Assignment 6

Tycho Stam (13303147), Mickey van Riemsdijk (13939432)

December 2, 2023

Problems

Q1

Within the mathematics multiple forms of ordinary differential equations (ODE) are known. To start with a ODE that can be written in the form $g(t)\frac{dx}{dt} = f(x)$ is called a separable equations Hereby separable, indicates that the expression on the right side of the equation can be separated into a function of x and a function of t.

For a linear differential equation, the equation is in the form of a linear polynomial. Hereby the differential equation leads to an expression in which the variable and its derivatives appear separately and not as a power of exponent different from 1, or as a product of each other. For example $\frac{dx}{dy} + 4x = 4$. Whereby the non-linear differential equations the unknown function and its derivatives doesn't produce a straight line when plotted on the graph. For example $\frac{dy}{dx} = y^2 + x$.

The order of the ODE is based on the highest order of the derivatives within the differential equation. For a second order a formula like $y'' + e^t y' = -y$, is an example. A first order formula would look like $y' - y - 1 = 0$.

Q2

The difference between explicit and implicit solutions are within the separation of the variables. An explicit solution is one in which the dependent variable can be expressed in terms of the independent variable(s) without involving any implicit relationships or additional parameters. The equation $x + 2y = 0$ is a explicit solution because it can be written in $y = -\frac{x}{2}$.

An ODE has fixed points when the measured outputs don't change over time. This can be found when the derivative is 0. It can be stable if the system conditions goes back to the original condition and unstable when The system quickly diverges from those conditions.

Q3

Formula 1. With $\frac{dx}{dt} = 1$ with $x(0) = 0$
Formula 2. With $\frac{dx}{dt} = 2t$ with $x(0) = -4$
Formula 3. With $\frac{dx}{dt} = -x$ with $x(0) = 4$

Analytical solution, formula 1. $x(t) = t$
Analytical solution, formula 2. $x(t) = t^2 - 4$
Analytical solution, formula 3. $x(t) = 4e^{-t}$

```python
# Formula's
def f1(x, t):
    return 1

def f2(x, t):
    return 2 * t

def f3(x, t):
    return -x
```

```python
# Analytical solutions
def f1_sol(x, t):
    return t

def f2_sol(x, t):
    return -2 * t**2 - 4

def f3_sol(x, t):
    return 4 * np.exp(-t)
```

Euler Implementation

```python
def euler_algo(function, x0, t):
    x = np.zeros(len(t))
    x[0] = x0

    for i, (first, second) in enumerate(zip(t, t[1:])):
        dt = second - first
        x[i+1] = x[i] + function(x[i], first) * dt

    return x
```

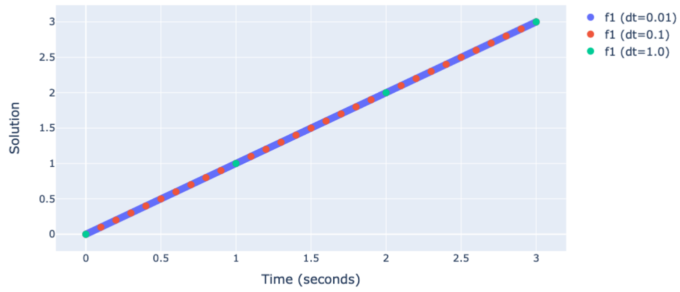

Figure 1: Euler Algorithm Results for Different ODEs and Time Steps for Formula 1

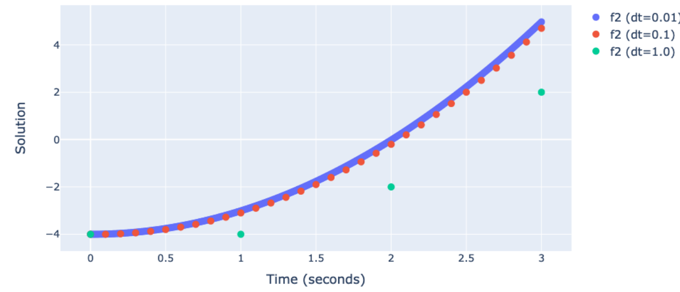

Figure 2: Euler Algorithm Results for Different ODEs and Time Steps for Formula 2

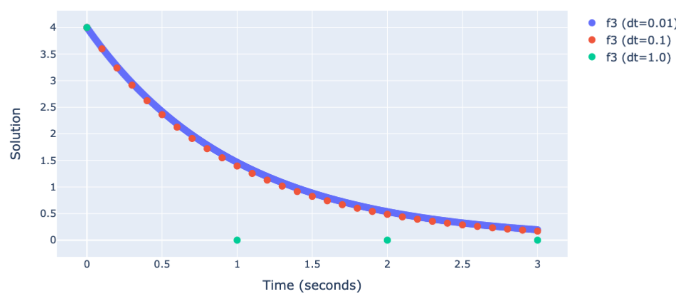

Figure 3: Euler Algorithm Results for Different ODEs and Time Steps for Formula 3

When comparing Figures 1, 2 and 3 the Euler algorithm works the best with linear equations. When the equations are not linear, step size becomes more important for the accuracy.

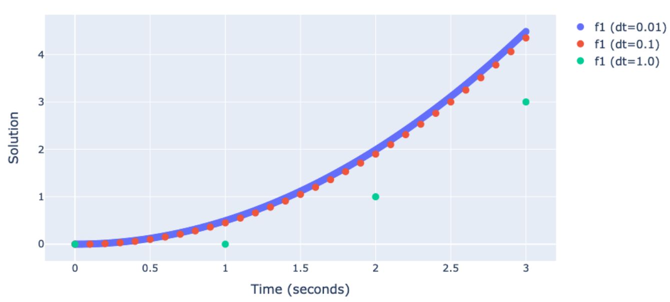

Figure 4: Euler Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 1

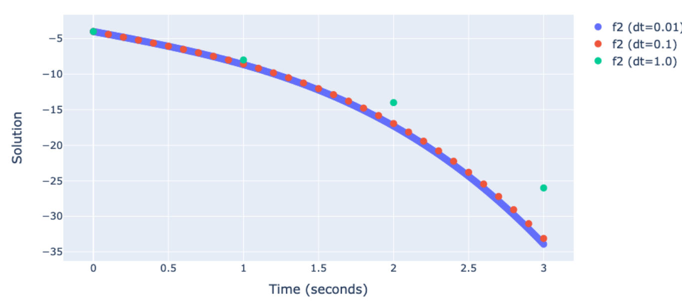

Figure 5: Euler Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 2

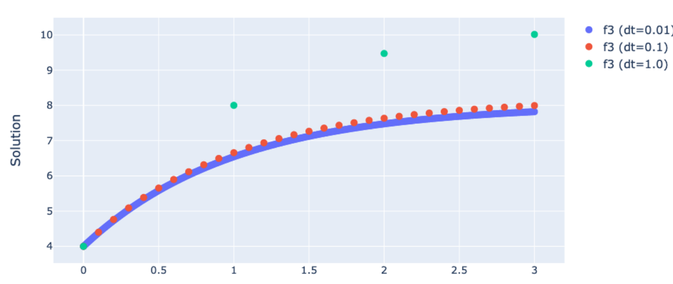

Figure 6: Euler Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 3

When comparing the analytical solutions in Figures 4, 5 and 6, the results are similar to the previous conclusion where smaller step sizes increase the accuracy.

Bonus: Runge Kutta Implementation

```
def runge_kutta(function, x0, t):
    x = np.zeros(len(t))
    x[0] = x0

    for i, (first, second) in enumerate(zip(t, t[1:])):
        h = second - first
        k_1 = h * function(x[i], first)
        k_2 = h * function(x[i] + 0.5 * k_1, first + 0.5 * h)
        k_3 = h * function(x[i] + 0.5 * k_2, first + 0.5 * h)
        k_4 = h * function(x[i] + k_3, second)
        x[i+1] = x[i] + (k_1 + 2 * k_2 + 2 * k_3 + k_4) / 6

    return x
```

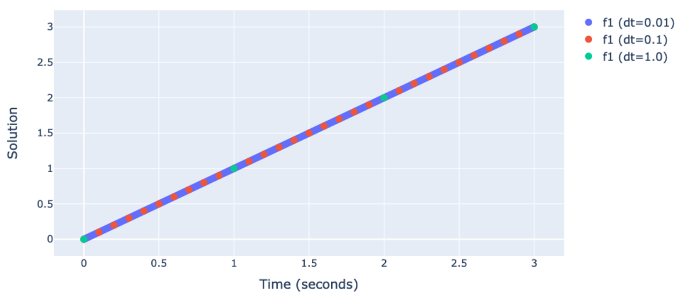

Figure 7: Runge Kutta Algorithm Results for Different ODEs and Time Steps for Formula 1

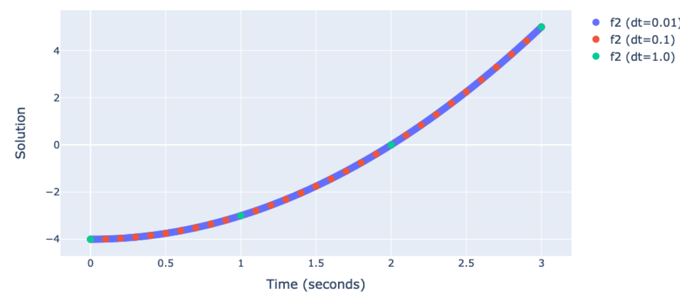

Figure 8: Runge Kutta Algorithm Results for Different ODEs and Time Steps for Formula 2

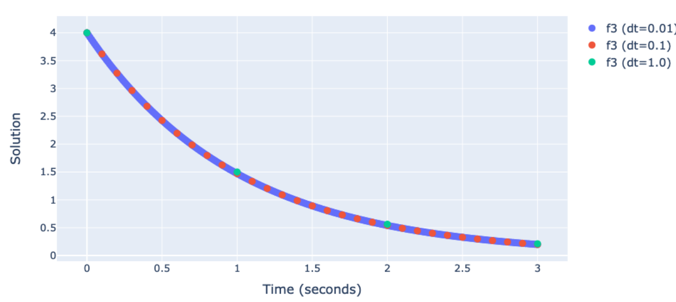

Figure 9: Runge Kutta Algorithm Results for Different ODEs and Time Steps Formula 3

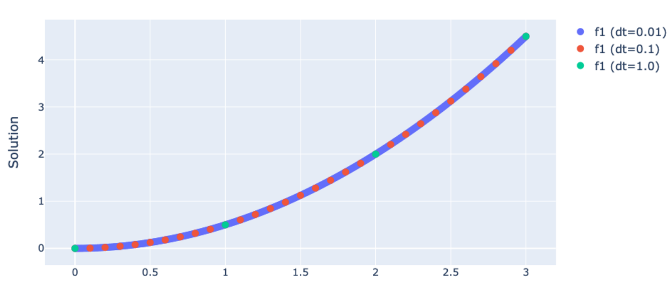

Figure 10: Runge Kutta Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 1

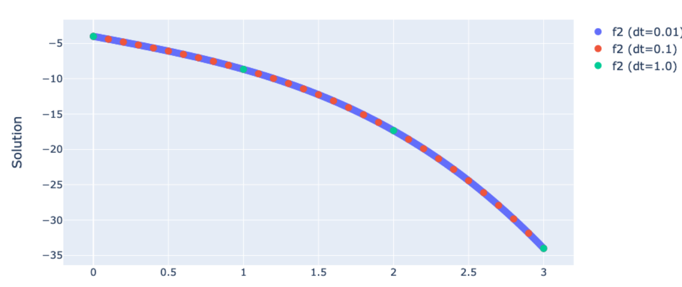

Figure 11: Runge Kutta Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 2

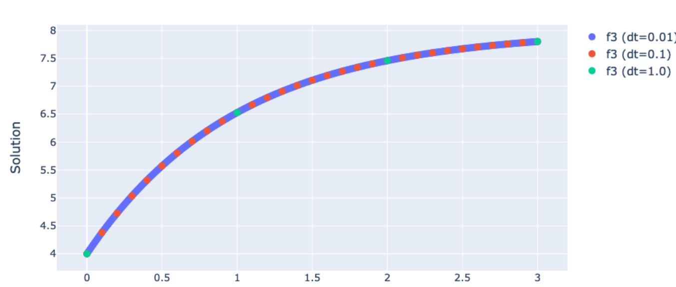

Figure 12: Runge Kutta Algorithm Results for Different ODEs and Time Steps for Analytical Solution of Formula 3

When comparing the Figures 7, 8, 9, 10, 11 and 12 to Figures 1, 2, 3, 4, 5 and 6, the Runge Kutta algorithm results in higher accuracy than the Euler algorithm even when using larger step sizes.

Q4

A

Based on the equation $\frac{dx}{dt} = g - kx$, I assume that g is the constant rate of which protein molecules are generated. Hereby is k the constant rate at which the proteins degrade.

B

Analytical solution with steps:

$$\frac{dx}{dt} = g - kx$$
$$dx = (g - kx)dt$$
$$\frac{1}{g - kx}dx = 1dt$$
$$\int \frac{1}{g - kx}dx = \int 1dt$$
$$-\frac{1}{k}\ln(g - kx) = t + C$$
$$\ln(g - kx) = -kt + C$$
$$e^{\ln(g-kx)} = e^{-kt+C}$$
$$g - kx = e^{-kt+C}$$
$$-kx = e^{-kt+C} - g$$
$$x = -\frac{1}{k}e^{-kt+C} + \frac{g}{k}$$
$$x = \frac{g}{k}(1 - \frac{1}{g}e^{-kt+C})$$
$$x = \frac{g}{k}(1 - \frac{1}{g}e^{-kt} * C_0)$$

C

The role of this constant is to slow the rate of which the proteins degrade over time. It accounts for the flexibility in the solution space and is determined by the initial conditions of the problem.

D

Analytical solution with steps: For $x(0) = 0$:

$$0 = \frac{g}{k}(1 - \frac{1}{g}e^{-k0} * C_0)$$
$$0 = \frac{g}{k}(1 - \frac{1}{g} * C_0)$$
$$0 = \frac{g}{k} - \frac{1}{k} * C_0$$
$$-\frac{g}{k} = -\frac{1}{k} * C_0$$
$$\frac{g}{k} = \frac{1}{k} * C_0$$
$$C_0 = g$$

E

As depicted in Figure 13, the variable g extends the formula over time. A higher value of g leads to the stabilization of the value. This pattern is also observed with a higher value of k, where a lower k results in a faster decline in the graph.

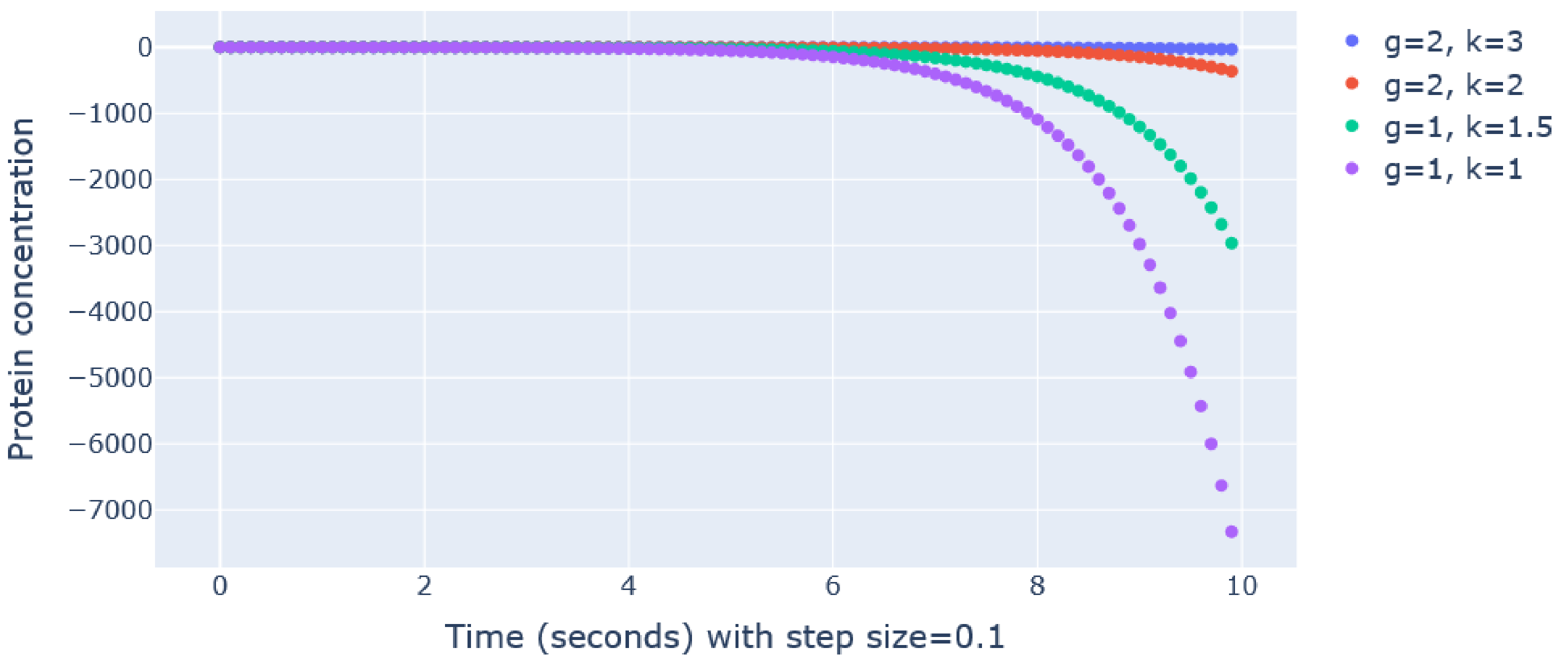

Figure 13: Concentration of a specific protein over time in a biological cell.

F

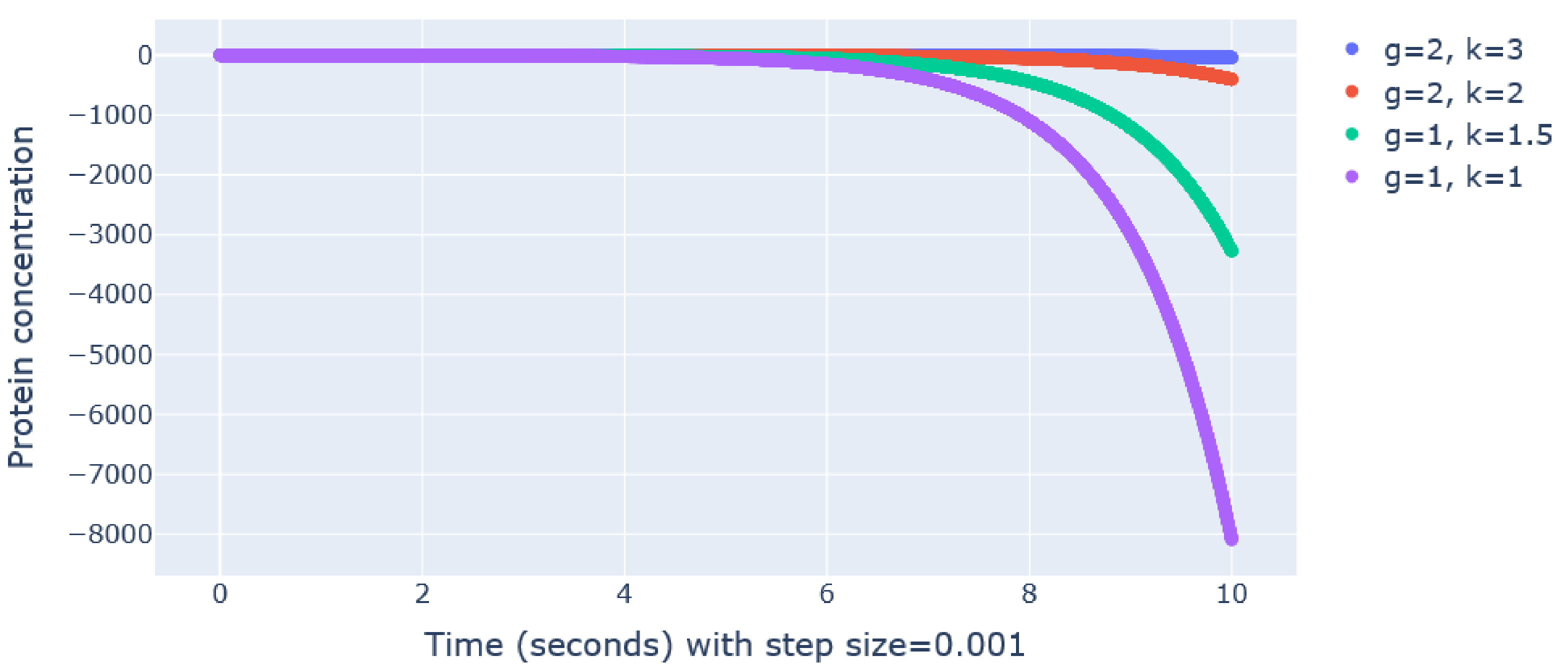

Figure 14: Concentration of a specific protein over time in a biological cell.

G

Analytical solution with steps: For $x'(t) = 0$:

$$\frac{d}{dt}\frac{1}{k}(g - e^{-kt} * C_0)$$

$$\frac{1}{k}(C_0 * -e^{-kt}\frac{d}{dt} + g\frac{d}{dt})$$

$$\frac{1}{k}(C_0 * -e^{-kt}\frac{d}{dt})$$

$$\frac{1}{k}(C_0 * ke^{-kt})$$

$$(C_0 * e^{-kt})$$

H

As depicted in Figure 15, the function exhibits a global point of stability around $t = 4$. This is seen by the fact that the value of $x(t)$ tends towards equilibrium around 0 as t approaches ∞.

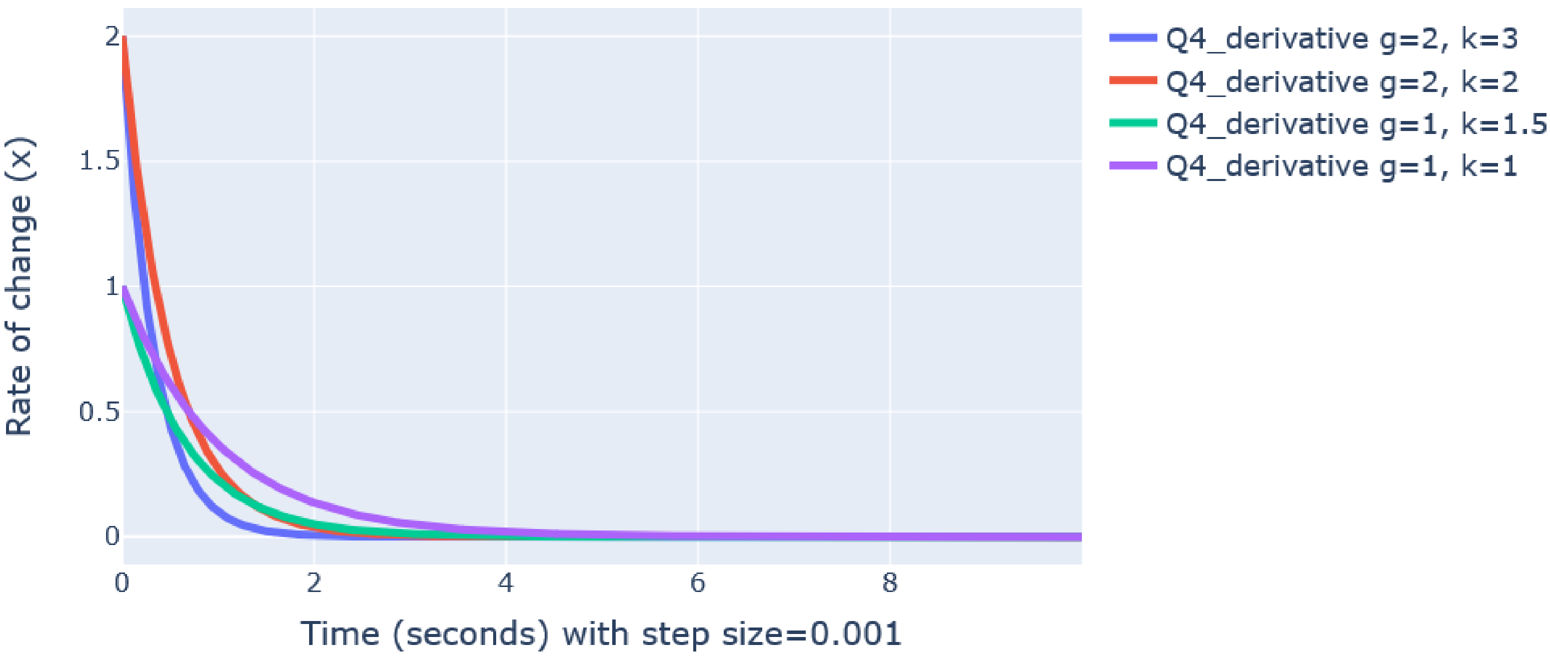

Figure 15: The rate of change for x'(t).

I

Based on the assumption that a cell doesn't want to produce more proteins than needed, a possible graph could depict an exponential increase in g, demonstrating a proactive response to reach a new stable concentration. Hereby the concentration will eventualy flatten over the x-as. In this way the cell needs to create protein at the same rate as the proteins are decaying. A possible graph of this is seen in Figure 16.

An alternative scenario involves an initial surplus of protein production, surpassing the cell's immediate requirements, to create a backup supply. Subsequently, the rate of protein production, slows to conserve energy. This cycle goes on until the cell undergoes decay as seen in Figure 17.

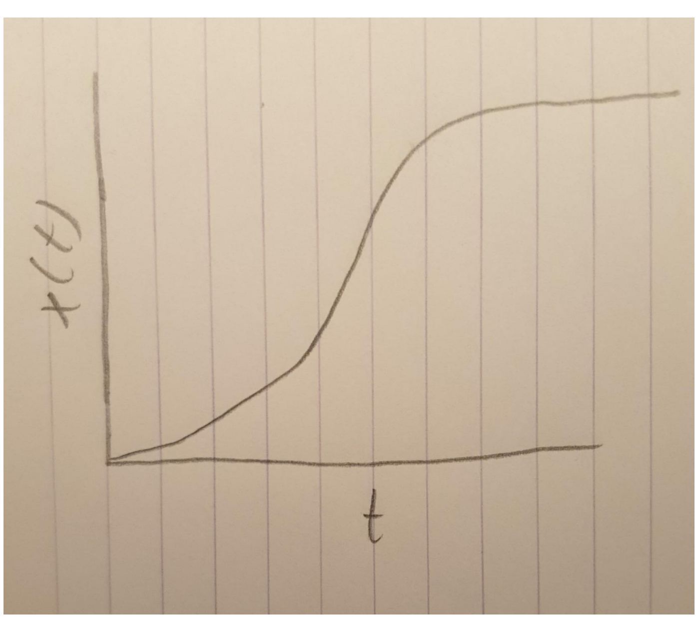

Figure 16: Assumption that a cell protein increases in the beginning and after that stabilizes.

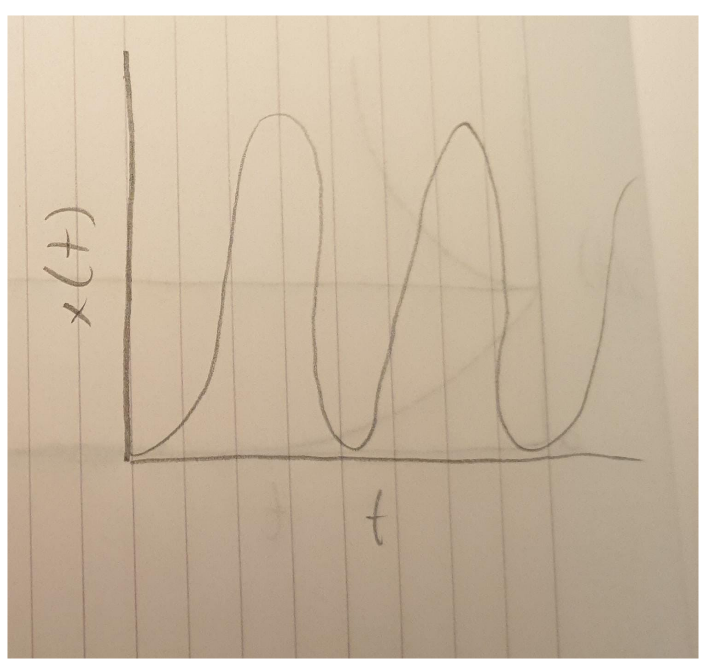

Figure 17: Assumption that a cell protein output changes periodically.

J

'Mean-field' simplifies complex interactions within a system by assuming that each component experiences an average or mean influence from all other components, facilitating a more manageable analysis of the system's behaviour. Hereby the protein model makes simplifying assumptions about the system, assuming that the protein concentration can be represented by an average or mean interaction term. By adopting a mean-field approach, the model can capture the overall trends and behavior of the system without the computational complexity of explicitly modeling the proteins. It allows for a more manageable and analytically tractable representation of the system's dynamics.

Q5

A

In this formula, r represents the rate of reproduction at any given time, while k denotes the rate of dying at any given time. The variable x serves as the expression that captures the occurrence of both reproduction and dying over time. By subtracting the reproduction rate and the dying rate from each other, we obtain a representation of the total population.

B

Analytical solution with steps:

$$\frac{dx}{dt} = rx - kx$$
$$dx = rx - kx\,dt$$
$$\frac{1}{x}dx = r - k\,dt$$
$$\int \frac{1}{x}dx = \int r - k\,dt$$
$$\ln(x) = rt - kt + C$$
$$e^{\ln(x)} = e^{rt-kt} * C_0$$
$$x = e^{rt-kt} * C_0$$
$$x = e^{t(r-k)} * C_0$$

C

The graph, illustrated in Figure 18, demonstrates three distinct behaviors corresponding to various inputs. In the case of the blue line, where the dying rate surpasses the reproduction rate, the graph stabilizes at 0. Conversely, for the red line with a higher reproduction rate, the graph exhibits exponential growth. When both rates are equal, the population stabilizes at the initial population level.

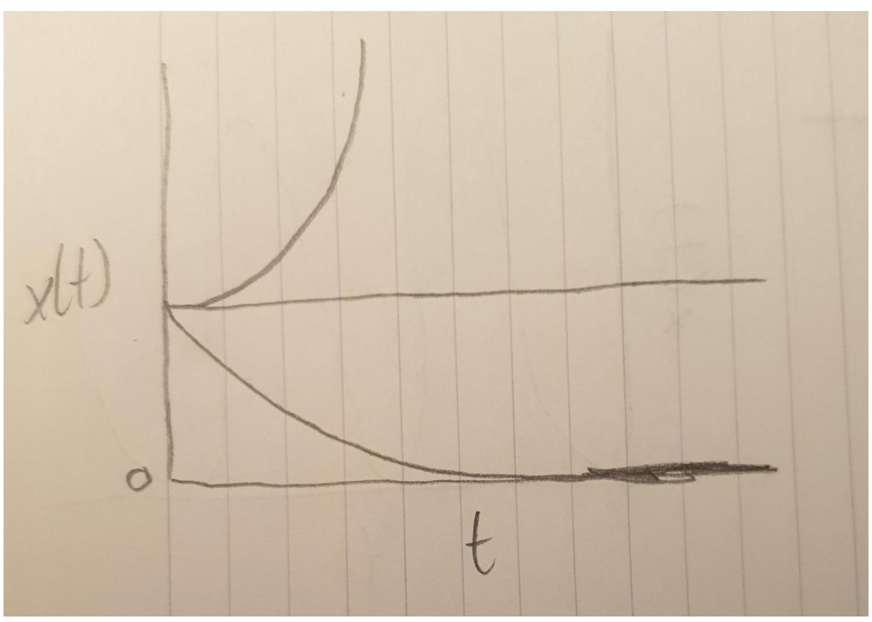

Figure 18: Possible behaviors of the rabbit population.

D

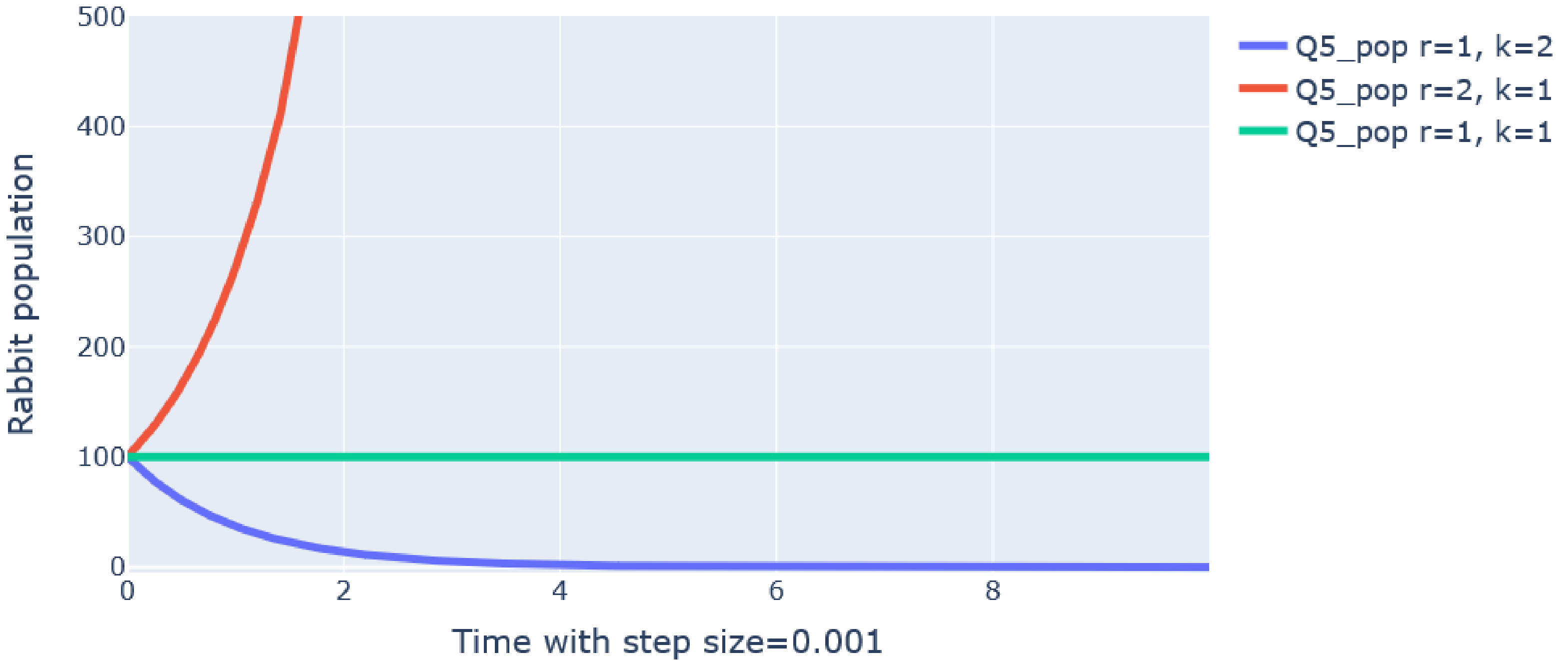

Figure 19: Possible behaviors of the rabbit population, numerical.

E

When r is increased, the exponential growth of the population remains largely unchanged. Although the graph may undergo slight alterations, the overall pattern remains consistent. In contrast, when k is larger, the graph still converges to 0, with the rate of convergence dependent on the values of k and r. When k and r are comparable, the graph exhibits fluctuations marked by both increases and decreases.

F

$$\frac{dx}{dt} = rx^2 - kx$$

The expression rx^2 indicates that the reproductive rate is directly proportional to the square of the current population size. The modification in the equation reflects the idea that the reproductive success of rabbits is not solely determined by intrinsic factors but also depends on the population density, as more rabbits mean that r will be higher.

G

Analytical solution with steps:

$$dx = rx^2 - kx\ dt$$
$$\frac{1}{rx^2 - kx}dx = dt$$
$$\frac{1}{x(rx-k)}dx = dt$$
$$\int \frac{1}{x(rx-k)}dx = \int dt$$
$$-\frac{1}{k}\ln(\frac{rx-k}{x}) + C = t$$
$$-\frac{1}{k}\ln(\frac{rx-k}{x}) = t - C$$
$$\frac{1}{k}\ln(\frac{rx-k}{x}) = -t + C$$
$$\ln(\frac{rx-k}{x}) = -tk + Ck$$
$$e^{\ln(\frac{rx-k}{x})} = e^{-tk+Ck}$$
$$\frac{rx-k}{x} = e^{-tk+Ck}$$
$$rx - k = xe^{-tk+Ck}$$
$$rx - xe^{-tk+Ck} = k$$
$$x(r - e^{-tk+Ck}) = k$$
$$x = \frac{k}{r - e^{-tk+Ck}}$$

For x(t) = 0:

$$\frac{k}{r - e^{Ck}} = x_0$$

H

```python
# Analytical solutions
def Q5_pop_squared(k, r, t, C):
    denominator = (r - np.exp(-t * k + k * C))
    if denominator == 0:
        return 0
    else:
        return k / denominator

# Version 2 is based on the solition from Wolfram Alpha
def Q5_pop_squared_v2(k,r,t,C):
    denominator = (r * np.exp(C * k) + np.exp(k*t))
    if denominator == 0:
        return 0
    else:
        return (k * np.exp(C * k)) / denominator
```

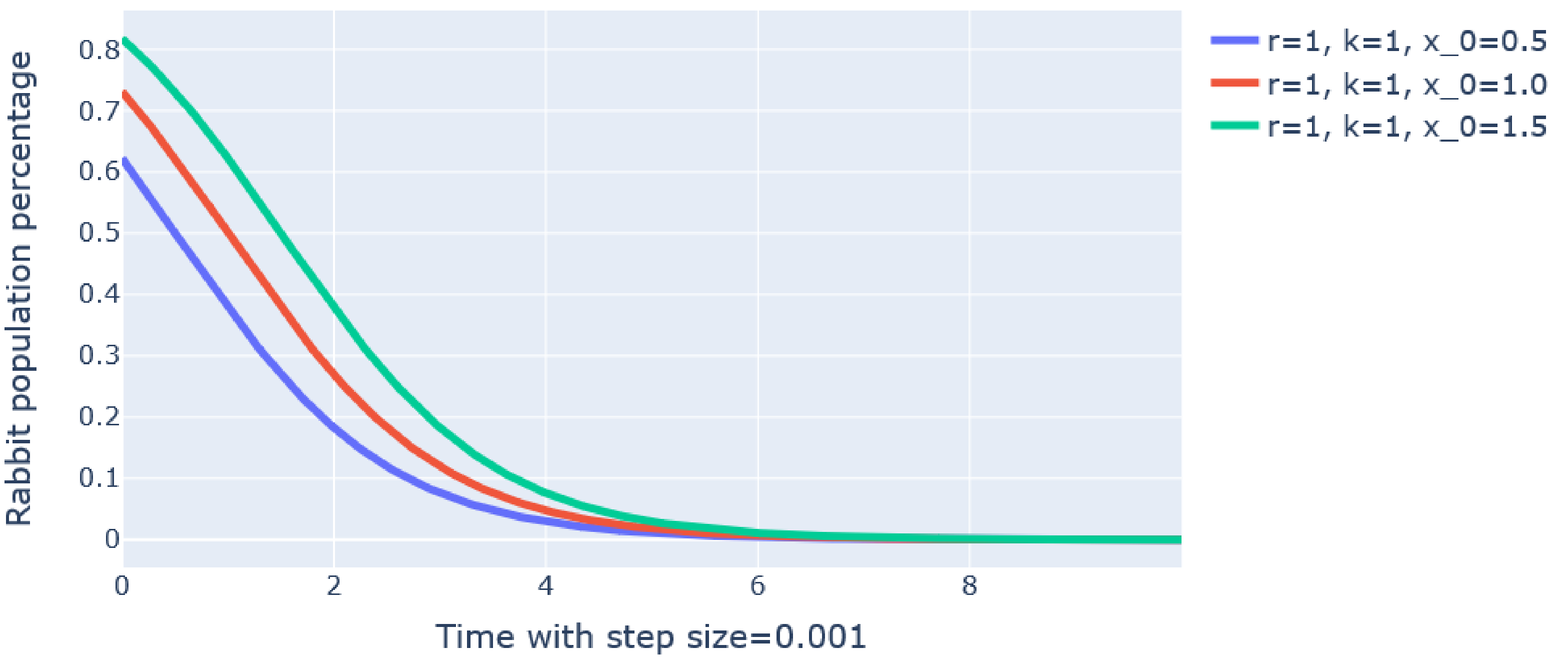

Figure 20: Numerical solution for rabbit population with density growth factor.

I

Analytical solution with steps:

$$\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right)$$

$$dx = x\left(1 - \frac{x}{x_{max}}\right)dt$$

$$\frac{1}{x\left(1 - \frac{x}{x_{max}}\right)} = \frac{1}{x_{max}}\left(\frac{1}{1 - \frac{x}{x_{max}}}\right)$$

$$\int \frac{1}{1 - \frac{x}{x_{max}}}\,dx = x_{max}\int dt$$

$$-x_{max}\ln|1 - \frac{x}{x_{max}}| = x_{max}t + C$$

$$1 - \frac{x}{x_{max}} = e^{-\frac{t}{x_{max}}}$$

$$x = x_{max}(1 - e^{-\frac{t}{x_{max}}})$$

```python
# Formula's
def Q5_v1(x, t):
    x_max = 10
    return x * (1 - (x/x_max))

def Q5_v2(x, t):
    x_max = 1000
    return x * (1 - x/x_max)
```

```python
# Second Derivative
def second_derivative(x, t):
    x_max = 10
    return 1 - 2 * x / x_max
```

```python
# Analytical solutions
def Q5_v1_sol(x, t):
    x_max = 10
    return x_max * (1 - np.exp(-t / x_max))
```

```python
def Q5_v2_sol(x, t):
    x_max = 1000
    return x_max * (1 - np.exp(-t / x_max))
```

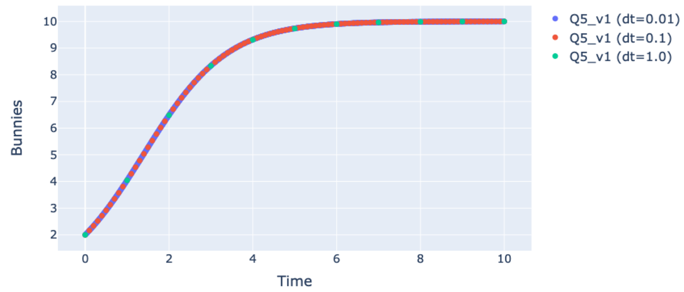

Figure 21: Runge Kutta Algorithm Results for Formula Q5_v1 at different time steps, with x_max = 10 and initial $x_0 = 2$.

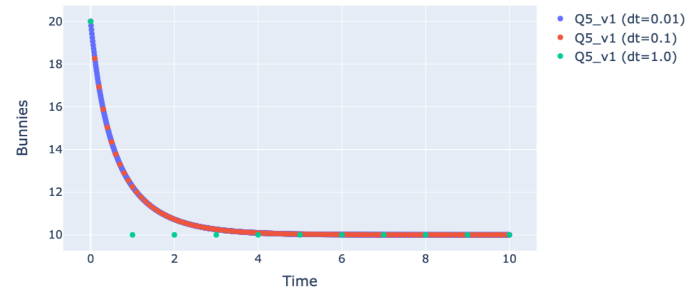

Figure 22: Runge Kutta Algorithm Results for Formula Q5_v1 at different time steps, with x_max = 10 and initial $x_0 = 20$.

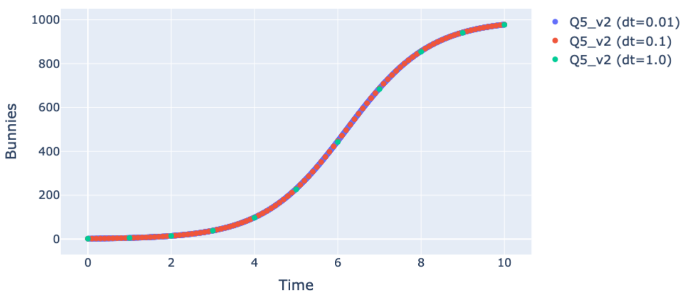

Figure 23: Runge Kutta Algorithm Results for Formula Q5_v2 at different time steps, with x_max = 1000 and initial $x_0 = 2$.

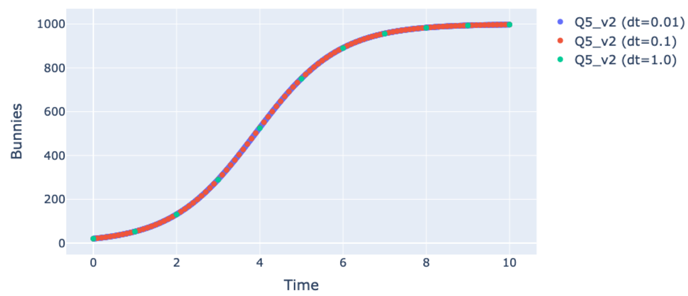

Figure 24: Runge Kutta Algorithm Results for Formula Q5_v2 at different time steps, with x_max = 1000 and initial $x_0 = 20$.

For the Figures 21, 22, 23, 24 and 25, the $dt = 0.1$ and $dt = 0.01$ are almost identical if the markers are not clear in the Figures. When comparing Figures 21 and 22, both show that the equation lead to the eventual value of $x == x_{max}$. It does not matter if the initial x value starts higher or lower than the x_{max}. However, Figures 23 and 24 show that when starting with an initial x value closer to the x_{max} the x value will reach a constant value of x_{max} earlier.

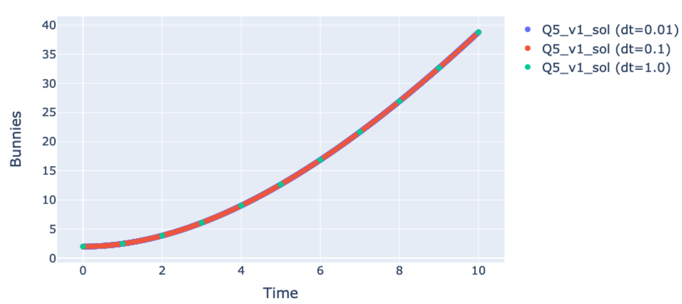

Figure 25: Runge Kutta Algorithm Results for Formula Q5_v1 analytical solution at different time steps, with x_max = 10 and initial $x_0 = 2$.

Figure 25 shows the markers for the analytical solution of Q5_v1. For all previously used initial values in this question, the plot shows the same pattern.

Question: What is the biological meaning of x_max?
Answer: The literal biological meaning of x_max is called 'Carrying capacity'. Carrying capacity is the maximum population size an environment can handle with the available resources in that environment (Buckley, 2021).

J

According to Figure 21 compared to Figure 22 when a carrying capacity limit is set growth is not only limited to the x_{max} value, but also when the initial value is higher than the maximum value the population will decrease to the same maximum value. An interpretation of this behaviour is, when it is possible for the maximum value to change, just like in real life, the population can decrease due to natural external factors. When comparing Figure 23 and Figure 24 initial population dictates the speed of reaching the maximum population value, but not the eventual population size. When comparing this equation to question 5F, 5F doesn't have an upper limit for the maximum amount of bunnies even when the reproduction is low. With enough time, the bunnies can become infinite in numbers.

K

Based on the equation $\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right)$, there are two fixed points in the model. $x_0 = 0$, because the model will not change due to the initial x being 0. The next x will be 0, because in plain terms, there are no bunnies to multiply. The second fixed point in the model is $x = x_{max}$. When $x = x_{max}$ the result will indefinitely be zero and maintain a fixed point in the equation.

Second Derivative:

$$\frac{d^2x}{dt^2} = 1 - \frac{2x}{x_{max}}$$

1. At $x = 0$:

$$\frac{d^2x}{dt^2} = 1$$

So the second derivative is positive, $x = 0$ is a unstable fixed point.

2. At $x = x_{max}$:

$$\frac{d^2x}{dt^2} = -1$$

So the second derivative is negative, $x = x_{max}$ is a stable fixed point.

L

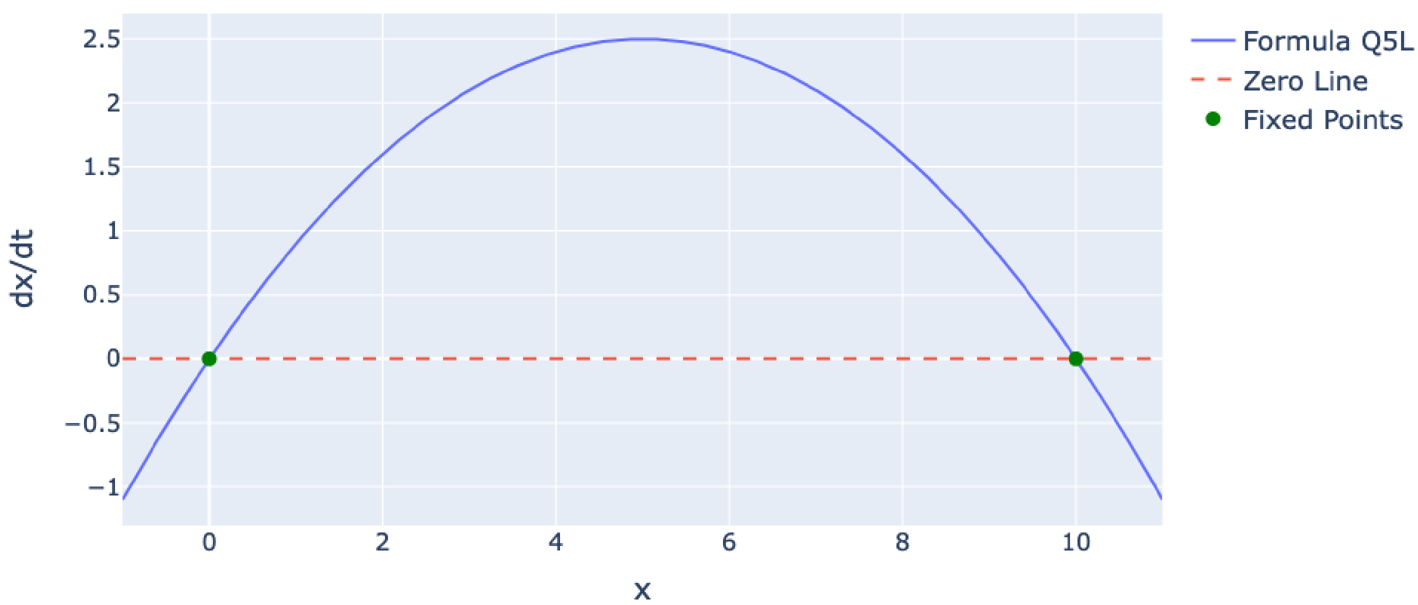

Figure 26: Fixed Points Analysis with $x_{max} = 10$ and Equation $\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right)$

Figure 26, shows that there are two intersection points on the x-axis with the formula $\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right)$. Both intersection points are fixed points where the derivative is equal to zero. Just as shown in question 5K, the fixed points are $x = 0$ and $x = x_{max}$.

M

Altered Formula with Death Rate (r): $\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right) - r$

```
# Formula
def Q5_with_r(x, t, x_max=1000, r = 10):
    return x * (1 - (x/x_max)) - r
```

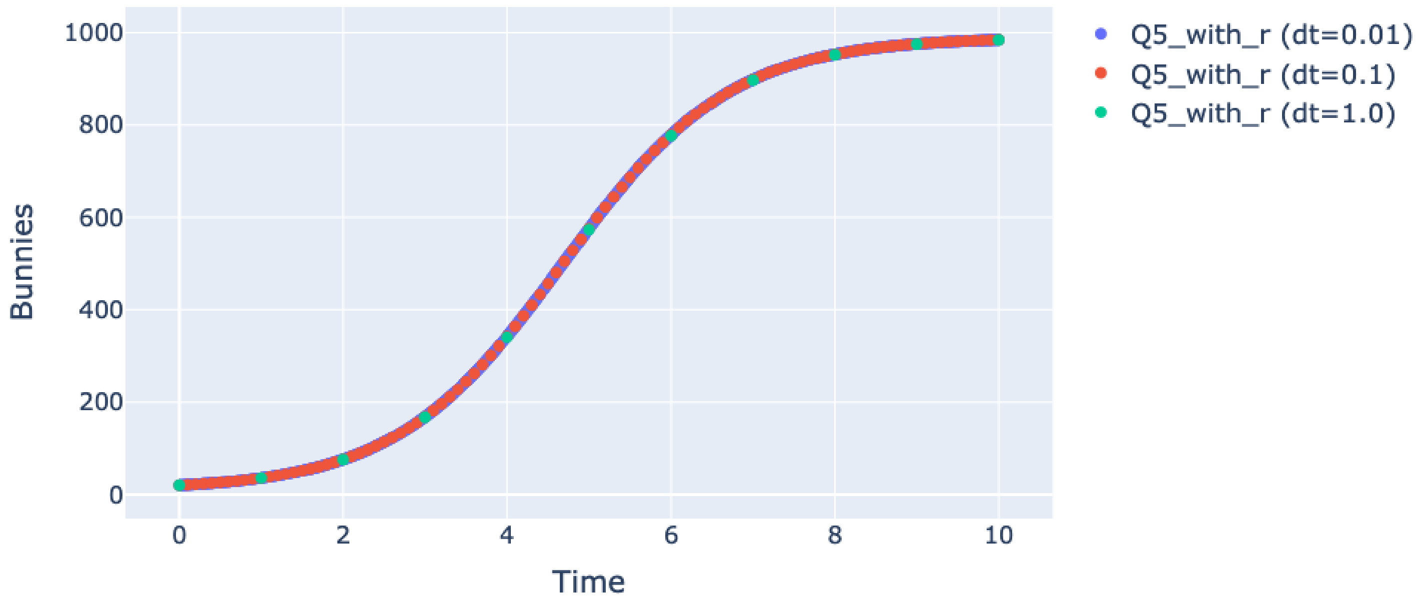

Figure 27: Runge Kutta Algorithm Results for Formula 'Q5_with_r' at different time steps, with $x_{max} = 1000$, $r = 10$ and initial $x_0 = 20$

Figure 27, shows that compared to the previous Figure 24 (with the same initial values) the markers make a similar pattern, but the maximum amount of bunnies is reached later and in the end the constant amount of bunnies is lower when a death rate (r) is added to the equation.

N

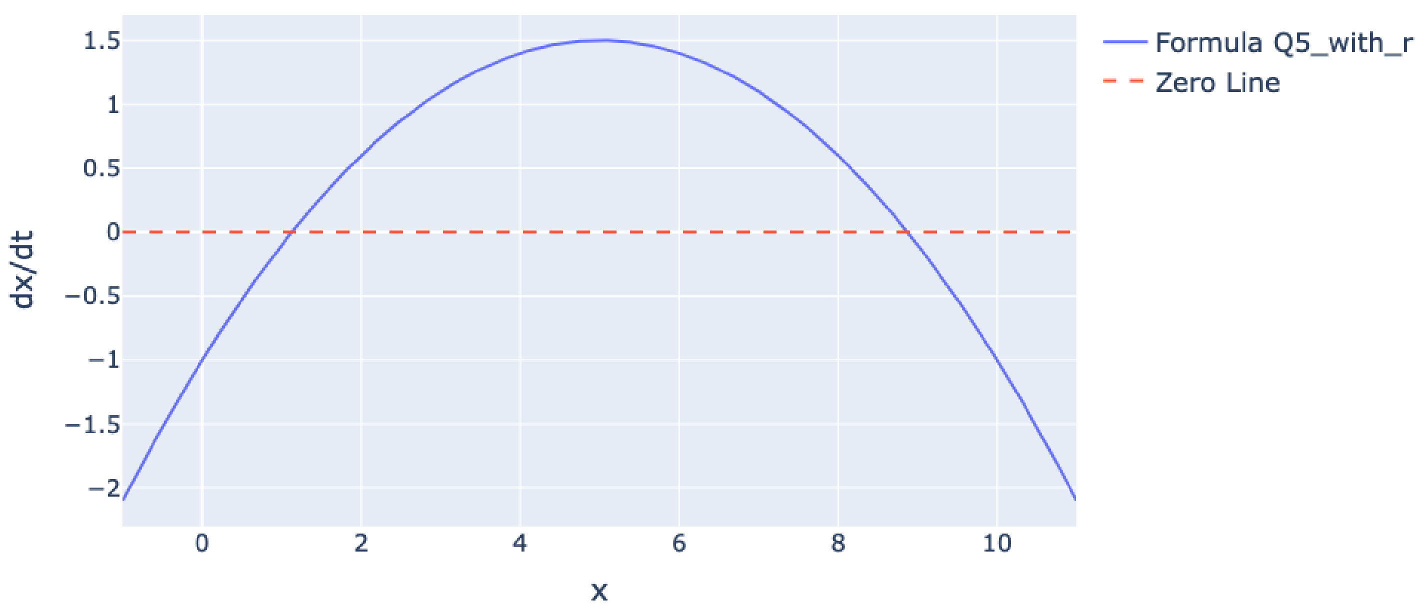

Figure 28: Fixed points for Formula 'Q5_with_r' at different initial x_0, with $x_{max} = 10$ and $r = 1$

Figure 28, shows the equation: $\frac{dx}{dt} = x\left(1 - \frac{x}{x_{max}}\right) - r$, compared to the zero line. The intersection points are also the fixed points. For values $x_{max} = 10$ and $r = 1$ the fixed points are around 1 and 9. From this you can estimate the fixed points are roughly $x = r$ and $x = x_{max} - r$. From question 5K, we can infer that $x = x_{max} - r$ is the only stable fixed point from this equation. This means to get a fixed point the value of r can not exceed the value of x_{max} otherwise there will form an exponential negative amount of bunnies.

O

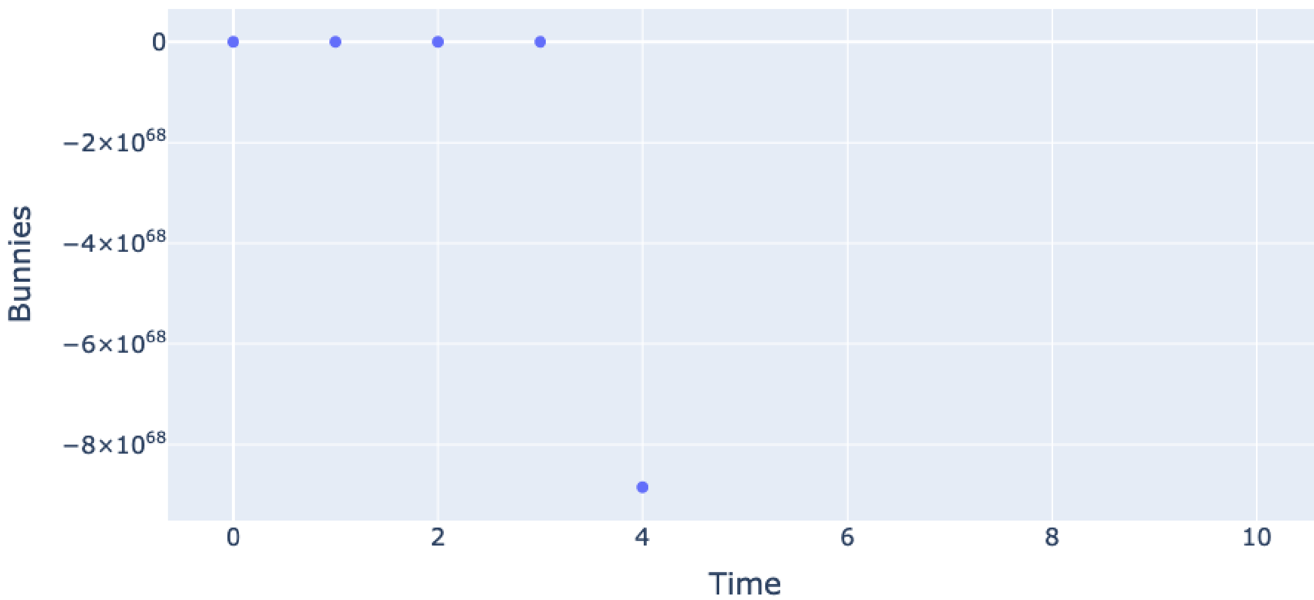

Figure 29: Runge Kutta Algorithm Results for Formula 'Q5_with_r' at different time steps, with $dt = 1.0$ $x_{max} = 100$, $r = 40$ and initial $x_0 = 20$

Figure 29, shows what happens when more bunnies die than multiply. Of course, there can no such thing in real life as a negative amount of bunnies, but in this equation it is possible. The Figure 29 illustrates the exponential negative decline of the amount of bunnies with $dt = 1.0$, $r = 40$ and initial $x_0 = 20$. From $t = 5$ there is a negative infinite amount of dead bunnies. It looks like the first steps from $t = 1$ to $t = 3$ the bunny amount is zero, but the negative amount step from three to four is so big the previous steps did not fit in the plot.

References

Buckley, G. (2021). Carrying capacity.