values of a between 0.1 and 0.5, keeping $b = 0.3$ and using various initial conditions. It would be useful to explore the behaviour of both a periodic orbit and strange attractor, so stick to the initial conditions used previously of $(1.5,1)$ and $(-1,-1)$. Write another program to automatically loop through five different values of a: $a = 0.1,0.2,0.3,0.4,0.5$.

The program used to loop through $a = 0.1,0.2,0.3,0.4,0.5$ is **Code 4** on page 21, labelled as

```
explore_duffing_a(initial_condition)
```

Using different initial conditions, we observe two distinct types of behaviour in the Duffing oscillator.

Starting from initial condition $(1.5,1)$:

- For smaller values of a, the system begins as a stable periodic orbit, visible as a closed loop in phase space.
- From $a = 0.2$ to around $a = 0.4$, the periodic orbit evolves into a strange attractor, showing visibly chaotic behaviour.
- From $a = 0.4$ to $a = 0.5$, the strange attractor evolves back into a different periodic orbit, one that's smaller.

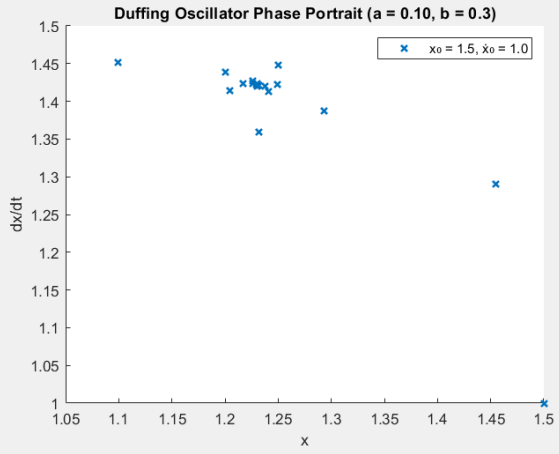

Figure 6a: $a = 0.1$

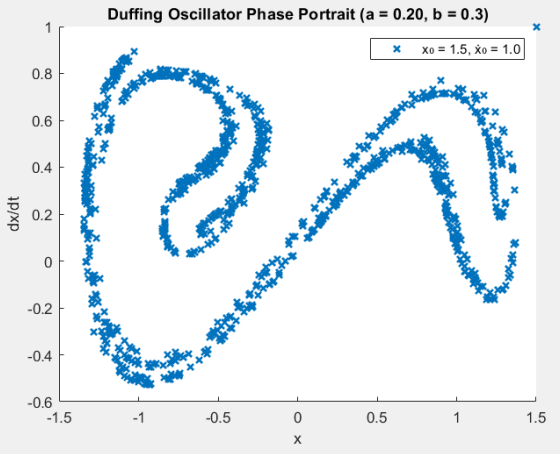

Figure 6b: $a = 0.2$

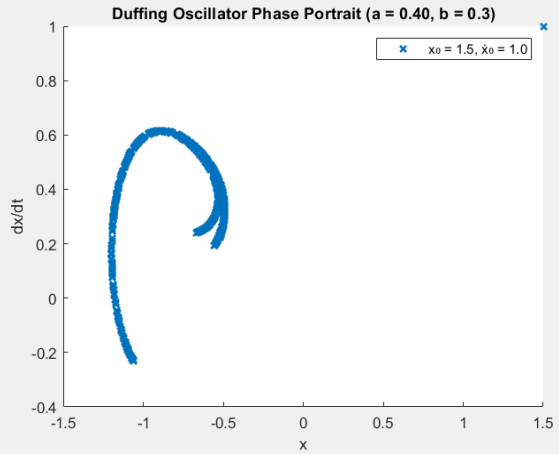

Figure 6c: $a = 0.4$

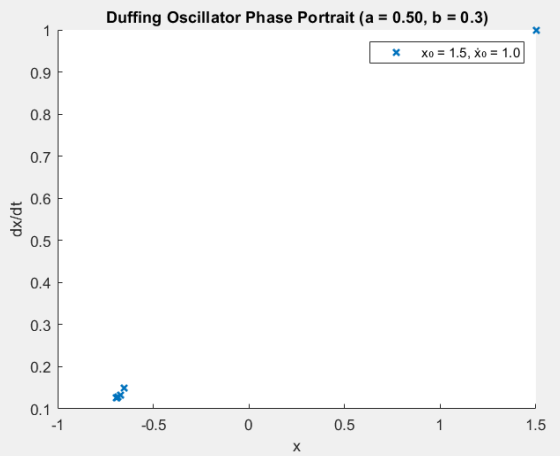

Figure 6d: $a = 0.5$

Figure 6: *Phase portraits of the forced Duffing oscillator for various values of the damping coefficient a (with fixed $b = 0.3$), initial condition [1.5 1] only plotting at time values $t = 2n\pi$.*

In contrast, starting from $(-1, -1)$:

- The system begins as a strange attractor, as expected.
- As the damping parameter a is increased from 0.1 to 0.5, we observe that the strange attractor tends to shrink and become more organized, eventually transitioning into periodic motion as damping becomes strong enough to suppress chaotic behaviour.
- So, the only difference here is that it doesn't have the initial stage of periodic motion.

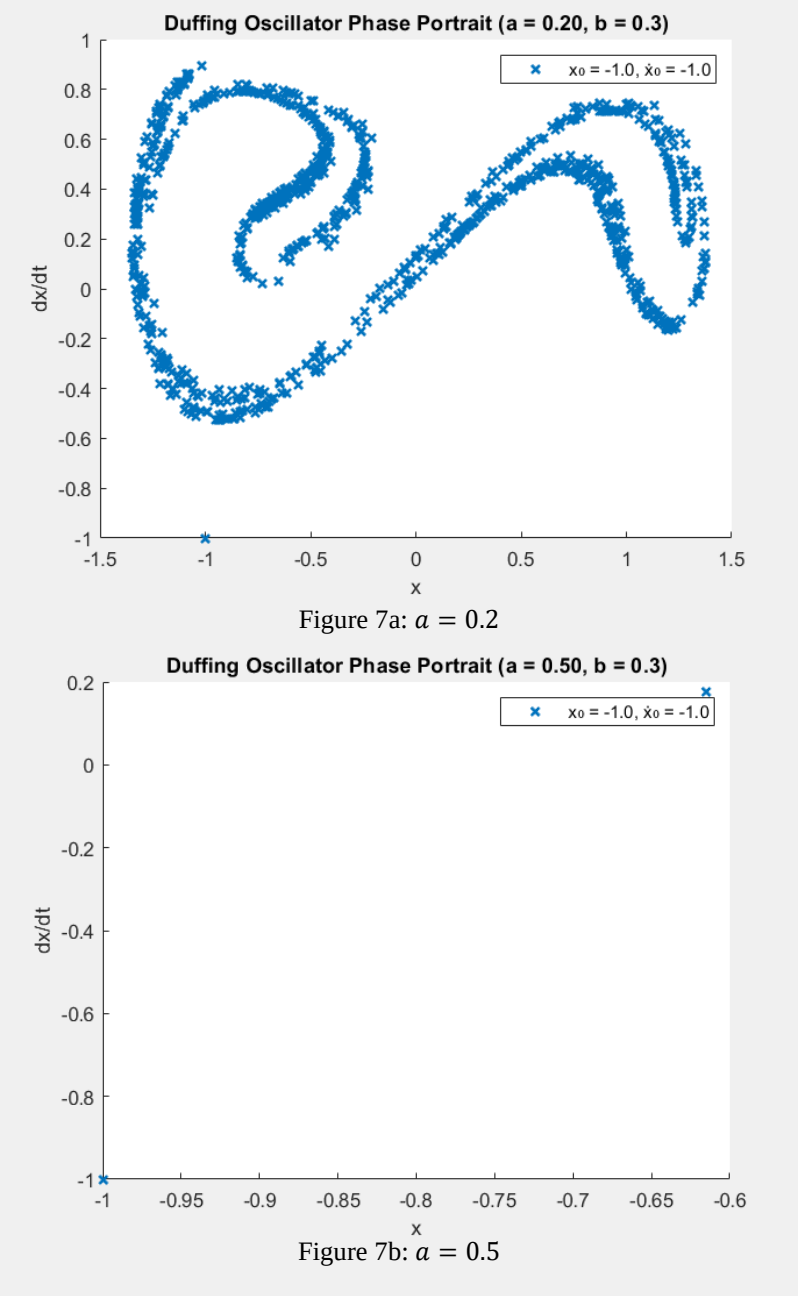

Figure 7a: $a = 0.2$

Figure 7b: $a = 0.5$

Figure 7: *Phase portraits of the forced Duffing oscillator for various values of the damping coefficient a (with fixed $b = 0.3$), initial condition [-1 -1] only plotting at time values $t = 2n\pi$.*

QUESTION 6

Now consider the equation

$$\ddot{x} + (x^2 - b)\dot{x} - ax + x^3 = 0$$

where a and b are constants.

To integrate this system, turn the second-order equation above into two first-order equations. Let

$$x_1 = x, \qquad x_2 = \dot{x}$$

Then the system becomes

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = (b - {x_1}^2)x_2 + ax_1 - {x_1}^3$$

The programs used to integrate this system using the RK4 method are **Code 5** and **Code 6** on pages 22, labelled as

```
fn_vector_2(t,Y,a,b)
```

which defines x & $\dot{x}$ in a vector and will be used in most other codes, and

```
nonlinear_phase_plot(h,t_max,a,b,ICs)
```

which carries out the RK4 method and plots a graph of $x(t)$ against $\dot{x}(t)$ for five suitable initial conditions, going up till $t = 1000$.

Region I: $(a, b) = (-1, -1)$

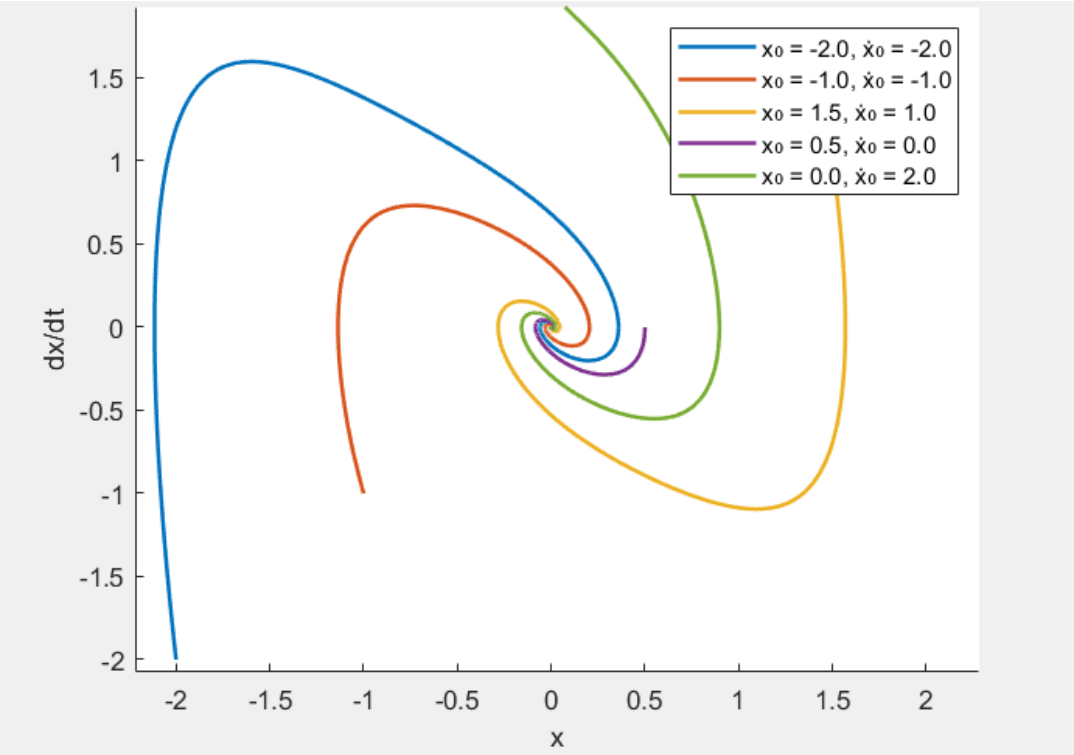

Figure 8: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 1.5 1; 0.5 0; 0 2] and values* $a = -1$ *and* $b = -1$.

Region II: $(a, b) = (-1,1)$

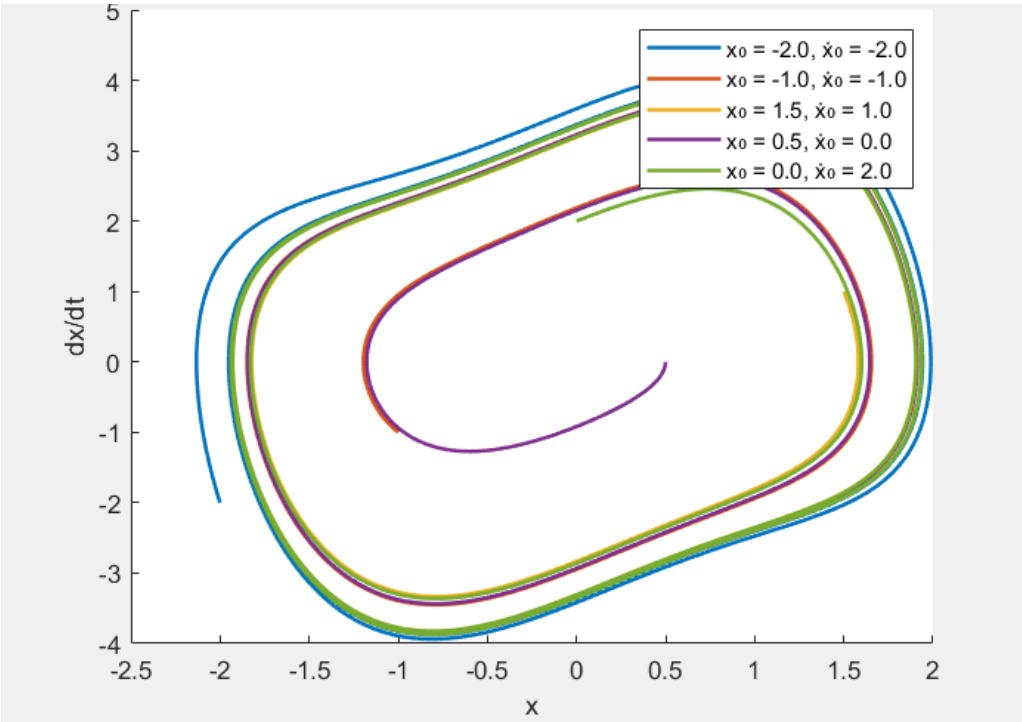

Figure 9: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 1.5 1; 0.5 0; 0 2] and values* $a = -1$ *and* $b = 1$.

Region III: $(a, b) = (0.8,1)$

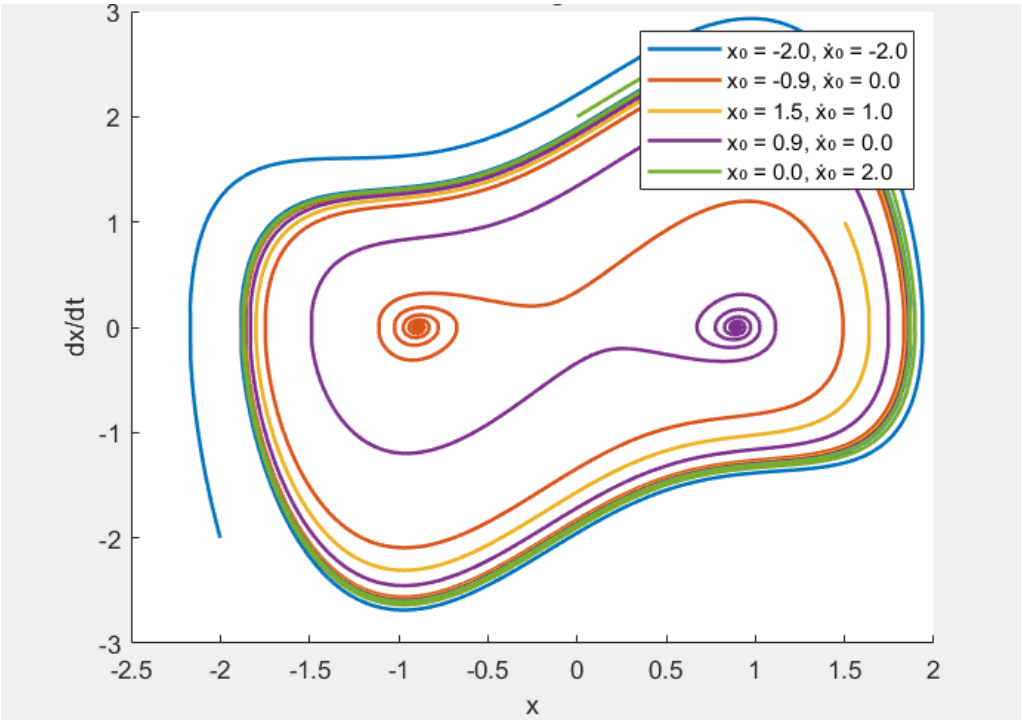

Figure 10: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2;* $-\sqrt{0.8}$ *0; 1.5 1;* $\sqrt{0.8}$ *0; 0 2] and values* $a = 0.8$ *and* $b = 1$.

Region IV: $(a, b) = (1.2,1)$

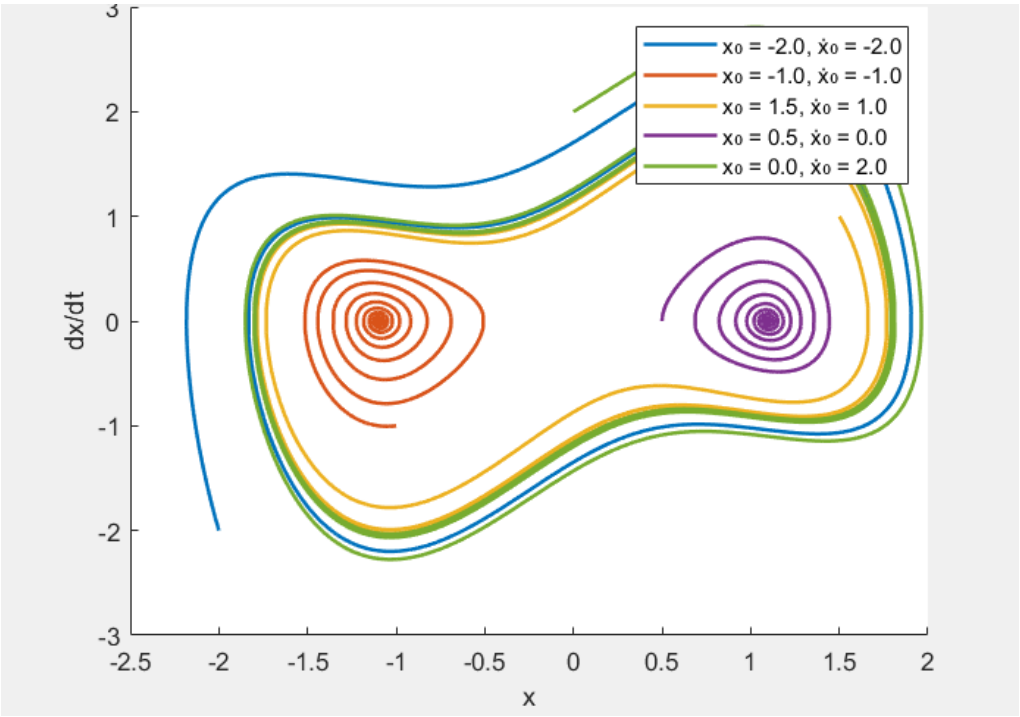

Figure 11: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 1.5 1; 0.5 0; 0 2] and values* $a = 1.2$ *and* $b = 1$.

Region V: $(a, b) = (1.5,1)$

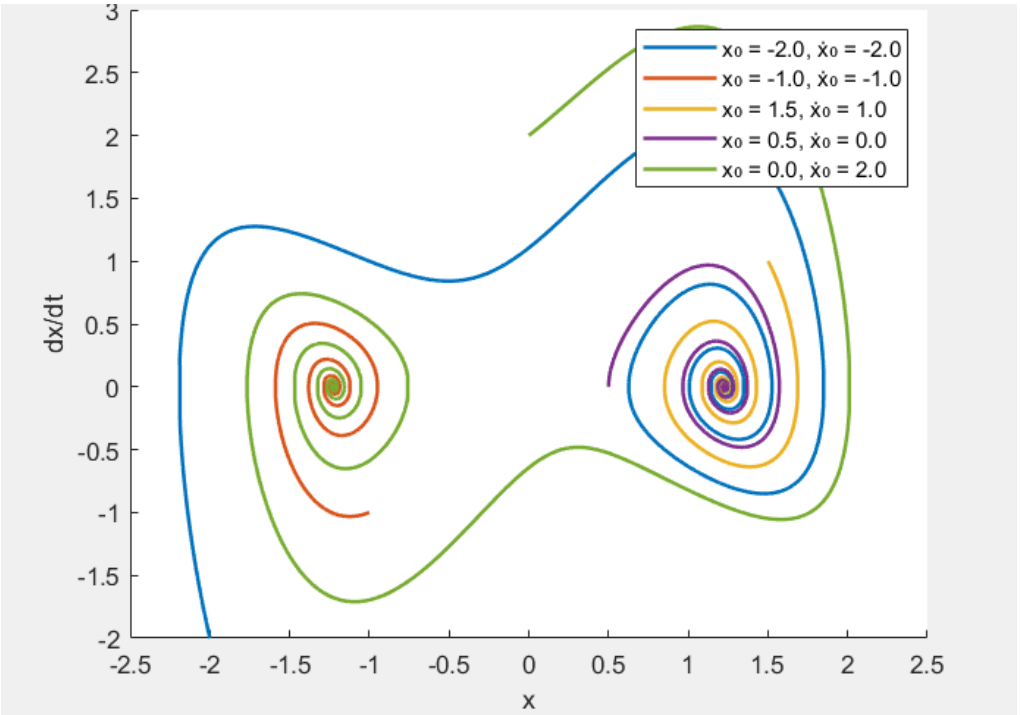

Figure 12: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 1.5 1; 0.5 0; 0 2] and values* $a = 1.5$ *and* $b = 1$.

Region VI: $(a, b) = (2,1)$

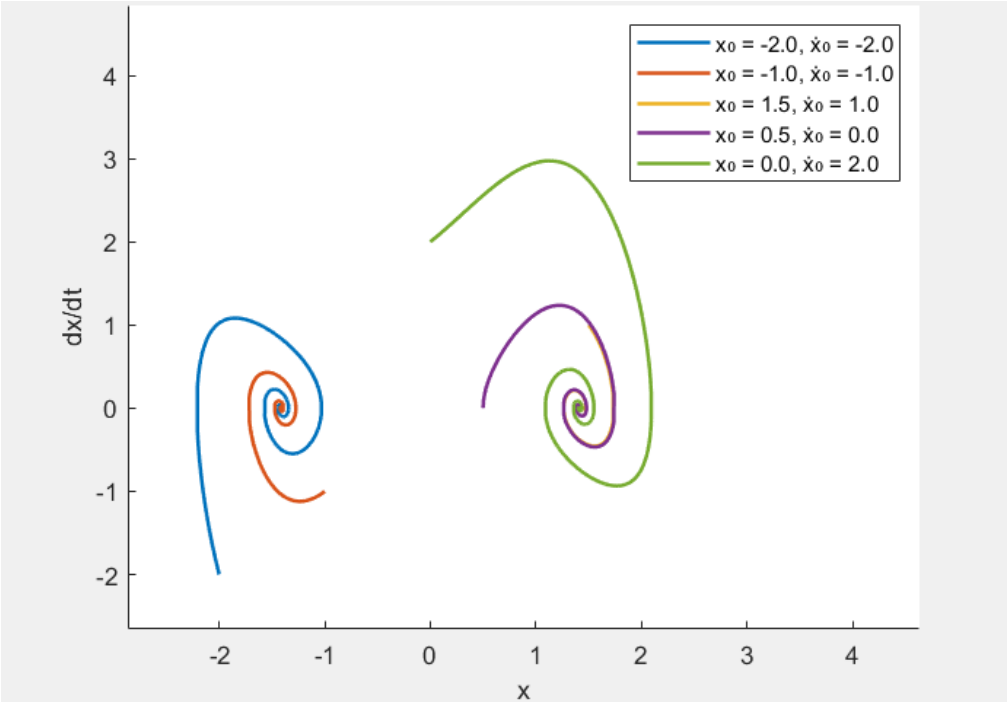

Figure 13: *A plot of the solution to the nonlinear differential equation obtained using the RK4 method, using initial conditions [-2 -2; -1 -1; 1.5 1; 0.5 0; 0 2] and values* $a = 2$ *and* $b = 1$.

QUESTION 7

Recall the coupled first-order differential equations:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = (b - x^2)x_2 + ax_1 - {x_1}^3$$

Find the equilibrium points by equating $\dot{x}_1 = \dot{x}_2 = 0$:

$$\dot{x}_1 = 0 \Rightarrow x_2 = 0$$

Substitute this into the second equation:

$$\dot{x}_2 = ax_1 - {x_1}^3 = x_1(a - {x_1}^2) = 0$$

$$\Rightarrow x_1 = 0, \pm\sqrt{a}$$

So, the three equilibrium points are $(x, \dot{x}) = (x_1, x_2) = (0,0), \left(\sqrt{a}, 0\right), (-\sqrt{a}, 0)$.

Calculate the Jacobian:

$$J = \begin{pmatrix} \frac{d\dot{x_1}}{dx_1} & \frac{d\dot{x_1}}{dx_2} \\ \frac{d\dot{x_2}}{dx_1} & \frac{d\dot{x_2}}{dx_2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix}$$

Now consider what happens near the equilibrium points in each region using the Jacobian:

Region I: $(a, b) = (-1, -1)$

$$J = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 - 1 - 3{x_1}^2 & -1 - {x_1}^2 \end{pmatrix}$$

We only need to consider equilibrium point $(0,0)$, as the other two equilibrium points require a to be positive, so the Jacobian becomes:

$$J = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}$$

Find the eigenvalues:

$$\det(J - \lambda I) = 0 \Rightarrow \begin{vmatrix} -\lambda & 1 \\ -1 & -1 - \lambda \end{vmatrix} = \lambda(\lambda + 1) + 1 = \lambda^2 + \lambda + 1 = 0$$

$$\Rightarrow \lambda = \frac{-1 \pm \sqrt{1 - 4}}{2} = -\frac{1}{2} \pm i\frac{\sqrt{3}}{2}$$

- The eigenvalues are complex numbers, meaning we would expect to see spirals.
- They have negative real parts, so it's **stable** – it'll spiral towards the origin, which is exactly what we see.

Region II: $(a, b) = (-1,1)$

Once again, only consider equilibrium point $(0,0)$, so the Jacobian becomes:

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}$$

Find the eigenvalues:

$$\det(J-\lambda I)=0 \Rightarrow \begin{vmatrix} -\lambda & 1 \\ -1 & 1-\lambda \end{vmatrix} = \lambda(\lambda-1)+1 = \lambda^2-\lambda+1 = 0$$

$$\Rightarrow \lambda = \frac{1 \pm \sqrt{1-4}}{2} = \frac{1}{2} \pm i\frac{\sqrt{3}}{2}$$

- The eigenvalues are complex numbers, meaning we would expect to see spirals.
- They have positive real parts, so it's **unstable.**
- This aligns with what we see – points near the fixed point, the origin, spiral out into a periodic orbit.

Region III: $(a,b)=(0.8,1)$

$$J = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + 0.8 - 3{x_1}^2 & 1 - {x_1}^2 \end{pmatrix}$$

Equilibrium point (0,0):

$$J = \begin{pmatrix} 0 & 1 \\ 0.8 & 1 \end{pmatrix}$$

$$\det(J-\lambda I)=0 \Rightarrow \begin{vmatrix} -\lambda & 1 \\ 0.8 & 1-\lambda \end{vmatrix} = \lambda(\lambda-1)-0.8 = \lambda^2-\lambda-0.8 = 0$$

$$\Rightarrow \lambda = \frac{1 \pm \sqrt{1+3.2}}{2} = \frac{1}{2} \pm \frac{\sqrt{4.2}}{2}$$

- The eigenvalues are real numbers of opposite sign, so we expect to see a saddle point near the origin, which we do see.

Equilibrium point $\left(\sqrt{a},0\right)=\left(\sqrt{0.8},0\right)$:

$$J = \begin{pmatrix} 0 & 1 \\ -1.6 & 0.2 \end{pmatrix}$$

$$\det(J-\lambda I)=0 \Rightarrow \begin{vmatrix} -\lambda & 1 \\ -1.6 & 0.2-\lambda \end{vmatrix} = \lambda(\lambda-0.2)+1.6 = \lambda^2-0.2\lambda+1.6 = 0$$

$$\Rightarrow \lambda = \frac{0.2 \pm \sqrt{0.04-6.4}}{2}$$

- The eigenvalues are complex numbers, meaning we would expect to see spirals.
- They have positive real parts, so it's **unstable** – it'll spiral away from $\left(\sqrt{0.8},0\right)$, which is exactly what we see; we see it spiralling away from the equilibrium point into a periodic orbit.
- Exact same behaviour for $\left(-\sqrt{a},0\right)=\left(-\sqrt{0.8},0\right)$.

Region IV: $(a,b)=(1.2,1)$

$$J = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + 1.2 - 3{x_1}^2 & 1 - {x_1}^2 \end{pmatrix}$$

Equilibrium point (0,0):

$$J = \begin{pmatrix} 0 & 1 \\ 1.2 & 1 \end{pmatrix}$$

- The eigenvalues are real numbers of opposite sign, so we expect to see a saddle point near the origin, which we do see.

Equilibrium point $(\sqrt{a}, 0) = (\sqrt{1.2}, 0)$:

$$J = \begin{pmatrix} 0 & 1 \\ -2.4 & -0.2 \end{pmatrix}$$

$$\det(J - \lambda I) = 0 \Rightarrow \begin{vmatrix} -\lambda & 1 \\ -2.4 & -0.2 - \lambda \end{vmatrix} = \lambda(\lambda + 0.2) + 2.4 = \lambda^2 + 0.2\lambda + 2.4 = 0$$

$$\Rightarrow \lambda = \frac{-0.2 \pm \sqrt{0.04 - 9.6}}{2}$$

- The eigenvalues are complex numbers, meaning we would expect to see spirals.
- They have negative real parts, so it's stable – it'll spiral towards $(\sqrt{1.2}, 0)$, which is exactly what we see.
- Exact same behaviour for $(-\sqrt{a}, 0) = (-\sqrt{1.2}, 0)$.

Region V: $(a, b) = (1.5,1)$

$$J = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + 1.5 - 3{x_1}^2 & 1 - {x_1}^2 \end{pmatrix}$$

- For this region, we see the exact same behaviour as in region IV.
- The eigenvalues are real numbers of opposite sign, so we expect to see a saddle point near the origin, which we do see.
- The eigenvalues are complex numbers, meaning we would expect to see spirals.
- They have negative real parts, so it's **stable** – it'll spiral towards $(\sqrt{1.2}, 0)$, which is exactly what we see.
- Exact same behaviour for $(-\sqrt{a}, 0) = (-\sqrt{1.2}, 0)$.

Region VI: $(a, b) = (2,1)$

$$J = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + a - 3{x_1}^2 & b - {x_1}^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2x_1x_2 + 2 - 3{x_1}^2 & 1 - {x_1}^2 \end{pmatrix}$$

- For this region, we see the exact same behaviour as in region IV and V, but much stronger convergence, so the saddle at the origin isn't really seen.

QUESTION 8

Now consider the forced Van der Pol oscillator

$$\ddot{x} + (x^2 - b)\dot{x} - ax + x^3 = 0$$

where a and b are constants.

To integrate this system, we are told that we can write the above equation in Liénard coordinates as

$$\dot{x} = y - a\left(\frac{x^3}{3} - x\right)$$

$$\dot{y} = -x + 1 + b$$

The programs used to integrate this system using the RK4 method are **Code 7** and **Code 8** on pages 22 & 23, labelled as

```
forced_vdp_vector(t,Y,a,b)
```

which defines the Liénard coordinates in a vector, and

```
vdp_phase_plot(h,t_max,a,b,ICs)
```

which carries out the RK4 method and plots a graph of $x(t)$ against $\dot{x}(t)$ for suitable initial conditions, going up till $t = 1000$ and using time step $h = 0.001$.

Using this program, we can identify periodic orbits for $b = -0.001$ and $a = 1, 5, 10$.

$\boldsymbol{a} = 1$:

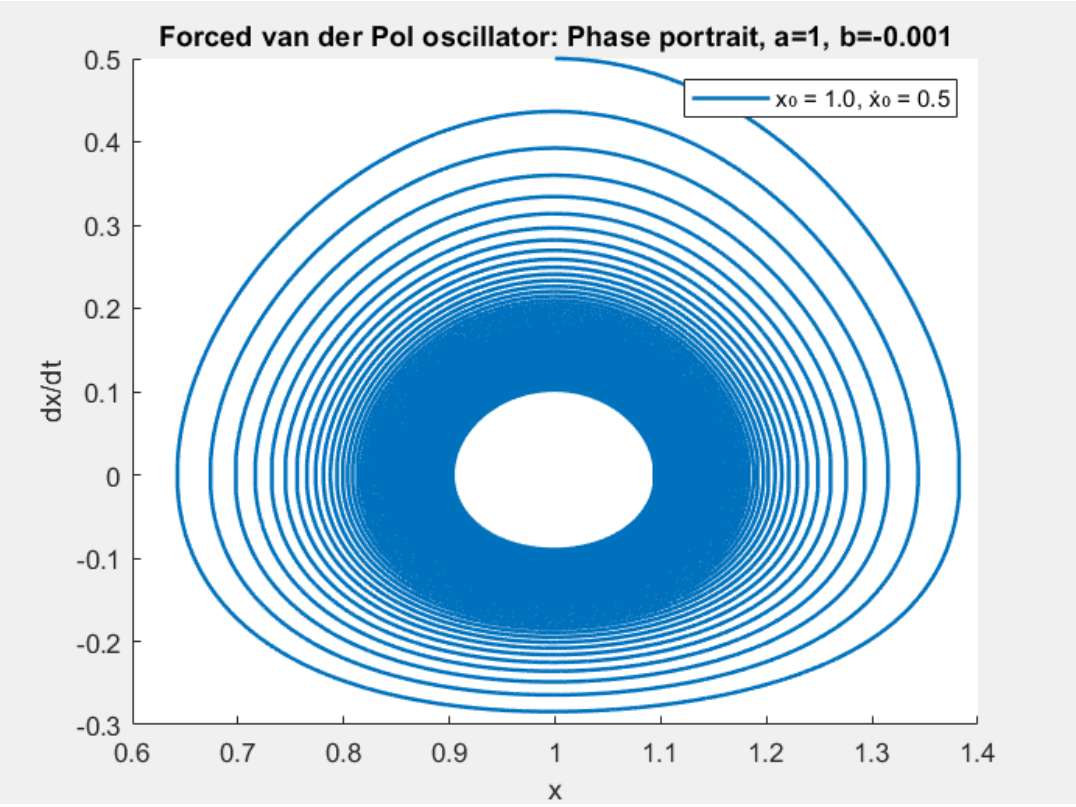

Figure 14: *A plot of the solution to the Van der Pol differential equation obtained using the RK4 method, using initial conditions [1 0.5] and values* $a = 1$ *and* $b = -0.001$.

$\boldsymbol{a} = 5$:

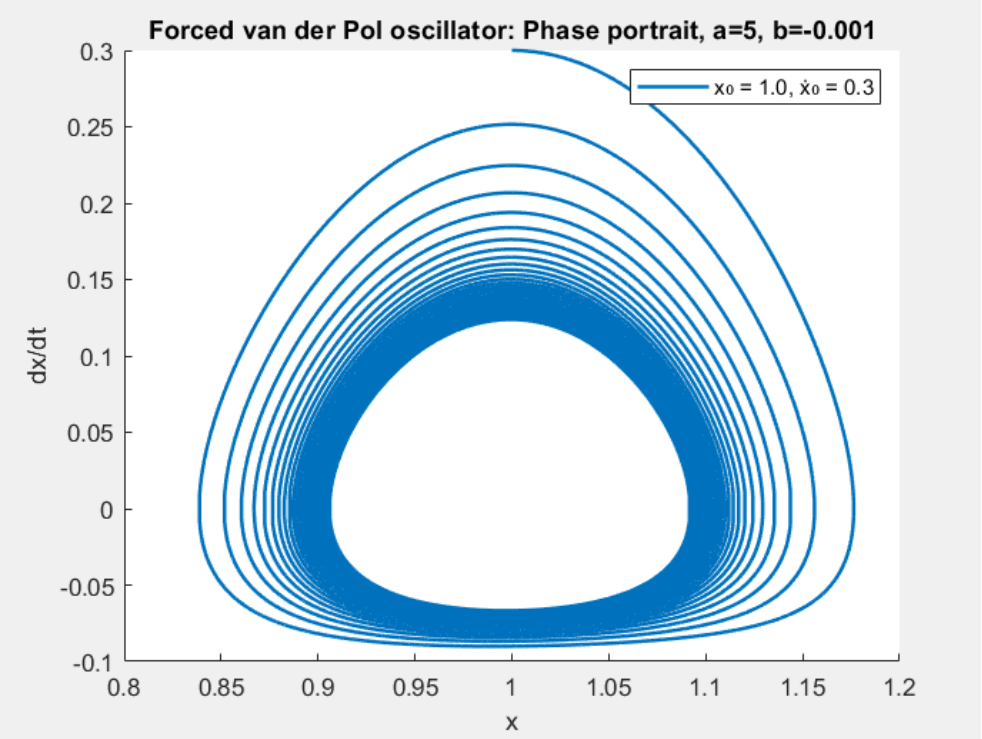

Figure 15: *A plot of the solution to the Van der Pol differential equation obtained using the RK4 method, using initial conditions [1 0.3] and values* $a = 5$ *and* $b = -0.001$.

$\boldsymbol{a} = 10$:

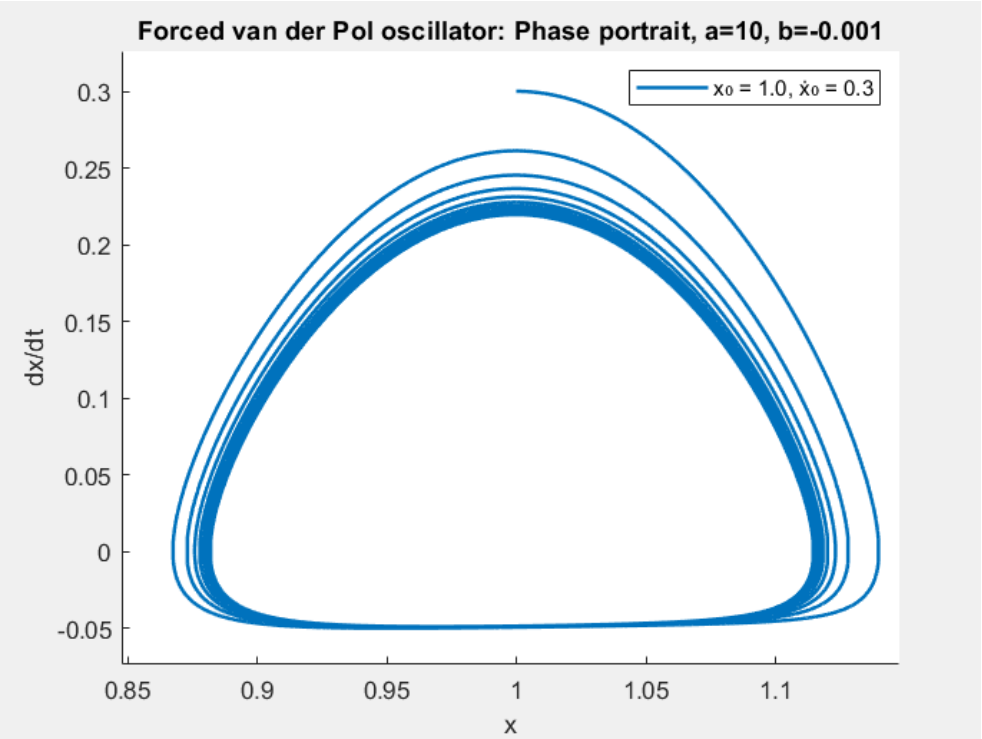

Figure 16: *A plot of the solution to the Van der Pol differential equation obtained using the RK4 method, using initial conditions [1 0.3] and values* $a = 10$ *and* $b = -0.001$.

QUESTION 9

Now, investigate the evolution of the periodic orbit for $b \in [-0.1, 0)$ at each of these values of a.

$a = 1$:

- As b gets smaller in magnitude from -0.1, the periodic orbit shrinks but still remains centred around the fixed point.
- $b = 0$ is a **Hopf Bifurcation,** so in the idealised Van der Pol system, the phase portrait should switch from a periodic orbit to a stable fixed point (inward spiral) at exactly $b = 0$.
- However, due to small numerical error from the RK4 method, this switch instead happens at around $b = 0.004$ to $b = 0.005$ instead, still close to 0.

See some diagrams below for reference, initial conditions picked so that the diagram is as clear as possible.

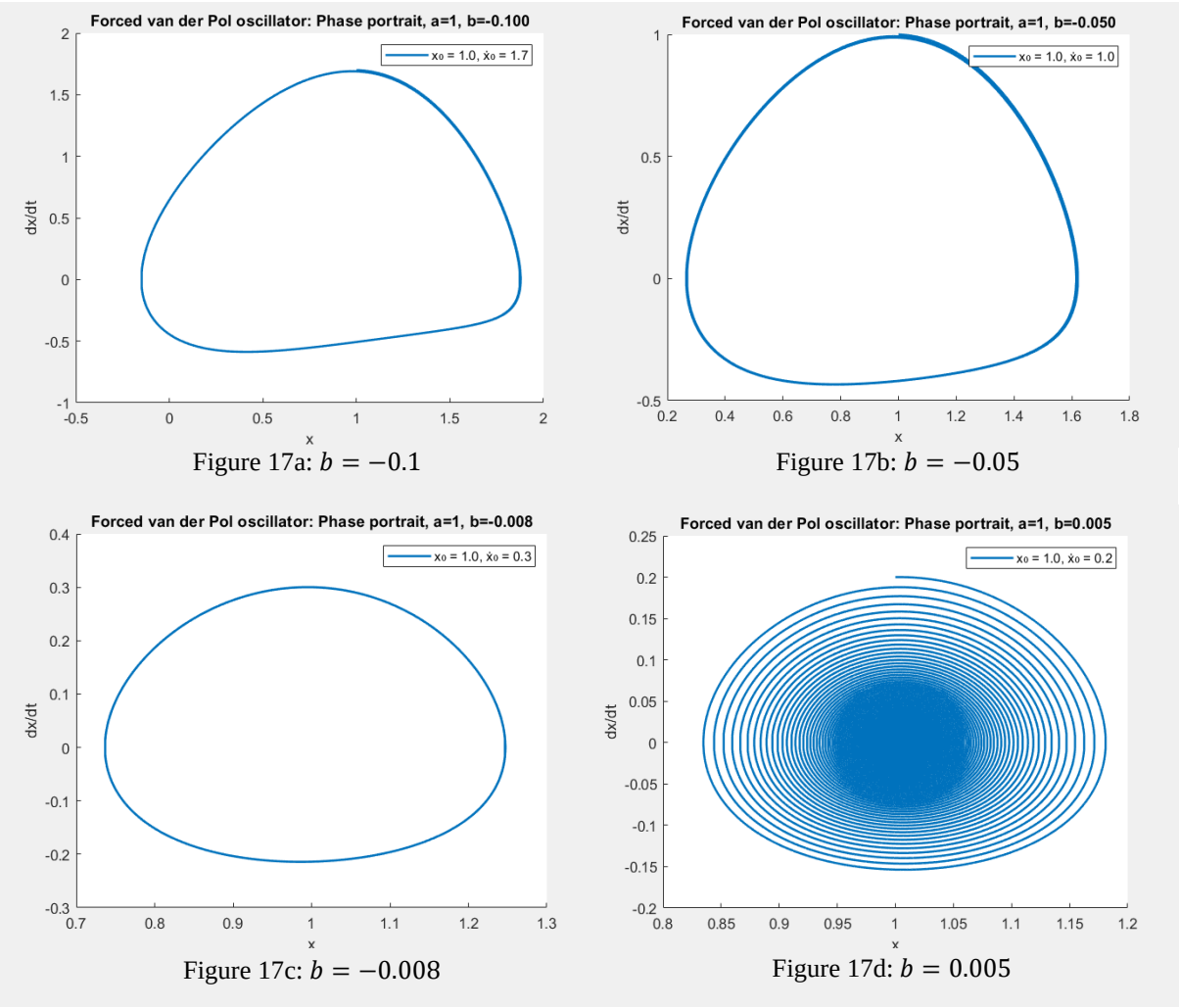

Figure 17a: $b = -0.1$

Figure 17b: $b = -0.05$

Figure 17c: $b = -0.008$

Figure 17d: $b = 0.005$

Figure 17: *Phase portraits of the Van der Pol oscillator obtained using the RK4 method for various values of b (with fixed* $a = 1$*).*

$a = 5$:

- The periodic orbit remains the exact same shape until $b = -0.005$, where it significantly shrinks in size and changes shape.
- From that point onwards, as b approaches 0, the periodic orbit shrinks a little as b decreases in magnitude, the same way it did for $a = 1$, until around a similar value of $b = 0.004$, it switches to a stable fixed point.

See some diagrams below for reference, initial conditions picked so that the diagram is as clear as possible.

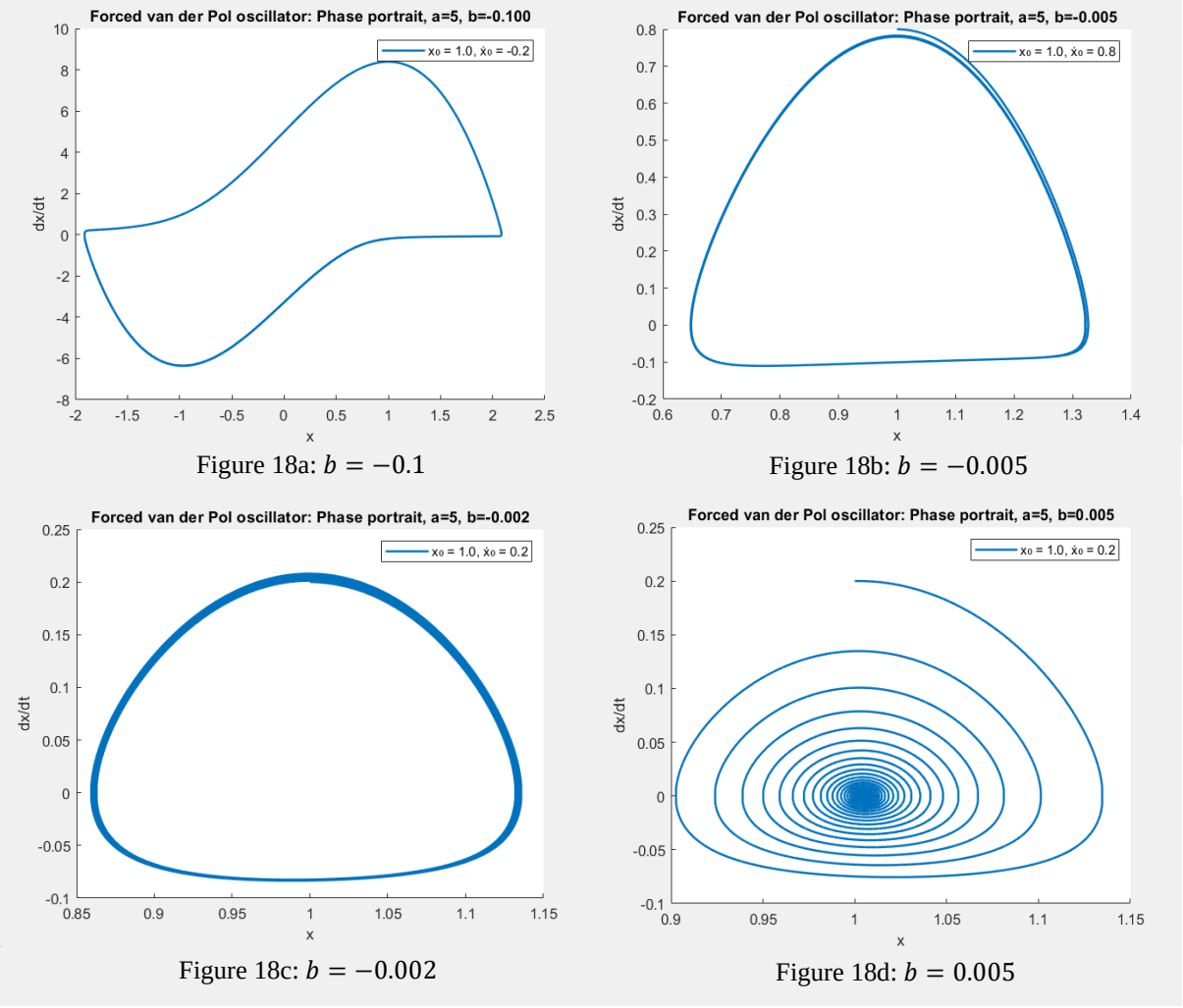

Figure 18a: $b = -0.1$

Figure 18b: $b = -0.005$

Figure 18c: $b = -0.002$

Figure 18d: $b = 0.005$

Figure 18: *Phase portraits of the Van der Pol oscillator obtained using the RK4 method for various values of b (with fixed $a = 5$).*

$a = 10$:

- The periodic orbit remains the exact same shape (the same shape that it was for $a = 5$ but bigger) until $b = -0.001$, where it significantly shrinks in size and changes shape.
- From that point onwards, as b approaches 0 in small increments, the periodic orbit shrinks a little, the same way it did for $a = 1$, until around a similar value of $b = 0.002$, it switches to a stable fixed point.

QUESTION 10

For large values of a, the periodic orbit takes a completely different shape to the usual circular orbit until negative values of b very close to 0. This is because when $a \gg 1$, the damping term is very large. In fact, if we write $Y = y\beta^{-1}$ the Liénard coordinates become

$$\begin{pmatrix} \dot{x} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} \beta\left(Y - \frac{x^3}{3} + x\right) \\ (-x + 1 + b)\beta^{-1} \end{pmatrix}$$

So $\dot{Y}$ is very small and Y varies only slowly. Either $|\dot{x}| \gg 1$ or $Y \approx \frac{x^3}{3} - x$. If it's the latter, it's only stable if the gradient of the curve is positive ($x^2 > 1$).

So, the trajectory follows a branch of the curve $Y = \frac{x^3}{3} - x$ with positive gradient until it runs out and then moves quickly to another branch of the curve with positive gradient. The geometry of the curve is what allows it to be a periodic orbit.

However, for smaller a, the system is nearly Hamiltonian, so this doesn't happen.

There are various numerical difficulties that can arise in calculating such an orbit, especially near bifurcation points:

- A very small step size h is often needed, otherwise the RK4 method may miss the fine structure of the orbit, leading to inaccurate damping or growth.
- Near the bifurcation, $b = 0$, the orbit decays very slowly towards the fixed point, meaning it may take a really long time until the true behaviour is clear.
- Adding on to the previous point, the periodic orbit becomes really small as b approaches 0, so it can become really difficult to distinguish between a periodic orbit and fixed point.
- There is a high sensitivity to initial conditions: small changes in initial conditions can lead to dramatic changes in long-term behaviour.
- In some areas, the oscillator has very fast and very slow dynamics happening at once – it becomes 'stiff'; methods like RK4 can struggle with this and would require very small time steps to remain stable.

Programs

CODE 1

```
function [vector_derivative] = fn_vector(t,Y,a,b)
vector_derivative=[Y(2), (-a*Y(2) + Y(1) - (Y(1))^3 + b*cos(t))];
end
```

CODE 2

```
function duffing_phase_plot(h,t_max, a,b, ICs)
num_conditions = size(ICs, 1);

colors = lines(num_conditions);
legend_entries = cell(1, num_conditions);
figure;
hold on;

for ic = 1:num_conditions
    Y_n=ICs(ic,:);
    t_n=0;
    vector_Y=zeros(1,1+(t_max/h));
    vector_Z=zeros(1,1+(t_max/h));
    vector_Y(1)=Y_n(1);
    vector_Z(1)=Y_n(2);
    for n=1:t_max/h
        k1=h*fn_vector(t_n,Y_n,a,b);
        k2=h*fn_vector(t_n+(h/2), Y_n+((k1)/2),a,b);
        k3=h*fn_vector(t_n+(h/2), Y_n+((k2)/2),a,b);
        k4=h*fn_vector(t_n+h, Y_n+k3,a,b);
        Y_n=Y_n + (k1+2*k2+2*k3+k4)/6;
        t_n=t_n + h;
        vector_Y(n+1)=Y_n(1);
        vector_Z(n+1)=Y_n(2);
    end
    plot(vector_Y,vector_Z,'LineWidth',1.5,'Color',colors(ic,:))
    legend_entries{ic} = sprintf('x₀ = %.1f, ẋ₀ = %.1f', ICs(ic,1), ICs(ic,2));
end
xlabel('x')
ylabel('dx/dt')
title('Phase Portrait of the Duffing Oscillator')
legend(legend_entries);
end
```

CODE 3

```
function duffing_phase_plot_2(h,t_max, a,b, ICs)
num_conditions = size(ICs, 1);
colors = lines(num_conditions);
legend_entries = cell(1, num_conditions);
figure;
hold on;

for ic = 1:num_conditions
    Y_n=ICs(ic,:);
    t_n=0;
    vector_Y=[Y_n(1)];
    vector_Z=[Y_n(2)];
    for n=1:t_max/h
        k1=h*fn_vector(t_n,Y_n,a,b);
        k2=h*fn_vector(t_n+(h/2), Y_n+((k1)/2),a,b);
        k3=h*fn_vector(t_n+(h/2), Y_n+((k2)/2),a,b);
        k4=h*fn_vector(t_n+h, Y_n+k3,a,b);
        Y_n=Y_n + (k1+2*k2+2*k3+k4)/6;
        t_n=t_n + h;
        if mod(t_n, 2*pi) < h/2
            vector_Y=[vector_Y, Y_n(1)];
            vector_Z=[vector_Z, Y_n(2)];
        end
    end
    plot(vector_Y,vector_Z,'x','LineWidth',1.5,'Color',colors(ic,:))
    legend_entries{ic} = sprintf('x₀ = %.1f, ẋ₀ = %.1f', ICs(ic,1), ICs(ic,2));
end
xlabel('x')
ylabel('dx/dt')
title('Phase Portrait of the Duffing Oscillator')
legend(legend_entries);
end
```

CODE 4

```
function explore_duffing_a(initial_condition)
a_values = [0.1, 0.2, 0.3, 0.4, 0.5];
h = pi/100;
t_max = 3200*pi;
b = 0.3;

for i = 1:length(a_values)
    figure;
    duffing_phase_plot_2(h, t_max, a_values(i), b, initial_condition);
    title(sprintf('Duffing Oscillator Phase Portrait (a = %.2f, b = 0.3)',
a_values(i)));
end
```

CODE 5

```
function [vector_derivative] = fn_vector_2(t,Y,a,b)
vector_derivative=[Y(2), ((b - (Y(1))^2)*Y(2) + a*Y(1) - (Y(1))^3)];
end
```

CODE 6

```
function nonlinear_phase_plot(h,t_max, a,b, ICs)
num_conditions = size(ICs, 1);

colors = lines(num_conditions);
legend_entries = cell(1, num_conditions);
figure;
hold on;

for ic = 1:num_conditions
    Y_n=ICs(ic,:);
    t_n=0;
    vector_Y=zeros(1,1+(t_max/h));
    vector_Z=zeros(1,1+(t_max/h));
    vector_Y(1)=Y_n(1);
    vector_Z(1)=Y_n(2);
    for n=1:t_max/h
        k1=h*fn_vector_2(t_n,Y_n,a,b);
        k2=h*fn_vector_2(t_n+(h/2), Y_n+((k1)/2),a,b);
        k3=h*fn_vector_2(t_n+(h/2), Y_n+((k2)/2),a,b);
        k4=h*fn_vector_2(t_n+h, Y_n+k3,a,b);
        Y_n=Y_n + (k1+2*k2+2*k3+k4)/6;
        t_n=t_n + h;
        vector_Y(n+1)=Y_n(1);
        vector_Z(n+1)=Y_n(2);
    end
    plot(vector_Y,vector_Z,'LineWidth',1.5,'Color',colors(ic,:))
    legend_entries{ic} = sprintf('x₀ = %.1f, ẋ₀ = %.1f', ICs(ic,1), ICs(ic,2));
end
xlabel('x')
ylabel('dx/dt')
title(sprintf('Phase Portrait of the Nonlinear Oscillator, a=%.1f, b=%.1f', a, b))
legend(legend_entries);
end
```

CODE 7

```
function [vector_derivative] = forced_vdp_vector(t,Y,a,b)
vector_derivative=[Y(2) - a*((Y(1)^3)/3 - Y(1)), -Y(1) + 1 + b];
end
```

CODE 8

```
function vdp_phase_plot(h,t_max, a,b, ICs)
num_conditions = size(ICs, 1);

colors = lines(num_conditions);
legend_entries = cell(1, num_conditions);
figure;
hold on;

for ic = 1:num_conditions
    Y_n(1)=ICs(ic,1);
    Y_n(2)=ICs(ic,2)+a*((Y_n(1)^3)/3 - Y_n(1));
    t_n=0;
    vector_Y=zeros(1,1+(t_max/h));
    vector_Z=zeros(1,1+(t_max/h));
    vector_Y(1)=Y_n(1);
    vector_Z(1)=ICs(ic,2);
    for n=1:t_max/h
        k1=h*forced_vdp_vector(t_n,Y_n,a,b);
        k2=h*forced_vdp_vector(t_n+(h/2), Y_n+((k1)/2),a,b);
        k3=h*forced_vdp_vector(t_n+(h/2), Y_n+((k2)/2),a,b);
        k4=h*forced_vdp_vector(t_n+h, Y_n+k3,a,b);
        Y_n=Y_n + (k1+2*k2+2*k3+k4)/6;
        t_n=t_n + h;
        derivative = Y_n(2) - a*((Y_n(1)^3)/3 - Y_n(1));
        vector_Y(n+1)=Y_n(1);
        vector_Z(n+1)=derivative;
    end
    plot(vector_Y,vector_Z,'LineWidth',1.5,'Color',colors(ic,:))
    legend_entries{ic} = sprintf('x₀ = %.1f, ẋ₀ = %.1f', ICs(ic,1), ICs(ic,2));
end
xlabel('x')
ylabel('dx/dt')
title(sprintf('Forced van der Pol oscillator: Phase portrait, a=%d, b=%.3f', a,
b))
legend(legend_entries);
end
```