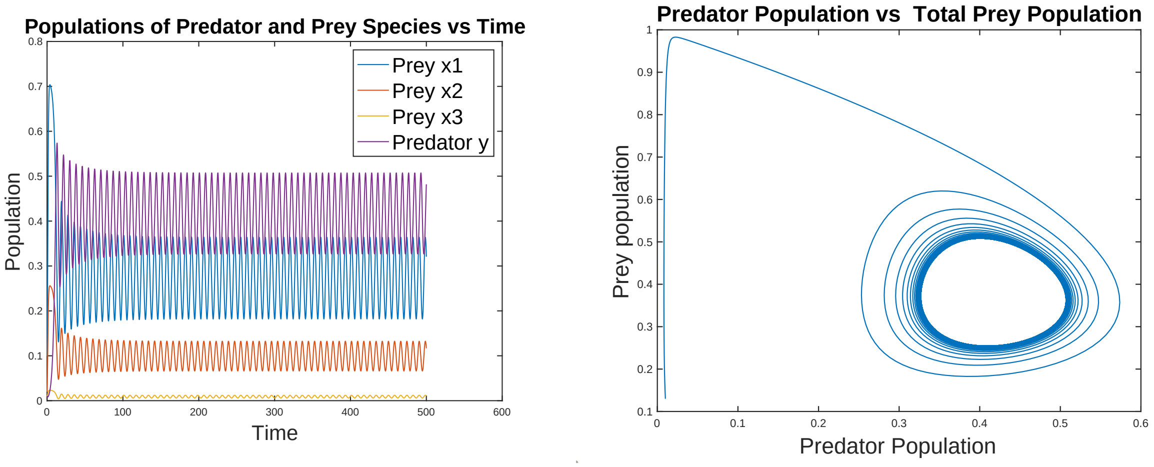
CP2

Numerical Ordinary Differential Equations and their Applications

Declan Mathews [s1610357][B103565]

February 12, 2019

Part 1



(a) A graph showing the populations of each prey species and predator species against time. This was done by Euler's method for a stepsize $h = 0.1$

(b) A graph showing the total prey population versus the predator population. This was also done by Euler's method for a stepsize $h = 0.1$

Figure 1: Part 1 Graphs, in this section "Setup A" from the Checkpoint sheet was used.

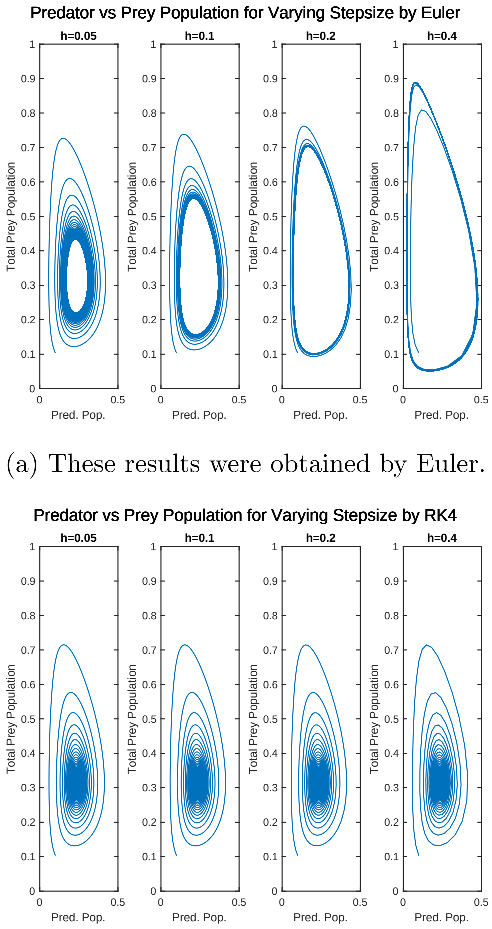
Discussion

The individual populations over time shows a limiting cycle, backed up by the second graph of the prey population versus the predator population. For Figure 1a after roughly 200 on the time axis it shows the same periodic solution, with the predator population increasing and decreasing with a slight lag to the prey species which all rise and fall at roughly the same time. For Figure 1b the spiral continues into a complete circle also representing a limit cycle is achieved.

Part 2

Discussion 1: Figure 2

Figure 2a shows that as the stepsize is increased Eulers method becomes noticeably less accurate and may not reach a limiting cycle at larger h, as the circle at the centre of the spiral is not



(a) These results were obtained by Euler.

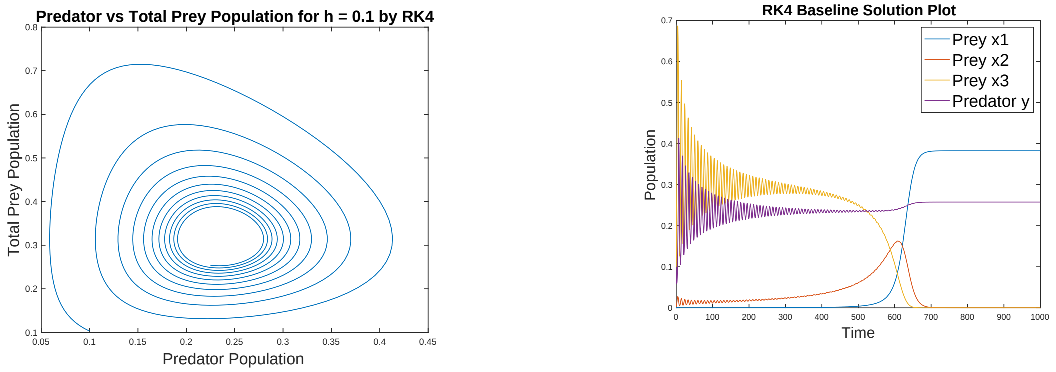
(b) These results were obtained by RK4.

Figure 2: These sets of graphs all show the total prey population versus the predator population for an increasing stepsize value, h. They were achieved by different methods but for the same parameters according to ”Setup B” from the Chechkpoint sheet.

achieved. Figure 2b shows that when the RK4 method is used, a limiting cycle is not achieved but instead converges to a single value, as the spiral continues to converge inward. The four graphs are all very similar showing that the RK4 method is more accurate at higher h values than Euler. This suggests that Euler provides an incorrect idea that a limiting cycle is achieved.

Discussion 2: Figure 3

Figure 3a shows that a limiting cycle has not yet been reached, but appears to almost be reached, as the circle in the centre in beginning to form. Due to the short timescale the process appears to be reaching a limiting cycle. However, looking at Figure 3b we can clearly see that when the solution is done for a longer timescale(and lower h; overall a more accurate solution, and taken as the ’baseline’) that no such limiting solution is achieved and the solutions in fact converge. If we had only looked at this graph up to a time of 200 then we would also be mislead to a possible limiting cycle.

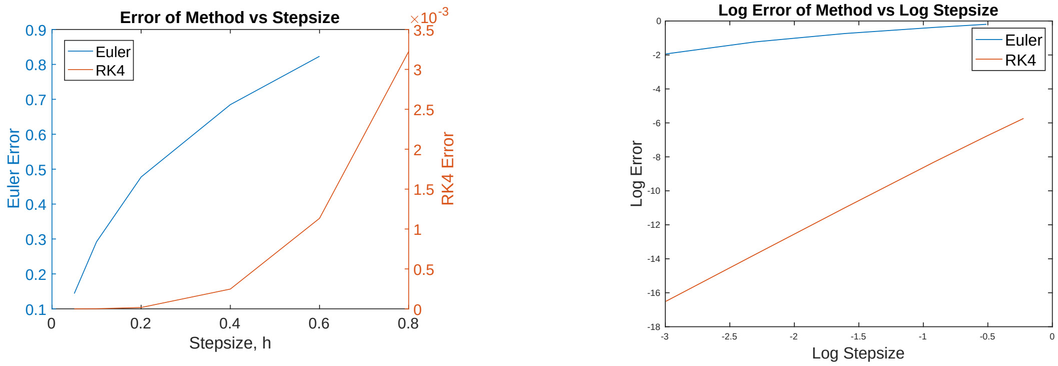


(a) This graph shows the total prey population versus the predator population for a fixed stepsize, $h = 0.1$, and final time of 100 achieved by RK4 for ”Setup B” from the Checkpoint sheet.

(b) A graph showing the populations of the individual species with respect to time. This is the baseline solution with the final time extended to 1000(Setup B, RK4; as by Part 3), provided for explanation and completeness.

Figure 3: The graphs for Part 3, Discussion 2.

Part 3



(a) A graph showing the error of a method versus the stepsize.

(b) A graph showing the log error of a method verses the log of the stepsize

Figure 4: These graphs show the error verus stepsize, h (a) and the log error versus $\log h$ (b). This was done for both Euler and RK4 for ”Setup B” from the Checkpoint sheet.

Discussion

Figure 4a shows that the error increases for both methods as the stepsize is increased. However, the error is much smaller for the RK4 method than the Euler method. The order of accuracy of a method can be found by finding the gradient of the log of its error versus the log of the stepsize. For Figure 4b the gradient for Euler is roughly 1 and the gradient for RK4 is roughly 4, meaning that Euler is 1st order and RK4 is fourth order.

Code Appendix

Part 1

```
% Setup A params and ICs
% m1 = 2; m2 = 2; m3 = 2;
% a= 0.3; sigma = 1.1; K = 1; alpha = 2;
% t0 = 0; tf = 500;
% x10 = 0.1; x20 = 0.02; x30 = 0.01; y0 = 0.01;

% --- P1a Euler ---
h = 0.1;
Method = 'Euler';
t0 = 0; tf = 500;
w0 = [0.1; 0.02; 0.01; 0.01];
Pars.m1 = 2; Pars.m2 = 2; Pars.m3 = 2;
Pars.a = 0.3; Pars.sigma = 1.1; Pars.K = 1; Pars.alpha = 2;
[T,W] = ODEsolver(t0,w0,tf,h,'BAR', Method, Pars);
figure(1)
plot(T,W, 'linewidth', 1)
title('Populations of Predator and Prey Species vs Time', 'FontSize', 20)
xlabel('Time', 'FontSize', 20)
ylabel('Population', 'FontSize', 20)
legend('Prey x1', 'Prey x2', 'Prey x3', 'Predator y', 'FontSize', 20)

% --- P1b Pred vs Prey pop ---
y =W(4,:);
x =W(1,:) +W(2,:) +W(3,:);
figure(2)
plot(y,x, 'linewidth', 1)
title('Predator Population vs  Total Prey Population', 'FontSize', 20)
ylabel('Prey population', 'FontSize', 20)
xlabel('Predator Population', 'FontSize', 20)

fig = gcf; fig.PaperPositionMode = 'auto'; fig_pos = fig.PaperPosition;
fig.PaperSize = [fig_pos(3) fig_pos(4)];
```

Part 2

```
% Setup B params and ICs
% m1 = 2.25; m2 = 2.5; m3 = 2.5;
% a = 0.4; sigma = 1.1; K = 1; alpha = 1.2;
% t0 = 0; tf = 500;
% x10 = 0.001; x20 = 0.002; x30 = 0.1; y0 = 0.1;

% --- P2a Euler Subplots ---
H1 = [0.05; 0.1; 0.2; 0.4];
```

```
Method = 'Euler';
t0 = 0; tf = 500;
w0 = [0.001; 0.002; 0.1; 0.1];
Pars.m1 = 2.25; Pars.m2 = 2.5; Pars.m3 = 2.5;
Pars.a = 0.4; Pars.sigma = 1.1; Pars.K = 1; Pars.alpha = 1.2;

for i=1:4
    [T1,W1] = ODEsolver(t0,w0,tf,H1(i),'BAR', Method, Pars);
    figure(1)
    subplot(1,4,i);
    x = W1(1,:) + W1(2,:) + W1(3,:); y = W1(4,:);
    plot(y, x, 'linewidth', 1);
    title(['h=' num2str(H1(i))]); axis([0 0.5 0 1]);
    sgtitle('Predator vs Prey Population for Varying Stepsize by Euler', 'FontSize', 16)
    xlabel('Pred. Pop.')
    ylabel('Total Prey Population')
end

% --- P2b RK4 Subplots ---
Method = 'RK4';

for i=1:4
    [T2,W2] = ODEsolver(t0,w0,tf,H1(i),'BAR', Method, Pars);
    figure(2)
    subplot(1,4,i);
    x = W2(1,:) + W2(2,:) + W2(3,:); y = W2(4,:);
    plot(y, x, 'linewidth', 1);
    title(['h=' num2str(H1(i))]); axis([0 0.5 0 1]);
    sgtitle('Predator vs Prey Population for Varying Stepsize by RK4', 'FontSize', 16)
    xlabel('Pred. Pop.')
    ylabel('Total Prey Population')
end

% --- P2c RK4 fixed h ---
h = 0.1;
tf = 1000;

% Pred vs individual species plots, as per hint, for understanding
for i=1:3
    [T3,W3] = ODEsolver(t0,w0,tf,h,'BAR', Method, Pars);
    figure(3)
    subplot(1,3,i);
    x = W3(i,:); y = W3(4,:);
    plot(y, x, 'linewidth', 1);
    title(['y vs x' num2str(i)]); axis([0 0.7 0 0.7]);
end
```

```matlab

% Total prey vs pred
tf = 100;

[T4,W4] = ODEsolver(t0,w0,tf,h,'BAR', Method, Pars);
x = W4(1,:) + W4(2,:) + W4(3,:); y = W4(4,:);
figure(4)
plot(y, x, 'linewidth', 1);
title('Predator vs Total Prey Population for h = 0.1 by RK4', 'FontSize', 18)
ylabel('Total Prey Population', 'FontSize', 18)
xlabel('Predator Population', 'FontSize', 18)

fig = gcf; fig.PaperPositionMode = 'auto'; fig_pos = fig.PaperPosition;
fig.PaperSize = [fig_pos(3) fig_pos(4)];
```

Part 3

```matlab
% Setup B params and ICs
% m1 = 2.25; m2 = 2.5; m3 = 2.5;
% a = 0.4; sigma = 1.1; K = 1; alpha = 1.2;
% t0 = 0; tf = 500;
% x10 = 0.001; x20 = 0.002; x30 = 0.1; y0 = 0.1;

% --- P3a Baseline Sol ---
h = 0.01;
Method = 'RK4';
t0 = 0; tf = 1000;
w0 = [0.001; 0.002; 0.1; 0.1];
Pars.m1 = 2.25; Pars.m2 = 2.5; Pars.m3 = 2.5;
Pars.a = 0.4; Pars.sigma = 1.1; Pars.K = 1; Pars.alpha = 1.2;

[Tref,Wref] = ODEsolver(t0,w0,tf,h,'BAR', Method, Pars);
Yref = Wref(4,:);

% Baseline solution plot for clarity and explanation
figure(1)
plot(Tref, Wref, 'linewidth', 1)
title('RK4 Baseline Solution Plot', 'FontSize', 18)
xlabel('Time', 'FontSize', 18)
ylabel('Population', 'FontSize', 18)
legend('Prey x1', 'Prey x2', 'Prey x3', 'Predator y', 'FontSize', 18)


% --- P3b Error Calc ---
H = [0.05; 0.1; 0.2; 0.4; 0.6; 0.8];
E1 = [];
Method = 'Euler';
```

```matlab

for i=1:6
    [T,W] = ODEsolver(t0,w0,tf,H(i),'BAR', Method, Pars);
    Emax = ErrorCalc(Tref, Wref, T, W);
    E1(i) = Emax;
end

figure(2)
y = E1;
yyaxis left
plot(H,E1, 'linewidth', 1)
ylabel('Euler Error', 'FontSize', 18)
hold on;
E2 = [];
Method = 'RK4';

for i=1:6
    [T,W] = ODEsolver(t0,w0,tf,H(i),'BAR', Method, Pars);
    Emax = ErrorCalc(Tref, Wref, T, W);
    E2(i) = Emax;
end

y = E2;
yyaxis right
plot(H,E2, 'linewidth', 1)
title('Error of Method vs Stepsize', 'FontSize', 18)
xlabel('Stepsize, h', 'FontSize', 18)
ylabel('RK4 Error', 'FontSize', 18)
legend('Euler', 'RK4', 'FontSize', 18)
hold off;

% --- P3c Log error graphs ---
logH = log(H);
logE1 = log(E1);
logE2 = log(E2);

figure(3)
plot(logH, logE1, 'linewidth', 1)
hold on;
plot(logH, logE2, 'linewidth', 1)
title('Log Error of Method vs Log Stepsize', 'FontSize', 18)
xlabel('Log Stepsize', 'FontSize', 18)
ylabel('Log Error', 'FontSize', 18)
legend('Euler', 'RK4', 'FontSize', 18)

fig = gcf; fig.PaperPositionMode = 'auto'; fig_pos = fig.PaperPosition;
fig.PaperSize = [fig_pos(3) fig_pos(4)];
```