Homework 3

Thomas Zhang

October 2023

1

a.

$$\frac{dS}{dt} = -\lambda_1 \frac{SI}{N} + \lambda_4 R$$
$$\frac{dE}{dt} = \lambda_1 \frac{SI}{N} - \lambda_2 E$$
$$\frac{dI}{dt} = \lambda_2 E - \lambda_3 I$$
$$\frac{dR}{dt} = \lambda_3 I - \lambda_4 R$$

$$S_{i+1} = S_i + (-\lambda_1 \frac{S_i I_i}{N_i} + \lambda_4 R_i)\Delta t$$
$$E_{i+1} = E_i + (\lambda_1 \frac{S_i I_i}{N_i} - \lambda_2 E_i)\Delta t$$
$$I_{i+1} = I_i + (\lambda_2 E_i - \lambda_3 I_i)\Delta t$$
$$R_{i+1} = R_i + (\lambda_3 I_i - \lambda_4 R_i)\Delta t$$

b. Let: k_s, k_e, k_i, k_r be proportionality constants. $\mathcal{N}(0,1)$ is a Gaussian with variance = 1.

$$\frac{dS}{dt} = -\lambda_1 \frac{SI}{N} + \lambda_4 R - k_1 \sqrt{\lambda_1 \frac{SI}{N}} \mathcal{N}_1(0,1) + k_4 \sqrt{\lambda_4 R} \mathcal{N}_4(0,1)$$
$$\frac{dE}{dt} = \lambda_1 \frac{SI}{N} - \lambda_2 E + k_1 \sqrt{\lambda_1 \frac{SI}{N}} \mathcal{N}_1(0,1) - k_2 \sqrt{\lambda_2 E} \mathcal{N}_2(0,1)$$
$$\frac{dI}{dt} = \lambda_2 E - \lambda_3 I + k_2 \sqrt{\lambda_2 E} \mathcal{N}_2(0,1) - k_3 \sqrt{\lambda_3 I} \mathcal{N}_3(0,1)$$
$$\frac{dR}{dt} = \lambda_3 I - \lambda_4 R + k_3 \sqrt{\lambda_3 I} \mathcal{N}_3(0,1) - k_4 \sqrt{\lambda_4 R} \mathcal{N}_4(0,1)$$

$$S_{i+1} = S_i + (-\lambda_1 \frac{S_i I_i}{N_i} + \lambda_4 R_i)\Delta t + \sqrt{\Delta t}(-k_1 \sqrt{\lambda_1 \frac{S_i I_i}{N_i}} \mathcal{N}_1(0,1) + k_4 \sqrt{\lambda_4 R_i} \mathcal{N}_4(0,1))$$
$$E_{i+1} = E_i + (\lambda_1 \frac{S_i I_i}{N_i} - \lambda_2 E_i)\Delta t + \sqrt{\Delta t}(k_1 \sqrt{\lambda_1 \frac{S_i I_i}{N_i}} \mathcal{N}_1(0,1) - k_2 \sqrt{\lambda_2 E_i} \mathcal{N}_2(0,1))$$
$$I_{i+1} = I_i + (\lambda_2 E_i - \lambda_3 I_i)\Delta t + \sqrt{\Delta t}(k_2 \sqrt{\lambda_2 E_i} \mathcal{N}_2(0,1) - k_3 \sqrt{\lambda_3 I_i} \mathcal{N}_3(0,1))$$
$$R_{i+1} = R_i + (\lambda_3 I_i - \lambda_4 R_i)\Delta t + \sqrt{\Delta t}(k_3 \sqrt{\lambda_3 I_i} \mathcal{N}_3(0,1) - k_4 \sqrt{\lambda_4 R_i} \mathcal{N}_4(0,1))$$

c.

$$
\begin{aligned}
\frac{dS}{dt} &= -\lambda_1 \frac{SI}{N} + \lambda_4 R + D_S \nabla^2 S \\
\frac{dE}{dt} &= \lambda_1 \frac{SI}{N} - \lambda_2 E + D_E \nabla^2 E \\
\frac{dI}{dt} &= \lambda_2 E - \lambda_3 I \\
\frac{dR}{dt} &= \lambda_3 I - \lambda_4 R + D_R \nabla^2 R
\end{aligned}
$$

$$
\begin{aligned}
S_{i+1} &= S_i + (-\lambda_1 \frac{S_i I_i}{N_i} + \lambda_4 R_i + D_S \nabla^2 S_i)\Delta t \\
E_{i+1} &= E_i + (\lambda_1 \frac{S_i I_i}{N_i} - \lambda_2 E_i + D_E \nabla^2 E_i)\Delta t \\
I_{i+1} &= I_i + (\lambda_2 E_i - \lambda_3 I_i)\Delta t \\
R_{i+1} &= R_i + (\lambda_3 I_i - \lambda_4 R_i + D_R \nabla^2 R_i)\Delta t
\end{aligned}
$$

$$
\begin{aligned}
\nabla^2 S_i &= D_S \begin{pmatrix} & \frac{S_i(x_{n+1},y_n,z_n)+S_i(x_{n-1},y_n,z_n)-2S_i(x_n,y_n,z_n)}{\Delta x^2} \\ + & \frac{S_i(x_n,y_{n+1},z_n)+S_i(x_n,y_{n-1},z_n)-2S_i(x_n,y_n,z_n)}{\Delta y^2} \\ + & \frac{S_i(x_n,y_n,z_{n+1})+S_i(x_n,y_n,z_{n-1})-2S_i(x_n,y_n,z_n)}{\Delta z^2} \end{pmatrix} \\
\nabla^2 E_i &= D_E \begin{pmatrix} & \frac{E_i(x_{n+1},y_n,z_n)+E_i(x_{n-1},y_n,z_n)-2E_i(x_n,y_n,z_n)}{\Delta x^2} \\ + & \frac{E_i(x_n,y_{n+1},z_n)+E_i(x_n,y_{n-1},z_n)-2E_i(x_n,y_n,z_n)}{\Delta y^2} \\ + & \frac{E_i(x_n,y_n,z_{n+1})+E_i(x_n,y_n,z_{n-1})-2E_i(x_n,y_n,z_n)}{\Delta z^2} \end{pmatrix} \\
\nabla^2 R_i &= D_R \begin{pmatrix} & \frac{R_i(x_{n+1},y_n,z_n)+R_i(x_{n-1},y_n,z_n)-2R_i(x_n,y_n,z_n)}{\Delta x^2} \\ + & \frac{R_i(x_n,y_{n+1},z_n)+R_i(x_n,y_{n-1},z_n)-2R_i(x_n,y_n,z_n)}{\Delta y^2} \\ + & \frac{R_i(x_n,y_n,z_{n+1})+R_i(x_n,y_n,z_{n-1})-2R_i(x_n,y_n,z_n)}{\Delta z^2} \end{pmatrix}
\end{aligned}
$$

2

a.

$$
\begin{aligned}
A_{i+1} &= A_i + \Delta t(k_2 C_i^2 - k_1 A_i B_i) \\
B_{i+1} &= B_i + \Delta t(k_2 C_i^2 - k_1 A_i B_i) \\
C_{i+1} &= C_i + \Delta t(2k_1 A_i B_i - 2k_2 C_i^2)
\end{aligned}
$$

b.

	t = 0	t = 0.1
A	2	$2 + 0.1(2 \times 0^2 - 1 \times 2 \times 1) = 1.8$
B	1	$1 + 0.1(2 \times 0^2 - 1 \times 2 \times 1) = 0.8$
C	0	$0 + 0.1(2 \times 1 \times 2 \times 1 - 2 \times 2 \times 0^2) = 0.4$

Table 1: Forward Euler. Single Step

c.

$$
\begin{aligned}
A &= A(0) - x \\
B &= B(0) - x \\
C &= C(0) + 2x
\end{aligned}
$$

$$
\begin{aligned}
\frac{dA_{i+1}}{dt} = 0 - \frac{dx}{dt} &= k_2 C^2 - k_1 AB \\
\frac{dx}{dt} &= k_1 AB - k_2 C^2
\end{aligned}
$$

$$
\begin{aligned}
\frac{dA_{i+1}}{dt} = 0 - \frac{dx}{dt} &= k_2 C^2 - k_1 AB \\
\frac{dx}{dt} &= k_1 AB - k_2 C^2 \\
\frac{dx}{dt} &= k_1((A(0) - x)(B(0) - x) - k_2(C(0) + 2x)^2
\end{aligned}
$$

$$
\begin{aligned}
x_{i+1} =& x_i + \Delta t \frac{dx_{i+1}}{dt} \\
x_{i+1} =& x_i + \Delta t(k_1((A(0) - x_{i+1})(B(0) - x_{i+1}) - k_2(C(0) + 2x_{i+1})^2) \\
x_{i+1} =& x_i + \Delta t(k_1 A(0)B(0) - k_1(A(0) + B(0))x_{i+1} + k_1 x_{i+1}^2 - k_2 C(0)^2 - k_2 4C(0)x_{i+1} - k_2 4x_{i+1}^2) \\
x_{i+1} =& x_i + \Delta t(k_1 A(0)B(0) - k_2 C(0)^2 - k_1(A(0) + B(0))x_{i+1} - k_2 4C(0)x_{i+1} + k_1 x_{i+1}^2 - k_2 4x_{i+1}^2) \\
x_{i+1} =& x_i + \Delta t(k_1 A(0)B(0) - k_2 C(0)^2 - (k_1(A(0) + B(0)) + k_2 4C(0))x_{i+1} + (k_1 - k_2 4)x_{i+1}^2) \\
0 =& x_i + k_1 A(0)B(0) - k_2 C(0)^2)\Delta t - (\frac{1}{\Delta t} + k_1(A(0) + B(0)) + k_2 4C(0))\Delta t x_{i+1} + (k_1 - k_2 4)\Delta t x_{i+1}^2
\end{aligned}
$$

The above is a quadratic equation of the form $0 = ax_{i+1}^2 + bx_{i+1} + c$ where

$$
\begin{aligned}
a &= (k_1 - 4k_2)\Delta t \\
b &= -(\frac{1}{\Delta t} + k_1(A(0) + B(0)) + 4k_2 C(0))\Delta t \\
c &= (x_i + k_1 A(0)B(0) - k_2 C(0)^2)\Delta t
\end{aligned}
$$

The solution to a quadratic is $x_{i+1} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

d.

	t= 0	t = 0.1
a	-0.7	-0.7
b	-1.3	-1.3
c		0.2
X	0	0.14286
A	2	1.85714
B	1	0.85714
C	0	0.28572

Table 2: Backward Euler

e.

$$A_{i+\frac{1}{2}} = A_i + \frac{\Delta t}{2}(k_2 C_i^2 - k_1 A_i B_i)$$
$$B_{i+\frac{1}{2}} = B_i + \frac{\Delta t}{2}(k_2 C_i^2 - k_1 A_i B_i)$$
$$C_{i+\frac{1}{2}} = C_i + \frac{\Delta t}{2}(2k_1 A_i B_i - 2k_2 C_i^2)$$
$$A_{i+1} = Ai + \Delta t(k_2 C_{i+\frac{1}{2}}^2 - k_1 A_{i+\frac{1}{2}} B_{i+\frac{1}{2}})$$
$$B_{i+1} = Bi + \Delta t(k_2 C_{i+\frac{1}{2}}^2 - k_1 A_{i+\frac{1}{2}} B_{i+\frac{1}{2}})$$
$$C_{i+1} = C_i + \Delta t(2k_1 A_{i+\frac{1}{2}} B_{i+\frac{1}{2}} - 2k_2 C_{i+\frac{1}{2}}^2)$$

f.

	t = 0	t = 0.1
A	2	1.837
B	1	0.837
C	0	0.326
$A_{i+\frac{1}{2}}$	1.9	
$B_{i+\frac{1}{2}}$	0.9	
$C_{i+\frac{1}{2}}$	0.2	

Table 3: Midpoint Euler

3

a.

$$
\begin{aligned}
C(x_0 + \Delta x) &= C(x_0) + \frac{\Delta x}{1!}\frac{dC(X_0)}{dx} + \frac{\Delta x^2}{2!}\frac{d^2C(X_0)}{dx^2} + \frac{\Delta x^3}{3!}\frac{d^3C(X_0)}{dx^3} \\
C(x_0 + \Delta x) - C(x_0) - \frac{\Delta x^2}{2!}\frac{d^2C(X_0)}{dx^2} &= \frac{\Delta x}{1!}\frac{dC(X_0)}{dx} + \frac{\Delta x^3}{3!}\frac{d^3C(X_0)}{dx^3} \\
\frac{C(x_0 + \Delta x) - C(x_0)}{\Delta x} - \frac{\Delta x}{2!}\frac{d^2C(X_0)}{dx^2} &= \frac{dC(X_0)}{dx} + \frac{\Delta x^2}{3!}\frac{d^3C(X_0)}{dx^3}
\end{aligned}
$$

Second order error term $\frac{\Delta x^2}{3!}\frac{d^3C(X_0)}{dx^3}$.

Second order approximation: $\frac{dC(X_0)}{dx} = \frac{C(x_0+\Delta x)-C(x_0)}{\Delta x} - \frac{\Delta x}{2!}\frac{d^2C(X_0)}{dx^2}$

b.

$$
\begin{aligned}
\frac{\partial C}{\partial x} &= aC(x_0) + bC(x_0 + \Delta x) + c\frac{\partial^2 C}{\partial x^2}(X_0) \\
\frac{\partial C}{\partial x} &= \frac{C(x_0 + \Delta x) - C(x_0)}{\Delta x} - \frac{\Delta x}{2!}\frac{\partial^2 C}{\partial x^2}(X_0)
\end{aligned}
$$

$$
\begin{aligned}
aC(x_0) &= \frac{-C(x_0)}{\Delta x} \\
bC(x_0 + \Delta x) &= \frac{C(x_0 + \Delta x)}{\Delta x} \\
c\frac{\partial^2 C}{\partial x^2}(X_0) &= -\frac{\Delta x}{2!}\frac{\partial^2 C}{\partial x^2}(X_0)
\end{aligned}
$$

c.

$$
\begin{aligned}
a &= -\frac{1}{\Delta x} \\
b &= \frac{1}{\Delta x} \\
c &= -\frac{\Delta x}{2!}
\end{aligned}
$$

$$
\frac{\partial C}{\partial x} = -\frac{C(x_0)}{\Delta x} + \frac{C(x_0 + \Delta x)}{\Delta x} - \frac{\Delta x}{2!}\frac{\partial^2 C}{\partial x^2}(X_0)
$$

d. The formula is the forward Euler formula when $\frac{\partial^2 C}{\partial x^2}(X_0) = 0$. This is because the expression $\frac{\Delta x}{2!}\frac{\partial^2 C}{\partial x^2}(X_0)$ is actually the smallest order error term of the forward Euler. So if we set this value to 0, then the error of the approximation derived in this question, and the error of forward Euler are equal. In other words, the formula is second order accurate, but the accuracy of the forward Euler is second order when $\frac{\partial^2 C}{\partial x^2}(X_0) = 0$.

4

a.

$$x_{i+1} = x_i + \Delta t(ax_i - bx_iy_i)$$
$$y_{i+1} = y_i + \Delta t(cx_iy_i - dy_i)$$

b.

$$x_{i+1} = x_i + \Delta t(ax_{i+1} - bx_{i+1}y_{i+1})$$
$$y_{i+1} = y_i + \Delta t(cx_{i+1}y_{i+1} - dy_{i+1})$$

$$\begin{aligned}
x_{i+1} &= x_i + \Delta t(ax_{i+1} - bx_{i+1}y_{i+1}) \\
x_{i+1} - \Delta t(ax_{i+1} - bx_{i+1}y_{i+1}) &= x_i \\
(1 - \Delta t(a - by_{i+1}))x_{i+1} &= x_i \\
x_{i+1} &= \frac{x_i}{1 - \Delta t(a - by_{i+1})}
\end{aligned}$$

$$\begin{aligned}
y_{i+1} &= y_i + \Delta t(cx_{i+1}y_{i+1} - dy_{i+1}) \\
y_{i+1} - \Delta t(cx_{i+1}y_{i+1} - dy_{i+1}) &= y_i \\
(1 - \Delta t(cx_{i+1} - d))y_{i+1} &= y_i \\
y_{i+1} &= \frac{y_i}{(1 - \Delta t(cx_{i+1} - d))}
\end{aligned}$$

Plug y_{i+1} into the expression for x_{i+1}.

$$\begin{aligned}
x_{i+1} &= \frac{x_i}{1 - \Delta t(a - by_{i+1})} \\
x_{i+1} &= \frac{x_i}{1 - \Delta t(a - \frac{by_i}{1-\Delta t(cx_{i+1}-d)})} \\
x_{i+1}(1 - \Delta t(a - \frac{by_i}{1 - \Delta t(cx_{i+1} - d)})) &= x_i \\
x_{i+1}((1 - \Delta ta)(1 - \Delta t(cx_{i+1} - d)) + \Delta tby_i) &= x_i(1 - \Delta t(cx_{i+1} - d)) \\
x_{i+1}((1 - \Delta ta)(1 - \Delta t(cx_{i+1} - d)) + \Delta tby_i) &= x_i(1 + \Delta td) - \Delta tcx_ix_{i+1} \\
x_{i+1}((1 - \Delta ta)(1 - \Delta t(cx_{i+1} - d)) + \Delta tby_i) + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
x_{i+1}(1 - \Delta t(cx_{i+1} - d) - \Delta ta + a\Delta t^2(cx_{i+1} - d) + \Delta tby_i) + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
x_{i+1}(1 - \Delta tcx_{i+1} + \Delta td - \Delta ta + a\Delta t^2cx_{i+1} - a\Delta t^2d + \Delta tby_i) + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
x_{i+1}(1 + \Delta td - \Delta ta - a\Delta t^2d + \Delta tby_i - \Delta tcx_{i+1} + a\Delta t^2cx_{i+1}) + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
x_{i+1}(1 + (d - a + by_i)\Delta t - ad\Delta t^2 - c\Delta tx_{i+1} + ac\Delta t^2x_{i+1}) + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
(1 + (d - a + by_i)\Delta t - ad\Delta t^2)x_{i+1} + (ac\Delta t^2 - c\Delta t)x_{i+1}^2 + \Delta tcx_ix_{i+1} &= x_i(1 + \Delta td) \\
(1 + (d - a + by_i + cx_i)\Delta t - ad\Delta t^2)x_{i+1} + (ac\Delta t^2 - c\Delta t)x_{i+1}^2 &= x_i(1 + \Delta td) \\
(ac\Delta t^2 - c\Delta t)x_{i+1}^2 + (1 + (d - a + by_i + cx_i)\Delta t - ad\Delta t^2)x_{i+1} - (1 + \Delta td)x_i &= 0
\end{aligned}$$

This is a quadratic of the form $Ax^2 + Bx + C = 0$ where:

$$\begin{aligned}
A &= ac\Delta t^2 - c\Delta t \\
B &= 1 + (d - a + by_i + cx_i)\Delta t - ad\Delta t^2 \\
C &= -(1 + \Delta td)x_i
\end{aligned}$$

so $x_{i+1} = \frac{-B\pm\sqrt{B^2-4AC}}{4AC}$ and then y_{i+1} can be solved by plugging in x_{i+1} into the backward Euler equation derived above.

c.

```
/* Set initial conditions */
```
Initialize $a, b, c, d, \Delta t$;

$p \leftarrow \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$;

for $i \leftarrow 0$ **to** 10 **do**

  $p \leftarrow p_0 - \text{JACOBIAN}(g(p))^{-1} g(p)$

end

Return p;

Algorithm 1: Newton-Raphson

where

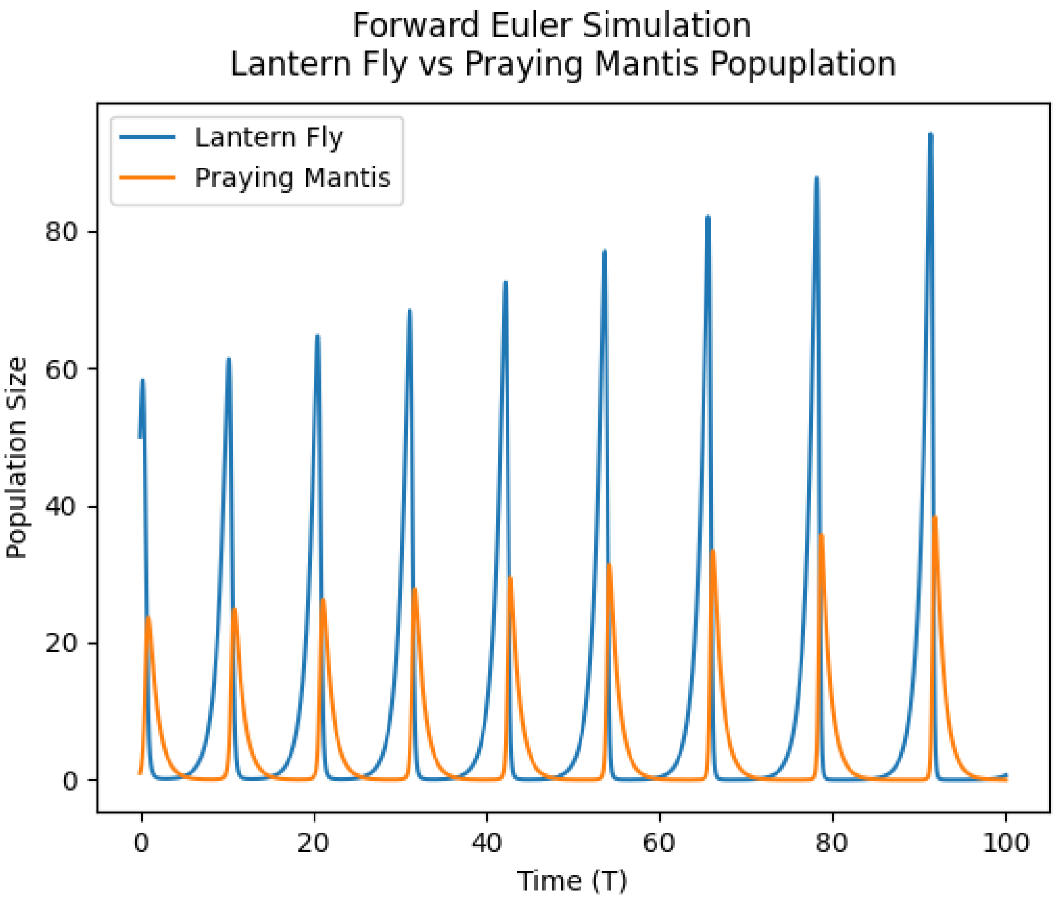
$$g(p) = g(x, y) = \begin{bmatrix} x - x_0 - \Delta t(ax - bxy) \\ y - y_0 - \Delta t(cxy - dy) \end{bmatrix}$$

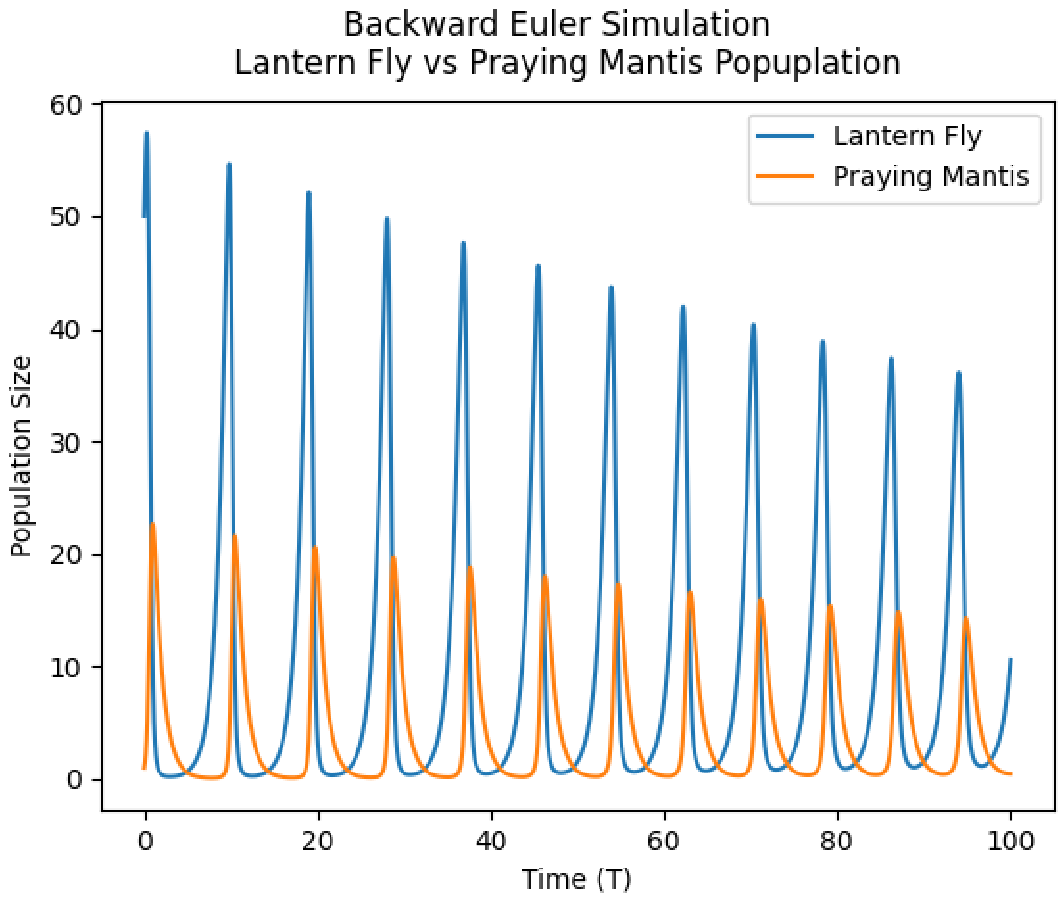
$$\text{JACOBIAN}(g(p)) = \text{JACOBIAN}(g(x, y)) = \begin{bmatrix} 1 - \Delta t(a - by) & \Delta t(bx) \\ -\Delta t(cy) & 1 - \Delta t(cx - d) \end{bmatrix}$$

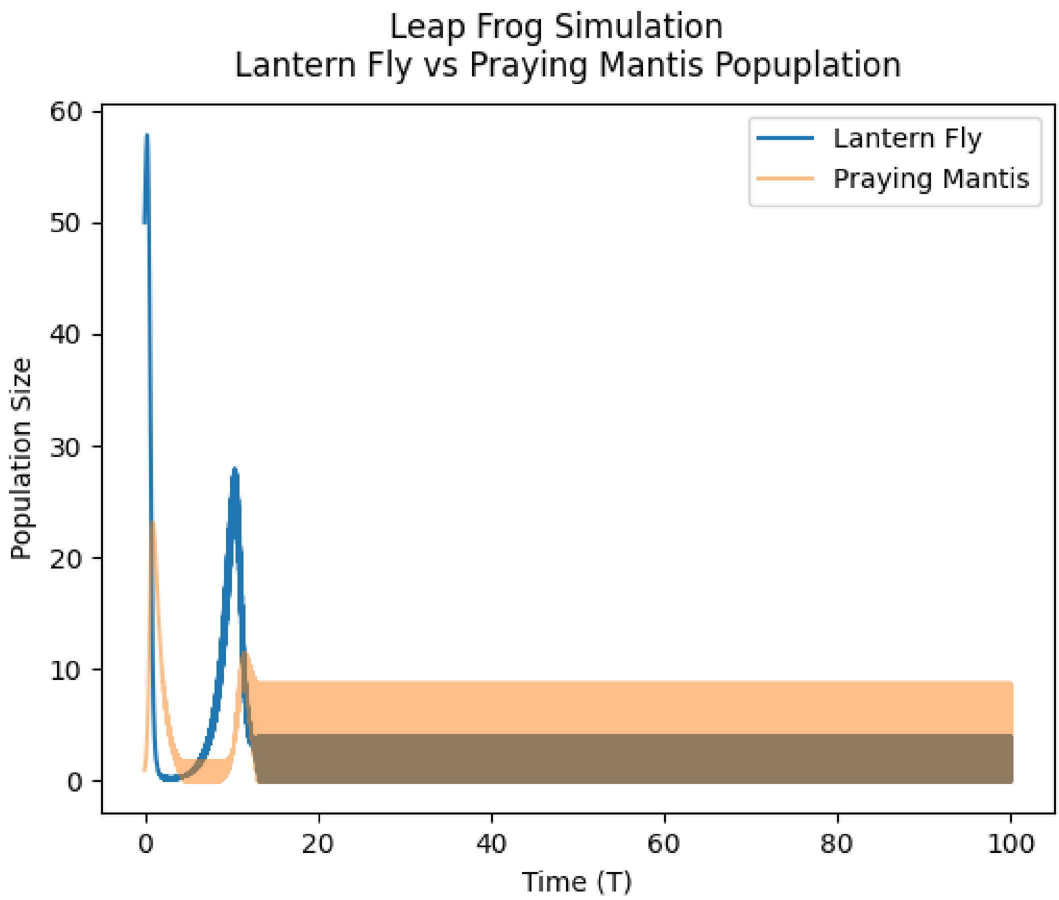
d.

$$x_{i+1} = x_{i-1} + 2\Delta t(ax_i - bx_i y_i)$$
$$y_{i+1} = y_{i-1} + 2\Delta t(cx_i y_i - dy_i)$$

f.




Backward Euler Simulation
Lantern Fly vs Praying Mantis Popuplation
Lantern Fly
Praying Mantis
Population Size
Time (T)
0
10
20
30
40
50
60
0
20
40
60
80
100


Leap Frog Simulation
Lantern Fly vs Praying Mantis Popuplation
Lantern Fly
Praying Mantis
Population Size
Time (T)
0
10
20
30
40
50
60
0
20
40
60
80
100

g. The forward Euler simulation implies that the lantern fly population will oscillate in an increasing trend. The backward Euler simulation implies that the lantern fly population will oscillate in an decreasing trend. The leap frog simulation implies that the lantern fly population will oscillate in neither a decreasing or increasing fashion.

The simulations do not lead to the same answer. The best model is most likely the backward Euler simulation because a perfect model would stabilize at either 0 or some constant. This is because in reality, populations can never grow indefinitely, they can only oscillate around a stable point or the population can crash to 0 if the species go extinct. This reflects the backward model which has a decreasing , oscillating trend, which can either stabilize at some constant or drop to 0.

The other two simulations are worse because the forward Euler simulation trends upwards which would imply an infinitely growing population, and the leap-frog simulation must be bounded by 0, otherwise there are negative populations.

5

a.

$$x(t-\Delta t_1) = x(t) - \frac{\Delta t_1}{1!}x'(t) + \frac{\Delta t_1^2}{2!}x''(t) - \frac{\Delta t_1^3}{3!}x'''(t) + \frac{\Delta t_1^4}{4!}x''''(t)$$
$$x(t+\Delta t_2) = x(t) + \frac{\Delta t_2}{1!}x'(t) + \frac{\Delta t_2^2}{2!}x''(t) + \frac{\Delta t_2^3}{3!}x'''(t) + \frac{\Delta t_2^4}{4!}x''''(t)$$

b.

$$x(t+\Delta t_2) = x(t) + \frac{\Delta t_2}{1!}x'(t) + \frac{\Delta t_2^2}{2!}x''(t) + \frac{\Delta t_2^3}{3!}x'''(t) + \frac{\Delta t_2^4}{4!}x''''(t)$$
$$x(t-\Delta t_1) = x(t) - \frac{\Delta t_1}{1!}x'(t) + \frac{\Delta t_1^2}{2!}x''(t) - \frac{\Delta t_1^3}{3!}x'''(t) + \frac{\Delta t_1^4}{4!}x''''(t)$$

$$ax(t-\Delta t_1) + bx(t) + cx'(t) \approx x(t+\Delta t_2)$$

$$a(x(t) - \frac{\Delta t_1}{1!}x'(t) + \frac{\Delta t_1^2}{2!}x''(t) - \frac{\Delta t_1^3}{3!}x'''(t)) + bx(t) + cx'(t) = x(t) + \frac{\Delta t_2}{1!}x'(t) + \frac{\Delta t_2^2}{2!}x''(t) + \frac{\Delta t_2^3}{3!}x'''(t)$$
$$(a+b)x(t) + (c - a\Delta t_1)x'(t) + a\frac{\Delta t_1^2}{2!}x''(t) - a\frac{\Delta t_1^3}{3!}x'''(t) = x(t) + \frac{\Delta t_2}{1!}x'(t) + \frac{\Delta t_2^2}{2!}x''(t) + \frac{\Delta t_2^3}{3!}x'''(t)$$

$$a + b = 1$$
$$c - a\Delta t_1 = \Delta t_2$$
$$a\frac{\Delta t_1^2}{2!} = \frac{\Delta t_2^2}{2!}$$

c.

$$a + b = 1$$
$$c - a\Delta t_1 = \Delta t_2$$
$$a\frac{\Delta t_1^2}{2!} = \frac{\Delta t_2^2}{2!}$$

$$a = \frac{\Delta t_2^2}{\Delta t_1^2}$$
$$b = 1 - \frac{\Delta t_2^2}{\Delta t_1^2}$$
$$c = \Delta t_2 + \frac{\Delta t_2^2}{\Delta t_1}$$

d. Dropped error term:

$$(a \times \Delta t_1^3)O(\Delta t^3) + \Delta t_2^3 O(\Delta t^3)$$
$$(\frac{\Delta t_2^2}{\Delta t_1^2} \times \Delta t_1^3)O(\Delta t^3) + \Delta t_2^3 O(\Delta t^3)$$
$$(\Delta t_2^2 \Delta t_1)O(\Delta t^3) + \Delta t_2^3 O(\Delta t^3)$$
$$(\Delta t_2^3 + \Delta t_2^2 \Delta t_1)O(\Delta t^3)$$
$$\Delta t_2^2(\Delta t_2 + \Delta t_1)O(\Delta t^3)$$

The order of accuracy of the method we derived is 2^{nd} order because the smallest order coefficient, with respect to Δt_2, of the error we dropped ($O(\Delta t^3)$) is a Δt_2^2 term.