Question 1.1

Given $\frac{du}{dt} = 2ut$ where u(t=0) = 2

⇨ $\frac{1}{u}du = 2t\ dt$

Integrating, we get:

⇨ $\ln u = t^2 + C$ or $u = ke^{t^2}$

Plugging in initial values of t = 0 and u =2 in the above equation, we get k = 2.

Therefore, closed form solution for u(t) is

$$u(t) = 2e^{t^2}$$

Question 1.2

Numerical integration schemes are applied to ordinary differential equations often when their closed-form solution is difficult to find or when the value of a definite integral needs to be estimated. There are multiple numerical integration techniques and below are described 4 of them:

1. **Euler's method**
   To apply Euler's method of numerical integration, the initial conditions of the function must be known. In our question we have u(t=0) = 2. In Euler's method, we start from the initial point and calculate the next point using the gradient of the function and a step size. Therefore, in the question, considering a step size h, we get;
   $$t_{i+1} = t_i + h$$
   $$U_{i+1} = U_i + h.(\frac{dU}{dt})_i$$
2. **Midpoint method**
   The midpoint method is used to estimate the area under a curve if it is difficult to solve analytically. In this method, we divide the input range, say (a, b) into n equal segments where size of each segment h = (b-a)/n is nothing but the step size. The value of the function is calculated at the midpoint of each segment i and eventually the estimation of the definite integral reduces to the sum of the area of the rectangles at each segment i with width equal to the step size h and height equal to u(t = i_{midpt}).
   $$\int_a^b u(t)dt = \sum_n u(t = i_{midpt}).h$$
3. **Runge Kutta 2nd order**
   Similar to Euler's Method, the Runge Kutta second order numerical method requires initial conditions of the ordinary differential equation. Starting from the initial value of u(t), it calculates the subsequent values using the weighted average of two gradients as following;
   $$t_{i+1} = t_i + h$$
   $$K_1 = h * f(t_i, u_i)$$
   $$K_2 = h * f\left(\left(t_i + \frac{h}{2}\right), \left(u_i + \frac{K_1 * h}{2}\right)\right)$$
   $$where\ \frac{du}{dt} = f(t, u)$$
   $$u_{i+1} = u_i + K_2$$
4. **Runge Kutta 4th order**
   Similar to Runge Kutta 2nd order, the 4th order takes weighted mean of 4 gradients to calculate the next value. The formulas are as below;
   $$t_{i+1} = t_i + h$$
   $$K_1 = h * f(t_i, u_i)$$
   $$K_2 = h * f\left(\left(t_i + \frac{h}{2}\right), \left(u_i + \frac{K_1}{2}\right)\right)$$
   $$K_3 = h * f\left(\left(t_i + \frac{h}{2}\right), \left(u_i + \frac{K_2}{2}\right)\right)$$
   $$K_4 = h * f((t_i + h), (u_i + K_3))$$
   $$where\ \frac{du}{dt} = f(t, u)$$
   $$u_{i+1} = u_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

Question 1.3

	h=0.01s	h=0.05s	h=0.1s	h=0.5s	h=1.0s
Euler	7.576	31.010	50.199	94.196	103.196
RK 2nd	5.510	22.558	36.525	68.963	81.196
RK 4th	0.000	0.001	0.023	5.026	23.863

Fig 1: Error in estimating u(t = 2s)

	h=0.01s	h=0.05s	h=0.1s	h=0.5s	h=1.0s
Euler	1.562	6.619	11.093	23.738	27.572
Midpoint	0.002	0.045	0.181	3.934	11.361
RK 2nd	1.048	4.408	7.322	14.789	17.572
RK 4th	0.000	0.000	0.002	0.807	3.317

Fig 2: Error in estimating area under the curve u(t) for 0 ≤ t ≤ 2

The above tables show the respective errors in estimating the value of u(t) at t=2s and the area of the curve u(t) where 0 ≤ t ≤ 2 for each of the four numerical methods and each step size. In figure 1, since the midpoint method uses the function u(t) to determine bar heights, it makes sense to use it to only estimate the area under the curve. On the other hand, the other techniques begin at the initial value u(t=0s) and evolve from that point based on gradient values and step size and therefore, the error in estimating the value of u(t=x) only for the other three techniques are valid.

We can see from the error values that any estimates made using numerical methods will give close to real values if we take a small step size. In our analysis, we have used step sizes of 0.01, 0.05, 0.1, 0.5 and 1 seconds. In figure 1 which outlines the errors in estimating u(t=2s), we see that the largest step size of 1 second gives the maximum error on all three methods whereas the least step size of 0.01 seconds gives the best result. Additionally, we see that a choice of method also makes a difference in the error measurement of our estimates. Figure 1 shows that even when minimal step size is considered, the best method to use to estimate the time evolution of function is the Runge Kutta 4^{th} order method as it takes into account the weighted mean of 4 gradients to calculate the succeeding values of the function from the current point.

Figure 2 demonstrates the error in estimating the area under u(t) for 0 ≤ t ≤ 2. As in the previous case, considering smaller step sizes leads to a better estimate. For calculating area under curves for functions that are difficult to solve analytically, the midpoint method with a step size of 0.01 seconds comes very close to the true value. Although Runge Kutta 4^{th} order comes out as the clear winner among the 4 methods used but the midpoint method is a very good alternative for solving definite integrals.

Question 2

Gradient descent is an optimization algorithm that is used to find the minima of a function when they are difficult to solve analytically. In the question, the given function to minimize is:

$$f(x,y) = (x^4 + y^4) - (21x^2 + 13y^2) + 2xy(x+y) - (14x + 22y) + 170$$

The way gradient descent works is as follows:

- Start at a random point ‘w’ on the plane.
- Calculate the gradient of the function at point ‘w’.
- Calculate the next point by moving a step from ‘w’ in a direction in which the value of the function drops the fastest. This implies going in the opposite direction of the gradient if gradient is positive and in the direction of the gradient if it is negative. The step size (h) is set at the beginning for whom a smaller value is beneficial for better results.
$$w_{i+1} = w_i - h.\nabla f(w_i)$$
- Repeat previous step until a point is reached from where value of f(x, y) starts to rise again and stop. The coordinates where the algorithm stops are potential minima points (might be local or global).
- Start from another random point and follow all steps again until true minima are obtained.

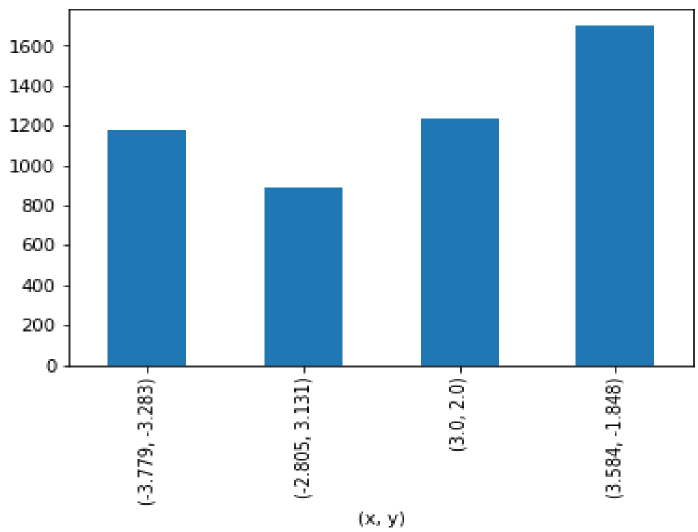
From the above equation, we get

$$\nabla f(x,y) = < (4x^3 + 2y^2 + 4xy - 42x - 14), (4y^3 + 2x^2 + 4xy - 26y - 22) >$$

The results obtained are as follows:

(x, y)	f(x, y)	count
(-3.779, -3.283)	0.0	1178
(-2.805, 3.131)	0.0	889
(3.0, 2.0)	0.0	1235
(3.584, -1.848)	0.0	1698

The histogram below shows the frequency of coordinates where the algorithm found a minima. The table above shows the value of f(x, y) at those coordinates.



Some problems encountered during gradient descent are:

1. Getting local minima instead of global minima and therefore, requiring higher number of iterations of the algorithm so that global minima are well represented.
2. Overshooting minima by taking high value of step size. In the analysis, a step size of 0.001 units was considered.
3. Oscillations in a region of the plane which is like a plateau and therefore does not differ in gradient by much for each subsequent point calculated.

To overcome these problems, a momentum term α is included in the formula to calculate the (i+1)th point so that $w_{i+1} = \alpha . w_i - h . \nabla f(w_i)$

Question 3

In the given plot we can see that y = 3 and y = -1 are critical points of the system. Equations 2 and 4 suffice that condition and are candidates to generate the given direction field. Additionally, we see that the system decreases in value at y = 0. Calculating the gradient for both equations at y = 0 gives:

$$Eq\ II: \quad \frac{dy}{dt} = -y^2 + 2y + 3$$

$$=> \quad \frac{d^2y}{dt^2} = -2y + 2 = -2(0) + 2 = 2$$

$$Eq\ IV: \quad \frac{dy}{dt} = y^2 - 2y - 3$$

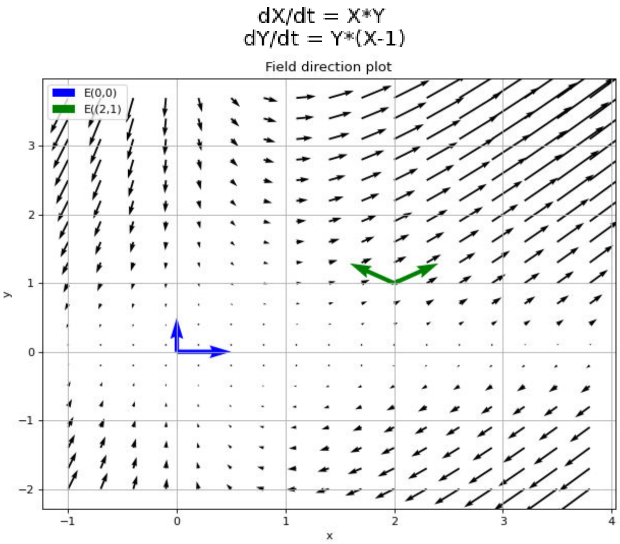
$$=> \quad \frac{d^2y}{dt^2} = 2y - 2 = 2(0) - 2 = -2$$

As we see a negative tangent at y = 0 for equation 4, it means that $\frac{dy}{dt} = (y+1)(y-3)$ is the equation that produced the given field plot.

Question 4.4

For equilibria points to be stable, they need to have real and negative eigenvalues. Point (0,0) has 2 eigenvalues 0 and -1 where one is a real negative value, therefore (0,0) is an unstable saddle point. There are similarly 2 eigenvalues at (2,1) namely $1 + \sqrt{2}$ and $1 - \sqrt{2}$ where one of them is positive and real and the other is negative and real and so, (2,1) is also an unstable saddle point. Unstable saddle points are those point where multiple field lines approach but never meet or only meet asymptotically.

Question 4.4



The above plot shows the field vector magnitude and direction in the given range along with eigenvectors plotted for each equilibria point.

The blue and green quiver plots in the field direction plot represent the eigenvectors of the system at the equilibrium points (0,0) and (2,1). The eigenvectors and their associated eigenvalues scale and give shape to the field direction vectors. We can see the eigenvectors at (0,0) are orthogonal. The eigenvector at y=0 has eigenvalue 0 and hence everything on the line is 0. The eigenvector given by x=0 has eigenvalue -1 therefore it mirrors the arrows on the line about the origin and shrinks them. Similarly, the point (2,1) has 2 eigenvectors. The one pointing to the right has a eigenvalue $1 + \sqrt{2}$ and it expands the arrows in that direction by the same factor and the other one pointing left has eigenvalue $1 - \sqrt{2}$ and so it shrinks the arrows. The eigenvectors at both the points affect the shape of the direction field and the eigenvalues affect the size of the direction vector.

4) 1) Given $x_{t+1} = x_t y_t$

$y_{t+1} = y_t(x_t - 1)$

In equilibrium state,

$X_{t+1} = X_t = X$

and $Y_{t+1} = Y_t = Y$

Plugging these values above,

$X = XY$ — (I)

$Y = Y(X-1)$ — (II)

for $X \neq 0$, from (I) we get $Y = 1$, putting in (II) we get $X = 2$. So equilibrium is $(2,1)$

for $X = 0$, we get $Y = 0$

So 2nd equilibrium is $(0,0)$

2) $J = \begin{pmatrix} \frac{\partial f^1}{\partial x} & \frac{\partial f^1}{\partial y} \\ \frac{\partial f^2}{\partial x} & \frac{\partial f^2}{\partial y} \end{pmatrix} = \begin{pmatrix} y_t & x_t \\ y_t & x_t - 1 \end{pmatrix}$

At $(0,0)$

$J_1 = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$

At $(2,1)$

$J_2 = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}$

3) for J_1, $A - \lambda I = \begin{pmatrix} -\lambda & 0 \\ 0 & -1-\lambda \end{pmatrix}$

$|A - \lambda I| = 0$

$\therefore \lambda = [0, -1]$ .. eigenvalues at $(0,0)$

for J_2, $A - \lambda I = \begin{pmatrix} 1-\lambda & 2 \\ 1 & 1-\lambda \end{pmatrix}$

$|A - \lambda I| = 0$ gives

$\lambda = [1+\sqrt{2}, 1-\sqrt{2}]$

eigenvalues at $(2,1)$

Eigenvectors:

for J_1 at $\lambda = 0$

$\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\therefore y = 0$ and $EV_{(0)} = \begin{bmatrix} x \\ 0 \end{bmatrix}$

for J_1 at $\lambda = -1$

$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\therefore x = 0$ and $EV_{(-1)} = \begin{bmatrix} 0 \\ y \end{bmatrix}$

for J_2 at $\lambda = 1+\sqrt{2}$

$\begin{pmatrix} -\sqrt{2} & 2 \\ 1 & -\sqrt{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\therefore x = 1, y = 1/\sqrt{2}$

$EV_{(1+\sqrt{2})} = \begin{bmatrix} 1 \\ 1/\sqrt{2} \end{bmatrix}$

for J_2 at $\lambda = 1-\sqrt{2}$

$\begin{pmatrix} \sqrt{2} & 2 \\ 1 & \sqrt{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$EV_{1-\sqrt{2}} = \begin{bmatrix} 1 \\ -1/\sqrt{2} \end{bmatrix}$

All eigen vectors checked as $A_i \cdot EV_i = \lambda_i \cdot EV_i$