Mathematical Modelling
Assignment 01

Name	**Student Number**	**contribution**
Gaurang Rowji	2552241	Modelling
Mahrulnisa Mehmood	2550587	Stability Analysis
Zahra Kotwal	2628128	Discussion, conclusion
Amaan Hanslod	2541305	Numerical solution, Discussion

1 Mathematical model

This model consists of three sub-populations namely the susceptible population denoted as X, the HIV-subtype-1-infected population denoted as Y_1 and the HIV-subtype-2-infected population denoted as Y_2. We then denote our entire population as $N = X + Y_1 + Y_2$. In this model our assumption is that individuals infected by one subtype do not become infected by another subtype later.

1.1 Susceptible population, X

New individuals are brought into the susceptible at the rate Π per year thus we have;

$$\frac{dX}{dt} = \Pi$$

One way to leave this population is through the natural cessation of sexual activity, this happens at a constant rate μ and we then obtain;

$$\frac{dX}{dt} = \Pi - \mu X$$

Another way to leave this population is due to being infected by Subtype-1, we start of with $\frac{Y_1}{N}$ which simply represents the number of Subtype-1-HIV infected individuals over the total population, it can be viewed as the probability of being infected by Subtype-1-HIV, next we have to consider what are the chances of a member or the susceptible population coming into sexual contact with the Subtype-1-infected population we denote this as $\frac{Y_1}{N}c$ the rate this event happens is β_1, putting everything together we have;

$$\frac{dX}{dt} = \Pi - \mu X - \frac{Y_1}{N}c\beta_1 X$$

Another way to leave this population is due to being infected by Subtype-2, similarly as above we have that fraction of Subtype-2 infected individuals is given as $\frac{Y_2}{N}$, next we have to consider what are the chances of a member or the susceptible population coming into sexual contact with the Subtype-2-infected population we denote this as $\frac{Y_2}{N}c$ the rate this event happens is β_2, putting everything together we have;

$$\frac{dX}{dt} = \Pi - \mu X - \frac{Y_1}{N}c\beta_1 X - \frac{Y_2}{N}c\beta_2 X$$

We also assume that the time is greater than the initial time and at the initial time we have a solution for X we thus obtain the final differential equation as;

$$\frac{dX}{dt} = \Pi - \mu X - \frac{1}{N}\beta_1 cXY_1 - \frac{1}{N}\beta_2 cXY_2, \quad t > t_0, \quad X(t_0) = X^0 \tag{1}$$

1.2 Subtype-1-infected population, Y_1

The first way to enter this population is from the susceptible population, we would use the exact term we used in the Susceptible population for Y_1;

$$\frac{dY_1}{dt} = \frac{Y_1}{N}c\beta_1 X$$

One way to leave this population is through the natural cessation of sexual activity, this happens at a constant rate μ and we then obtain;

$$\frac{dY_1}{dt} = \frac{Y_1}{N} c\beta_1 X - \mu Y_1$$

Another way to leave this population is through disease-induced death, this happens at a constant rate γ_1 we then obtain;

$$\frac{dY_1}{dt} = \frac{Y_1}{N} c\beta_1 X - \mu Y_1 - \gamma_1 Y_1$$

Another way to leave this population is by the administration of therapy and this happens at a constant rate τ we then obtain;

$$\frac{dY_1}{dt} = \frac{Y_1}{N} c\beta_1 X - \mu Y_1 - \gamma_1 Y_1 - \tau Y_1$$

We also assume that the time is greater than the initial time and at the initial time we have a solution for Y_1 we thus obtain the final differential equation as;

$$\frac{dY_1}{dt} = \frac{1}{N}\beta_1 X Y_1 - (\mu + \gamma_1 + \tau) Y_1, \quad t > t_0, \quad Y_1(t_0) = Y_1^0 \tag{2}$$

1.3 Subtype-2-infected population, Y_2

The first way to enter this population is from the susceptible population, we would use the exact term we used in the Susceptible population for Y_2;

$$\frac{dY_2}{dt} = \frac{Y_2}{N} c\beta_2 X$$

One way to leave this population is through the natural cessation of sexual activity, this happens at a constant rate μ and we then obtain;

$$\frac{dY_2}{dt} = \frac{Y_2}{N} c\beta_2 X - \mu Y_2$$

Another way to leave this population is through disease-induced death, this happens at a constant rate γ_1 we then obtain;

$$\frac{dY_2}{dt} = \frac{Y_2}{N} c\beta_2 X - \mu Y_2 - \gamma_2 Y_2$$

Another way to leave this population is by the administration of therapy and this happens at a constant rate τ we then obtain;

$$\frac{dY_2}{dt} = \frac{Y_2}{N} c\beta_2 X - \mu Y_2 - \gamma_2 Y_2 - \tau Y_2$$

We also assume that the time is greater than the initial time and at the initial time we have a solution for Y_1 we thus obtain the final differential equation as;

$$\frac{dY_2}{dt} = \frac{1}{N}\beta_2 X Y_2 - (\mu + \gamma_2 + \tau) Y_2, \quad t > t_0, \quad Y_2(t_0) = Y_2^0 \tag{3}$$

. We see that we have obtained a system of non-linear IVP problem given as;

$$
\begin{aligned}
\frac{dX}{dt} &= g_1 = \Pi - \mu X - \frac{1}{N}\beta_1 cXY_1 - \frac{1}{N}\beta_2 cXY_2, \quad t > t_0, \quad X(t_0) = X^0 \\
\frac{dY_1}{dt} &= g_2 = \frac{1}{N}\beta_1 XY_1 - (\mu + \gamma_1 + \tau)Y_1, \quad t > t_0, \quad Y_1(t_0) = Y_1^0 \\
\frac{dY_2}{dt} &= g_3 = \frac{1}{N}\beta_2 XY_2 - (\mu + \gamma_2 + \tau)Y_2, \quad t > t_0, \quad Y_2(t_0) = Y_2^0
\end{aligned}
$$

2 Stability Analysis

The steady-states of the initial value problem described in the equations above are determined by setting the time derivatives to zero, resulting in the system being at rest.

1 Disease-Free Equilibrium: We begin our stability analysis by considering the disease-free equilibrium, which represents a trivial critical point. In this state, both subtypes are eradicated, resulting in the absence of both subtype 1-infected population (Y_1) and subtype 2-infected population (Y_2), both reduced to zero.

Let $\frac{dX}{dt} = 0$:

$$0 = \Pi - \mu X - \frac{1}{N}\beta_1 c X Y_1 - \frac{1}{N}\beta_2 c X Y_2$$

$$0 = \Pi - X\left(\mu + \frac{1}{N}\beta_1 c Y_1 + \frac{1}{N}\beta_2 c Y_2\right)$$

$$X\left(\mu + \frac{1}{N}\beta_1 c Y_1 + \frac{1}{N}\beta_2 c Y_2\right) = \Pi$$

Thus, $X^* = \frac{\Pi}{\mu}$, since Y_1 and Y_2 are eradicated.

$$(X^*, Y_1^*, Y_2^*) = (\frac{\Pi}{\mu}, 0, 0)$$

2 Subtype-1-Only Equilibrium: To determine the Subtype-1-only equilibrium, we'll look for a point where only the Subtype-1-infected population (Y_1) is present, while the Subtype-2-infected population (Y_2) is eradicated. This equilibrium signifies the persistence of the endemic Subtype-1 infection without the presence of the invading Subtype-2.

Let $\frac{dX}{dt} = 0$ and $\frac{dY_1}{dt} = 0$

$$0 = \Pi - \mu X - \frac{1}{N}\beta_1 c X Y_1 - \frac{1}{N}\beta_2 c X Y_2 \tag{4}$$

$$0 = \frac{1}{N}\beta_1 c X Y_1 - (\mu + \gamma_1 + \tau) Y_1 \tag{5}$$

From (5) we get:

$$\frac{1}{N}\beta_1 c X Y_1 = (\mu + \gamma_1 + \tau) Y_1$$

Using the above equation and substituting in (4):

$$0 = \Pi - \mu X - (\mu + \gamma_1 + \tau) Y_1 - 0$$

$$X = \frac{\Pi - (\mu + \gamma_1 + \tau)Y_1}{\mu} \tag{6}$$

It's important to note that according to the mathematical model, the total, N, is typically defined as the sum of X, Y_1, and Y_2. However, in this particular scenario, Y_2 is not considered, hence N simplifies to $X + Y_1$.

Using (5):

$$0 = \frac{\beta_1 c X Y_1}{N} - (\mu + \gamma_1 + \tau)Y_1$$

$$\frac{\beta_1 c X Y_1}{X + Y_1} = (\mu + \gamma_1 + \tau)Y_1$$

$$\beta_1 c X Y_1 = (\mu + \gamma_1 + \tau)Y_1(X + Y_1)$$

$$\beta_1 c X Y_1 = (X Y_1 + Y_1^2)(\mu + \gamma_1 + \tau)$$

$$\beta_1 c X Y_1 = X Y_1(\mu + \gamma_1 + \tau) + Y_1^2(\mu + \gamma_1 + \tau)$$

$$\beta_1 c X Y_1 - X Y_1(\mu + \gamma_1 + \tau) = Y_1^2(\mu + \gamma_1 + \tau)$$

$$X(\beta_1 c Y_1 - Y_1(\mu + \gamma_1 + \tau)) = Y_1^2(\mu + \gamma_1 + \tau)$$

$$X = \frac{Y_1^2(\mu + \gamma_1 + \tau)}{(\beta_1 c Y_1 - Y_1(\mu + \gamma_1 + \tau))}$$

$$X = \frac{Y_1(\mu + \gamma_1 + \tau)}{(\beta_1 c - (\mu + \gamma_1 + \tau))}$$

Therefore;

$$X = \frac{Y_1(\mu + \gamma_1 + \tau)}{\beta_1 c - \mu - \gamma_1 - \tau} \tag{7}$$

Equating (6) and (7) we obtain;

$$\frac{Y_1(\mu + \gamma_1 + \tau)}{\beta_1 c - \mu - \gamma_1 - \tau} = \frac{\Pi - (\mu + \gamma_1 + \tau)Y_1}{\mu}$$

$$Y_1(\mu + \gamma_1 + \tau)\mu = [\Pi - (\mu + \gamma_1 + \tau)Y_1](\beta_1 c - \mu - \gamma_1 - \tau)$$

$$Y_1(\mu + \gamma_1 + \tau)\mu = \Pi(\beta_1 c - \mu - \gamma_1 - \tau) - (\mu + \gamma_1 + \tau)Y_1(\beta_1 c - \mu - \gamma_1 - \tau)$$

Dividing the equation by $(\mu + \gamma_1 + \tau)$ we obtain

$$Y_1\mu = \frac{\Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{(\mu + \gamma_1 + \tau)} - Y_1(\beta_1 c - \mu - \gamma_1 - \tau)$$

Simplifying the equation we get;

$$Y_1^* = \frac{\Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{(\mu + \gamma_1 + \tau)(\beta_1 c - \gamma_1 - \tau)} \tag{8}$$

Substituting the outcome of Y_1 into (6) we get;

$$X = \frac{\Pi - \cancel{(\mu + \gamma_1 + \tau)}\frac{\Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{\cancel{(\mu + \gamma_1 + \tau)}(\beta_1 c - \gamma_1 - \tau)}}{\mu} \tag{9}$$

$$X = \frac{\Pi - \frac{\Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{(\beta_1 c - \gamma_1 - \tau)}}{\mu}$$

$$X = \frac{\Pi(\beta_1 c - \gamma_1 - \tau) - \Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{(\beta_1 c - \gamma_1 - \tau)\mu}$$

$$X^* = \frac{\Pi}{(\beta_1 c - \gamma_1 - \tau)}$$

Therefore the equilibirum point is ;

$$(X^*, Y_1^*, Y_2^*) = \left(\frac{\Pi}{(\beta_1 c - \gamma_1 - \tau)}, \frac{\Pi(\beta_1 c - \mu - \gamma_1 - \tau)}{(\mu + \gamma_1 + \tau)(\beta_1 c - \gamma_1 - \tau)}, 0\right)$$

3 Subtype-2-Only Equilibrium: Using the same approach as before, in order to find the equilibrium points ,we will only consider Sub type 2 population while the Sub type-1-infected population Y_1 is eradicated

Let $\frac{dX}{dt} = 0$ and $\frac{dY_2}{dt} = 0$

$$0 = \Pi - \mu X - \frac{1}{N}\beta_1 cXY_1 - \frac{1}{N}\beta_2 cXY_2 \tag{10}$$

$$0 = \frac{1}{N}\beta_2 cXY_2 - (\mu + \gamma_2 + \tau)Y_2 \tag{11}$$

From (11) we get:

$$\frac{1}{N}\beta_2 cXY_2 = (\mu + \gamma_2 + \tau)Y_2$$

Using the above equation and substituting in (10):

$$0 = \Pi - \mu X - 0 - (\mu + \gamma_2 + \tau)Y_2$$

$$X = \frac{\Pi - (\mu + \gamma_2 + \tau)Y_2}{\mu} \tag{12}$$

Note: In this scenario, N simplifies to $X + Y_2$, as Y_1 is not considered.

Using (11):

$$0 = \frac{\beta_2 cXY_2}{N} - (\mu + \gamma_2 + \tau)Y_2$$

$$\frac{\beta_2 cXY_2}{X + Y_2} = (\mu + \gamma_2 + \tau)Y_2$$

$$\beta_2 cXY_2 = (\mu + \gamma_2 + \tau)Y_2(X + Y_2)$$

$$\beta_2 cXY_2 = (XY_2 + Y_2^2)(\mu + \gamma_2 + \tau)$$

$$\beta_2 cXY_2 = XY_2(\mu + \gamma_2 + \tau) + Y_2^2(\mu + \gamma_2 + \tau)$$

$$\beta_2 cXY_2 - XY_2(\mu + \gamma_2 + \tau) = Y_2^2(\mu + \gamma_2 + \tau)$$

$$X(\beta_2 cY_2 - Y_2(\mu + \gamma_2 + \tau)) = Y_2^2(\mu + \gamma_2 + \tau)$$

$$X = \frac{Y_2^2(\mu + \gamma_2 + \tau)}{(\beta_2 cY_2 - Y_2(\mu + \gamma_2 + \tau))}$$

$$X = \frac{Y_2(\mu + \gamma_2 + \tau)}{(\beta_2 c - (\mu + \gamma_2 + \tau))}$$

Therefore;

$$X = \frac{Y_2(\mu + \gamma_2 + \tau)}{\beta_2 c - \mu - \gamma_2 - \tau} \tag{13}$$

Equating (12) and (13) we obtain;

$$\frac{Y_2(\mu + \gamma_2 + \tau)}{\beta_2 c - \mu - \gamma_2 - \tau} = \frac{\Pi - (\mu + \gamma_2 + \tau)Y_2}{\mu}$$

$$Y_2(\mu + \gamma_2 + \tau)\mu = [\Pi - (\mu + \gamma_2 + \tau)Y_2](\beta_2 c - \mu - \gamma_2 - \tau)$$

$$Y_2(\mu + \gamma_2 + \tau)\mu = \Pi(\beta_2 c - \mu - \gamma_2 - \tau) - (\mu + \gamma_2 + \tau)Y_1(\beta_2 c - \mu - \gamma_2 - \tau)$$

Dividing the equation by $(\mu + \gamma_2 + \tau)$ we obtain

$$Y_2\mu = \frac{\Pi(\beta_2 c - \mu - \gamma_2 - \tau)}{(\mu + \gamma_2 + \tau)} - Y_2(\beta_2 c - \mu - \gamma_2 - \tau)$$

Simplifying the equation we get;

$$Y_1^* = \frac{\Pi(\beta_1 c - \mu - \gamma_2 - \tau)}{(\mu + \gamma_2 + \tau)(\beta_1 c - \gamma_2 - \tau)} \tag{14}$$

Substituting the outcome of Y_1 into (12) we get;

$$X = \frac{\Pi - \cancel{(\mu + \gamma_2 + \tau)}\frac{\Pi(\beta_2 c - \mu - \gamma_2 - \tau)}{\cancel{(\mu + \gamma_2 + \tau)}(\beta_2 c - \gamma_2 - \tau)}}{\mu} \tag{15}$$

$$X = \frac{\Pi - \frac{\Pi(\beta_2 c - \mu - \gamma_2 - \tau)}{(\beta_2 c - \gamma_2 - \tau)}}{\mu}$$

$$X = \frac{\Pi(\beta_2 c - \gamma_2 - \tau) - \Pi(\beta_2 c - \mu - \gamma_2 - \tau)}{(\beta_2 c - \gamma_2 - \tau)\mu}$$

$$X^* = \frac{\Pi}{(\beta_2 c - \gamma_2 - \tau)}$$

Therefore the equilibrium point is ;

$$(X^*, Y_1^*, Y_2^*) = \left(\frac{\Pi}{(\beta_2 c - \gamma_2 - \tau)}, 0, \frac{\Pi(\beta_1 c - \mu - \gamma_2 - \tau)}{(\mu + \gamma_2 + \tau)(\beta_1 c - \gamma_2 - \tau)} \right)$$

To determine if the points mentioned above are stable, we must find the eigenvalues of the Jacobian matrix, which is defined as:

$$J = \begin{bmatrix} \frac{\partial g_1}{\partial X} & \frac{\partial g_1}{\partial Y_1} & \frac{\partial g_1}{\partial Y_2} \\ \frac{\partial g_2}{\partial X} & \frac{\partial g_2}{\partial Y_1} & \frac{\partial g_2}{\partial Y_2} \\ \frac{\partial g_3}{\partial X} & \frac{\partial g_3}{\partial Y_1} & \frac{\partial g_3}{\partial Y_2} \end{bmatrix}$$

$$J = \begin{bmatrix} -\mu - \frac{\beta_1 c Y_1}{N} - \frac{\beta_2 Y_2}{N} & \frac{\beta_{1c} X}{N} & \frac{\beta_2 c X}{N} \\ \frac{\beta_1 c Y_1}{N} & \frac{\beta_1 c X}{N} - (\mu + \gamma_1 + \tau) & 0 \\ \frac{\beta_2 c Y_2}{N} & 0 & \frac{\beta_2 c X}{N} - (\mu + \gamma_2 + \tau) \end{bmatrix}$$

Evaluating the Jacobian Matrix at the equilibrium point obtained above $(X^*, Y_1^*, Y_2^*) = (\frac{\Pi}{\mu}, 0, 0)$ we get that;

$$J^* = \begin{bmatrix} -\mu & \beta_{1c} & \beta_2 c \\ 0 & \beta_1 c - (\mu + \gamma_1 + \tau) & 0 \\ 0 & 0 & \beta_2 c - (\mu + \gamma_2 + \tau) \end{bmatrix}$$

Determining the eigenvalues:

$$J^* - \lambda I = \begin{bmatrix} -\mu - \lambda & \beta_{1c} & \beta_2 c \\ 0 & \beta_1 c - (\mu + \gamma_1 + \tau) - \lambda & 0 \\ 0 & 0 & \beta_2 c - (\mu + \gamma_2 + \tau) - \lambda \end{bmatrix}$$

$$\det(J^* - \lambda I) = \begin{vmatrix} -\mu - \lambda & \beta_{1c} & \beta_2 c \\ 0 & \beta_1 c - (\mu + \gamma_1 + \tau) - \lambda & 0 \\ 0 & 0 & \beta_2 c - (\mu + \gamma_2 + \tau) - \lambda \end{vmatrix}$$

$$\det(J^* - \lambda I) = (-\mu - \lambda)(\beta_1 c - (\mu + \gamma_1 + \tau) - \lambda)(\beta_2 c - (\mu + \gamma_2 + \tau) - \lambda)$$

$$\det(J^* - \lambda I) = 0 = (-\mu - \lambda)(\beta_1 c - (\mu + \gamma_1 + \tau) - \lambda)(\beta_2 c - (\mu + \gamma_2 + \tau) - \lambda)$$

$$\therefore \lambda_1 = -\mu \quad \lambda_2 = \beta_1 c - (\mu + \gamma_1 + \tau) \quad \lambda_3 = \beta_2 c - (\mu + \gamma_2 + \tau)$$

Based on the eigenvalues obtained from the Jacobian matrix, we can assess the stability of the equilibrium point. The eigenvalues $\lambda_1 = -\mu$, $\lambda_2 = \beta_1 c - (\mu + \gamma_1 + \tau)$, and $\lambda_3 = \beta_2 c - (\mu + \gamma_2 + \tau)$ indicate the stability characteristics.

If all eigenvalues have negative real parts, i.e., $\lambda_1 < 0$, $\lambda_2 < 0$, and $\lambda_3 < 0$, the equilibrium point is stable. Conversely, if any eigenvalue has a positive real part, the equilibrium point is unstable.

To ascertain the negativity of the eigenvalues, we proceed by assuming the positivity of all model parameters. Specifically, let us consider that μ, β_1, γ_1, c, τ, γ_2, and β_2 are all positive.

Given the aforementioned assumption regarding the positivity of model parameters, it becomes apparent that $\lambda_1 < 0$. To ensure $\lambda_2 < 0$, it follows that $\beta_1 c - (\mu + \gamma_1 + \tau) < 0$, which simplifies to $\frac{\beta_1 c}{(\mu+\gamma_1+\tau)} < 1$. Therefore, for $\lambda_2 < 0$, the condition $\frac{\beta_1 c}{(\mu+\gamma_1+\tau)} < 1$ must be satisfied. Similarly, for $\lambda_3 < 0$, it is necessary that $\frac{\beta_2 c}{(\mu+\gamma_2+\tau)} < 1$.

Hence, it can be inferred that the disease-free equilibrium point remains stable if the eigenvalues of the Jacobian matrix(J^*) have negative real components, which is only feasible when both λ_2 and λ_3 are negative. Conversely, disease invasion occurs when at least one of these eigenvalues possesses a positive real component. Consequently, instability in the equilibrium point is evident when any of the basic reproduction numbers $R_0^{(i)} > 1$.Where;

$R_0^{(i)} = \frac{\beta_i c}{\mu+\gamma_i+\tau}$

The concept of reproductive numbers plays a crucial role in understanding the spread and establishment of different disease subtypes within a population.

1. *Basic Reproductive Number ($R_0^{(i)}$):* Each subtype of the pathogen has its own basic reproductive number, denoted as $R_0^{(i)}$. This number represents the average number of secondary cases generated by a single infective individual of subtype i when the entire population is susceptible. In simpler terms, it indicates how contagious the subtype is and its potential to spread within a susceptible population.

2. *Invasion Reproductive Number:* The invasion reproductive number of Subtype-2, denoted as $R^{(2)}$, measures the average number of new infections caused by a single individual infected with Subtype-2, considering that Subtype-1 is already established and at equilibrium in the community.

3. *Threshold for Establishment:* If the basic reproductive number $R_0^{(i)}$ for a subtype i is less than 1 ($R_0^{(i)} < 1$), it indicates that the subtype will eventually die out and be eradicated from the population. This is because, on average, each infected individual of subtype i will cause fewer than one new infection, leading to a decline in the prevalence of the subtype over time.

4. *Establishment Condition:* Conversely, if the basic reproductive number $R_0^{(i)}$ for a subtype i exceeds 1 ($R_0^{(i)} > 1$), it suggests that the subtype has the potential to become established within the population. In other words, if an infected individual of subtype i is introduced into the community, the subtype will persist and spread if the average number of secondary cases generated by each infective individual is greater than one.

3 Numerical Solution

4 Finite difference method

We apply the finite difference method and approximate the time derivative by its first order forward-difference approximate given by:

$$\frac{dX(t)}{dt} = \frac{X(t+l) - X(t)}{l} + O(\updownarrow^{\in})$$

as $l \to 0$ any time t after t_0 can be given by $t_n = nl$ for some n= 0,1,2.. Now we make the approximations by calculating some first order change for our 3 equations in the Initial value problem: M_x, M_{Y_1}, M_{Y_2} with l as some discrete time step:

$$M_x : \frac{1}{l}(X^{n+1} - X^n) = \Pi - \mu X^{n+1} - \frac{\beta_1 c X^{n+1} {Y_1}^{n+1}}{X^n + {Y_1}^n + {Y_2}^n} - \frac{\beta_2 c X^{n+1} {Y_2}^{n+1}}{X^n + {Y_1}^n + {Y_2}^n} \tag{16}$$

$$M_{Y_1} : \frac{1}{l}(Y_1^{n+1} - Y_1^n) = \frac{\beta_1 c X^{n+1} Y_1^{n+1}}{X^{n+1} + Y_1^n + Y_2^n} - (\mu + \gamma_1 + \tau) Y_1^{n+1} \tag{17}$$

$$M_{Y_2} : \frac{1}{l}(Y_2^{n+1} - Y_2^n) = \frac{\beta_2 c X^{n+1} Y_2^{n+1}}{X^{n+1} + Y_1^{n+1} + Y_2^n} - (\mu + \gamma_2 + \tau) Y_2^{n+1} \tag{18}$$

we approximate time step, l as

$$l = \Delta t \to \frac{1 - e^{-2l}}{2} \tag{19}$$

This approximation was obtained from Mickens[1] and is important as it is free of chaos and oscilllations.
we now substitute l as described and write the expected form of the difference equation for each discrete change for M_x: first substitute in the the discrete change Δt for l then multiply on both sides

$$\begin{aligned} \frac{1}{l}(X^{n+1} - X^n) &= \Pi - \mu X^{n+1} - \frac{\beta_1 c X^{n+1} {Y_1}^n}{X^n + {Y_1}^n + {Y_2}^n} - \frac{\beta_2 c X^{n+1} {Y_2}^n}{X^n + {Y_1}^n + {Y_2}^n} \\ \Delta t(\frac{1}{\Delta t}(X^{n+1} - X^n)) &= (\Pi - \mu X^{n+1} - \frac{\beta_1 c X^{n+1} {Y_1}^n}{X^n + {Y_1}^n + {Y_2}^n} - \frac{\beta_2 c X^{n+1} {Y_2}^n}{X^n + {Y_1}^n + {Y_2}^n})\Delta t \\ (X^{n+1} - X^n) &= (\Pi - \mu X^{n+1} - \frac{\beta_1 c X^{n+1} {Y_1}^n}{X^n + {Y_1}^n + {Y_2}^n} - \frac{\beta_2 c X^{n+1} {Y_2}^n}{X^n + {Y_1}^n + {Y_2}^n})\Delta t \end{aligned}$$

taking all the X^{n+1} terms to the left and factorising gives:

$$X^{n+1} + \Delta t X^{n+1}(\mu + \frac{\beta_1 c {Y_1}^n}{X^n + {Y_1}^n + {Y_2}^n} + \frac{\beta_2 c {Y_2}^n}{X^n + {Y_1}^n + {Y_2}^n}) = \Pi \Delta t + X^n \tag{20}$$

$$X^{n+1} + \Delta t X^{n+1}(\mu + \frac{\beta_1 c {Y_1}^n + \beta_2 c {Y_2}^n}{X^n + {Y_1}^n + {Y_2}^n}) = \Pi \Delta t + X^n \tag{21}$$

$$X^{n+1} + \Delta t X^{n+1}(\mu + \frac{c}{X^n + {Y_1}^n + {Y_2}^n}(\beta_1 {Y_1}^n + \beta_2 {Y_2}^n)) = \Pi \Delta t + X^n \tag{22}$$

we now simply make X^{n+1} the subject

$$X^{n+1}(1+\Delta t(\mu+\frac{c}{X^n+{Y_1}^n+{Y_2}^n}(\beta_1{Y_1}^n+\beta_2{Y_2}^n)))=\Pi\Delta t+X^n \tag{23}$$

$$X^{n+1}(1+\Delta t(\mu+\frac{c}{X^n+{Y_1}^n+{Y_2}^n}(\beta_1{Y_1}^n+\beta_2{Y_2}^n)))=(X^n+\Pi\Delta t) \tag{24}$$

$$X^{n+1}=(X^n+\Pi\Delta t)/(1+\Delta t(\mu+\frac{c}{X^n+{Y_1}^n+{Y_2}^n}(\beta_1{Y_1}^n+\beta_2{Y_2}^n))) \tag{25}$$

similarly the equations for Y_1^{n+1} and Y_2^{n+1} can be obtained and this gives our 3 equations:

$$X^{n+1}=(X^n+\Pi\Delta t)/(1+\Delta t(\mu+\frac{c}{X^n+{Y_1}^n+{Y_2}^n}(\beta_1{Y_1}^n+\beta_2{Y_2}^n))) \tag{26}$$

$$Y_1^{n+1}=Y_1^n/(1+\Delta t(\mu+\gamma_1+\tau-\frac{\beta_1 cX^{n+1}}{X^{n+1}+Y_1^n+Y_2^n})) \tag{27}$$

$$Y_2^{n+1}=Y_2^n/(1+\Delta t(\mu+\gamma_2+\tau-\frac{\beta_2 cX^{n+1}}{X^{n+1}+Y_1^{n+1}+Y_2^n})) \tag{28}$$

The truncation error associated is of order l^2 confirming the first order accuracy of the solution we hope to obtain.

5 Numerical experiments

We aim to observe the effect of certain variables in the difference equations independently of each other. However we need to first consider the method of solution. That is to compare the implicit method for solving the initial value problem subject to initial values with the $4th$ order Rung katta method (RK4).

Experiment 1: Effect of time-step, l

We apply the inital values: $\Pi = 2000$, $\mu = \frac{1}{32}$, $\tau = 0.4$, $\beta_1 = 0.06$, $\beta_2 = 0.5$, $c = 4$, $\gamma_1 = 0.1$, $\gamma_2 = 0.05$, $X^0 = 8000$, $Y_1^0 = 200$, and $Y_2^0 = 300$.
These parameter estimations were obtained from san francisco in [2] From the application of the implicit method, found here, it is clear the implicit method converges regardless of time step. Rk4 does not offer the same competitive results. While initially accurate, for larger time steps it returned inaccurate results or simply did not converge. It therefore follows that implicit methods are more suited to non-linear initial value problems.

Time-step (Δt)	RK4	Implicit method
0.01	Monotonic convergence	Monotonic convergence
1	Monotonic convergence	Monotonic convergence
3	Wrong solution	Monotonic convergence
3.3	Divergence (method failed)	Monotonic convergence
10	Divergence	Monotonic convergence
1000	Divergence	Monotonic convergence

Table 1: Effect of time-step with RK4 compared to Implicit method

Experiment 2: effect of basic reproductive numbers $R_0^{(i)}$ we consider 5 cases for the reproductive numbers

i)$R_0^{(1)} < 1$ $R_0^{(2)} < 1$
i)i$R_0^{(i)} < 1$ $R_0^{(2)} > 1$
i)ii$R_0^{(i)} > 1$ $R_0^{(2)} < 1$
i)v$R_0^{(i)} > 1$ $R_0^{(2)} > 1$ where they are equal
v)$R_0^{(i)} > 1$ $R_0^{(2)} > 1$ where $R_0^{(2)}$ is significantly larger

As determined before the value of R is the composition of the coefficients in the system $R_0^{(i)} = \frac{\beta_i c}{\mu+\gamma_i+\tau}$ and it suffices to simply consider the resulting reproductive number in each of the above cases. To show this we use different values of beta and gamma then calculate the values of convergence as shown here.
The results are summarised in table 2 below

It can be seen that both sub-types can be eradicated if the reproductive number is below threshhold

R_0^1	R_0^2	X^*	Y_1^*	Y_2^*
0.45176	0.83117	64000	0	0
0.45176	4.15584	1290	0	4072
4.51765	0.83117	1053	3703	0
1.34783	1.34783	3976	553	830
1.82857	4.92301	1042	0	4088

Table 2: Results from varying reproductive numbers

as seen in (i) In the case that one reproductive number is higher that subtype will dominate the other and the only case where they coexist is if the reproductive numbers are equal as seen in case (iv)

Experiment 3: Effect of anti-retroviral therapy, τ:

We apply the following initial values: $\Pi = 2000$, $\mu = \frac{1}{32}$, $\beta_1 = 0.06$, $\beta_2 = 0.1$, $c = 4$, $\gamma_1 = 0.1$, $\gamma_2 = 0.05$, $X^0 = 8000$, $Y_1^0 = 200$, and $Y_2^0 = 300$. Since β_2 is greater than β_1, it should be noted that Subtype-2 will be expected to dominate Subtype-1. From the application of the implicit method, found here and the results in the table below, it can be shown that based on the parameters used in these simulations, a minimum of a 40% efficient anti-retroviral therapy administered across the community is required for the virus to be eliminated.

τ	X^*	Y_1^*	Y_2^*
0	5714	0	22 418
0.1	8000	0	9655
0.2	13 333	0	5630
0.3	40 000	0	1967
0.4	64 000	0	0

Table 3: Results from varying anti-retroviral therapy numbers, τ

6 Conclusion

A model of the subtypes of HIV was developed into a system of equations, given the non-linearity of the system it was found that an implicit, Gauss-Siedel method was best suited to the simulation of the model. This method was found to be more competitive to the explicit schemes like, Euler and RK4 methods, in terms of numerical stability. This model predicts that the basic reproductive numbers of the two subtypes must be below a certain threshold simultaneously (in the case where one reproductive number is higher that subtype will dominate the other, the only case where the subtypes coexist is when the reproductive numbers are equal) with the anti-retroviral therapy coverage level at a minimum of 40% , for the virus to be eliminated.

References

1. R.E. Mickens, Comment on a second-order, choas-free, explicit method for the numerical solution of a cubic reaction problem in neurophysiology, Numer. Methods Partial Differential Equations 10 (1994) 587–590.

2. T.C. Porco, S.M. Blower, Designing HIV vaccination policies: sub-types and cross-immunity, Interfaces 28 (3) (1998) 167–190

Table 1

comparing implicit and RK4 methods

Implicit method:

```
In [ ]:
```

```
import numpy as np
from tabulate import tabulate
```

```
In [ ]:
```

```
# Given constants
P = 2000
mu = 1/32
gamma1 = 0.1
gamma2 = 0.05
tau = 0.4
B1 = 0.06
B2 = 0.5
c = 4
X0 = 8000
Y01 = 200
Y02 = 300
delta_t_values = [0.01, 1, 3, 3.3, 10, 1000]

# Function to perform one iteration of the system of equations using Gauss-Seidel-li
def iterate_system(Xn, Y1n, Y2n, dt):
    Xnp1 = (Xn + P * dt) / (1 + dt * (mu + c / (Xn + Y1n + Y2n) * (B1 * Y1n + B2 * Y
    Y1np1 = Y1n / (1 + dt * (mu + gamma1 + tau - (B1 * c * Xnp1) / (Xnp1 + Y1n + Y2n
    Y2np1 = Y2n / (1 + dt * (mu + gamma2 + tau - (B2 * c * Xnp1) / (Xnp1 + Y1np1 + Y
    return Xnp1, Y1np1, Y2np1

# Function to solve the system for all time steps
def solve_system(delta_t):
    Xn = X0
    Y1n = Y01
    Y2n = Y02
    max_iterations = 200000
    tol = 1e-10

    # Gauss-Seidel-like iteration
    for _ in range(max_iterations):
        Xnp1, Y1np1, Y2np1 = iterate_system(Xn, Y1n, Y2n, delta_t)

        # Check convergence
        if abs(Xnp1 - Xn) < tol and abs(Y1np1 - Y1n) < tol and abs(Y2np1 - Y2n) < to
            return Xnp1, Y1np1, Y2np1

        # Update variables for next iteration
        Xn, Y1n, Y2n = Xnp1, Y1np1, Y2np1

    # If max_iterations reached without convergence
    raise ValueError("Failed to converge within the maximum number of iterations.")

# Solve the system for all time steps
for dt in delta_t_values:
    X, Y1, Y2 = solve_system(dt)
    print(f"For delta_t = {dt}: X = {X}, Y1 = {Y1}, Y2 = {Y2}")
```

```
For delta_t = 0.01: X = 1290.3225806456214, Y1 = 8.873416070380664e-09, Y2 = 4072.056
975302486
For delta_t = 1: X = 1290.3225806451783, Y1 = 1.459474974662082e-15, Y2 = 4072.056975
2826727
```

```
For delta_t = 3: X = 63999.99999999903, Y1 = 5.655120406472925e-95, Y2 = 2.8507236165
568116e-193
For delta_t = 3.3: X = 63999.99999999915, Y1 = 3.3149975955781596e-93, Y2 = 1.0467850
118779407e-193
For delta_t = 10: X = 63999.999999999745, Y1 = 1.0274511853837072e-70, Y2 = 9.1253164
20824659e-139
For delta_t = 1000: X = 64000.0, Y1 = 5.043036460757328e-28, Y2 = 2.0509825879325442e
-36
```

the implicict method is clearly convergent

RK4 method to show convergence:

In [ ]:

```python
def dX_dt(P, mu, B1, B2, c, X, Y1, Y2):
    N = X + Y1 + Y2
    if N == 0:
        return 0
    else:
        return P - mu * X - (1/N) * B1 * c * X * Y1 - (1/N) * B2 * c * X * Y2


def dY1_dt(B1, c, mu, gamma1, tau, X, Y1):
    N = X + Y1 + Y2
    if N == 0:
        return 0
    else:
        return (1/N) * B1 * c * X * Y1 - (mu + gamma1 + tau) * Y1



def dY2_dt(B2, c, mu, gamma2, tau, X, Y2):
    N = X + Y1 + Y2
    if N == 0:
        return 0
    else:
        return (1/N) * B2 * c * X * Y2 - (mu + gamma2 + tau) * Y2

def RK4_step(P, mu, B1, B2, c, gamma1, gamma2, tau, X, Y1, Y2, dt):
    k1_X = dX_dt(P, mu, B1, B2, c, X, Y1, Y2)
    k1_Y1 = dY1_dt(B1, c, mu, gamma1, tau, X, Y1)
    k1_Y2 = dY2_dt(B2, c, mu, gamma2, tau, X, Y2)

    X_half = X + 0.5 * dt * k1_X
    Y1_half = Y1 + 0.5 * dt * k1_Y1
    Y2_half = Y2 + 0.5 * dt * k1_Y2

    k2_X = dX_dt(P, mu, B1, B2, c, X_half, Y1_half, Y2_half)
    k2_Y1 = dY1_dt(B1, c, mu, gamma1, tau, X_half, Y1_half)
    k2_Y2 = dY2_dt(B2, c, mu, gamma2, tau, X_half, Y2_half)

    X_half2 = X + 0.5 * dt * k2_X
    Y1_half2 = Y1 + 0.5 * dt * k2_Y1
    Y2_half2 = Y2 + 0.5 * dt * k2_Y2

    k3_X = dX_dt(P, mu, B1, B2, c, X_half2, Y1_half2, Y2_half2)
    k3_Y1 = dY1_dt(B1, c, mu, gamma1, tau, X_half2, Y1_half2)
    k3_Y2 = dY2_dt(B2, c, mu, gamma2, tau, X_half2, Y2_half2)

    X_end = X + dt * k3_X
    Y1_end = Y1 + dt * k3_Y1
    Y2_end = Y2 + dt * k3_Y2

    return X_end, Y1_end, Y2_end
```

```
# Parameters and initial conditions
P = 2000
mu = 1/32
gamma1 = 0.1
gamma2 = 0.05
tau = 0.4
B1 = 0.06
B2 = 0.5
c = 4
X0 = 8000
Y01 = 200
Y02 = 300
dt_values = [0.01, 1, 3, 3.3, 10, 1000]

# Simulation
for dt in dt_values:
    X = X0
    Y1 = Y01
    Y2 = Y02
    print(f"Time step: {dt}")
    for i in range(130):  # Adjust number of steps as needed
        X_prev, Y1_prev, Y2_prev = X, Y1, Y2
        X, Y1, Y2 = RK4_step(P, mu, B1, B2, c, gamma1, gamma2, tau, X, Y1, Y2, 1/dt)
    # Check convergence using epsilon
    epsilon = 1e-6  # Define a small tolerance
    if abs(X - X_prev) < epsilon and abs(Y1 - Y1_prev) < epsilon and abs(Y2 - Y2_pre
        print("Converged")
    else:
        print("Diverged")
    print()
```

```
Time step: 0.01
Converged

Time step: 1
Converged

Time step: 3
Converged

Time step: 3.3
Diverged

Time step: 10
Diverged

Time step: 1000
Diverged
```

Inconsistency in convergence for the time steps in RK4

Table 2 Effect of basic reproductive numbers

In [ ]:

```
# Given constants
P = 2000
mu = 1/32
gamma1 = 0.1
gamma2 = 0.05
tau = 0.4
B1 = 0.06
B2 = 0.5
c = 4
X0 = 8000
```

```
Y01 = 200
Y02 = 300
delta_t_values = [0.01]

# Function to perform one iteration of the system of equations using Gauss-Seidel-li
def iterate_system(Xn, Y1n, Y2n, dt):
    Xnp1 = (Xn + P * dt) / (1 + dt * (mu + c / (Xn + Y1n + Y2n) * (B1 * Y1n + B2 * Y
    Y1np1 = Y1n / (1 + dt * (mu + gamma1 + tau - (B1 * c * Xnp1) / (Xnp1 + Y1n + Y2n
    Y2np1 = Y2n / (1 + dt * (mu + gamma2 + tau - (B2 * c * Xnp1) / (Xnp1 + Y1np1 + Y
    return Xnp1, Y1np1, Y2np1

# Function to solve the system for all time steps
def solve_system(delta_t):

    max_iterations = 2000000
    tol = 1e-6

    # Solve for each delta_t
    for dt in delta_t:
        Xn = X0
        Y1n = Y01
        Y2n = Y02

        # Gauss-Seidel-like iteration
        for _ in range(max_iterations):
            Xnp1, Y1np1, Y2np1 = iterate_system(Xn, Y1n, Y2n, dt)

            # Check convergence
            if abs(Xnp1 - Xn) < tol and abs(Y1np1 - Y1n) < tol and abs(Y2np1 - Y2n)
                R1 =  (B1 * c)/(mu + gamma1 + tau)
                R2 =  (B2 * c)/(mu + gamma2 + tau)
                print(f"R1 = {round(R1, 5)}, R2 = {round(R2, 5)}, X = {round(Xnp1)},
                break  # Break out of the loop if converged

            # Update variables for next iteration
            Xn, Y1n, Y2n = Xnp1, Y1np1, Y2np1

beta1_array = [0.06, 0.06, 0.6, 0.457, 0.3]
gamma1_array = [0.1, 0.1, 0.1, 0.925, 0.225]
beta2_array = [0.1, 0.5, 0.1, 0.457, 0.5923]
gamma2_array = [0.05, 0.05, 0.05, 0.925, 0.05]
for i in range(5):

    B1 = beta1_array[i]
    gamma1 = gamma1_array[i]
    B2 = beta2_array[i]
    gamma2 = gamma2_array[i]
    print()
    solve_system(delta_t_values)
```

```
R1 = 0.45176, R2 = 0.83117, X = 64000, Y1 = 0, Y2 = 0

R1 = 0.45176, R2 = 4.15584, X = 1290, Y1 = 0, Y2 = 4072

R1 = 4.51765, R2 = 0.83117, X = 1053, Y1 = 3703, Y2 = 0

R1 = 1.34783, R2 = 1.34783, X = 3976, Y1 = 553, Y2 = 830

R1 = 1.82857, R2 = 4.92301, X = 1042, Y1 = 0, Y2 = 4088
```

Table 3 Effect of anti-retroviral therapy

```
In [ ]:

def table_3(P, mu, b1, b2, c, gamma1, gamma2, X0, Y10, Y20, tau, dt, num_steps):
    X = np.zeros(num_steps)
    Y1 = np.zeros(num_steps)
    Y2 = np.zeros(num_steps)

    X[0] = X0
    Y1[0] = Y10
    Y2[0] = Y20

    for n in range(num_steps - 1):
        X[n+1] = (X[n] + P * dt) / (1 + dt * (mu + c / (X[n] + Y1[n] + Y2[n]) * (b1
        Y1[n+1] = Y1[n] / (1 + dt * (mu + gamma1 + tau - (b1 * c * X[n+1]) / (X[n+1]
        Y2[n+1] = Y2[n] / (1 + dt * (mu + gamma2 + tau - (b2 * c * X[n+1]) / (X[n+1]

    return X, Y1, Y2

#Parameters and initial values
P = 2000
mu = 1/32
b1 = 0.06
b2 = 0.1
c = 4
gamma1 = 0.1
gamma2 = 0.05
X0 = 8000
Y10 = 200
Y20 = 300
tau_values = [0, 0.1, 0.2, 0.3, 0.4]
dt = 0.01
num_steps = 200000

for tau in tau_values:
    X, Y1, Y2 = table_3(P, mu, b1, b2, c, gamma1, gamma2, X0, Y10, Y20, tau, dt, num
    print(f"For tau = {tau}: X = {round(X[-1])}, Y1 = {round(Y1[-1])}, Y2 = {round(Y
```

```
For tau = 0: X = 5714, Y1 = 0, Y2 = 22418
For tau = 0.1: X = 8000, Y1 = 0, Y2 = 9655
For tau = 0.2: X = 13333, Y1 = 0, Y2 = 5630
For tau = 0.3: X = 40000, Y1 = 0, Y2 = 1967
For tau = 0.4: X = 64000, Y1 = 0, Y2 = 0
```