Math 342: Homework 5

Connor Emmons

Documentation: I used ChatGPT solely for looking up Latex commands. The main Homework 5 MatLab script and all required dependencies are located in the Homework 5 folder found here: https://github.com/Connor-Lemons/Emmons-Math-342. No other resources used.

Problem 1

1b $y' = 1 + (t-y)^2; \quad 2 \le t \le 3; \quad y(2) = 1; \quad h = 0.5$

$y(2) = 1$
$y(2.5) \approx y(2) + 0.5\, f(2, y(2)) = 1 + 0.5(2) = 2$
$y(3) \approx y(2.5) + 0.5\, f(2.5, y(2.5)) = 2 + 0.5(1.25) = \boxed{2.625}$

3b $y(t) = t + \frac{1}{1-t}$ $\qquad f(t,y) = 1 + (t-y)^2$

$y' = 1 + (1-t)^{-2}$ $\qquad$ Error bound @ step 1: $\left|\frac{h^2}{2} y''(\xi)\right| = \left|\frac{0.5^2}{2} 2(1-2)^{-3}\right| = 0.25$
$y'' = 2(1-t)^{-3}$ $\qquad$ @ step 2: same as step 1

$y(2.5) = 1.8333$ $\qquad |y(3) - 2.625| = 0.125 < 0.25$
$y(3) = 2.5$ $\qquad |y(2.5) - 2| = 0.1666 < 0.25$

Problem 3

$y' = 1 + (t-y)^2; \quad 2 \le t \le 3; \quad y(2) = 1; \quad h = 0.5$

Mod Euler

$w_0 = 1$ —
$w_1 = 1 + \frac{0.5}{2}\left(f(2,1) + f(2.5, 1 + 0.5 f(2,1))\right) = 1.8125$ —
$w_2 = 1.8125 + \frac{0.5}{2}\left(f(2.5, 1.8125) + f(3, 1.8125 + 0.5 f(2.5, 1.8125))\right) = 2.48155\,3078$ —

Midpoint

$w_0 = 1$ —
$w_1 = 1 + 0.5 f(2.25, 1 + 0.25 f(2,1)) = 1.78125$ —
$w_2 = 1.78125 + 0.5 f(2.75, 1.78125 + 0.25 f(2.5, 1.78125)) = 2.45506385$ —

RK4

$w_0 = 1$ —
$k_1 = 0.5 f(2,1) = 1$
$k_2 = 0.5 f(2.25, 1 + \frac{1}{2}) = 0.78125$
$k_3 = 0.5 f(2.25, 1 + \frac{1}{2}(0.78125)) = 0.8692626953$
$k_4 = 0.5 f(2.5, 1 + k_3) = 0.698914773$
$w_1 = 1 + 1/6 (k_1 + 2k_2 + 2k_3 + k_4) = 1.833323361$ —
$k_1' = 0.5 f(2.5, w_1) = 0.7222288707$
$k_2 = 0.5 f(2.75, w_1 + 1/2 k_1) = 0.6543246012$
$k_3 = 0.5 f(2.75, w_1 + 1/2 k_2) = 0.6737638846$
$k_4 = 0.5 f(3, w_1 + k_3) = 0.6214481547$
$w_2 = k_1 + 1/6 (k_1 + 2k_2 + 2k_3 + k_4) = 2.49997 1193$ —

	error Mod Euler	error midpoint	error RK4
$t=2$	0	0	0
$t=2.5$	$0.0208\bar{3}$	$0.05208\bar{3}$	$9.97233333 \times 10^{-2}$
$t=3$	0.01844692	0.04493615	2.8807×10^{-5}

Problem 2

The output of the code is:

t_i	w_i	y_i	|y_i-w_i|
1	-1	-1	0
1.05	-0.95	-0.952381	0.002380952
1.1	-0.9045354	-0.9090909	0.004555478
1.15	-0.8630071	-0.8695652	0.00655813
1.2	-0.8249169	-0.8333333	0.008416415
1.25	-0.7898476	-0.8	0.01015245
1.3	-0.7574466	-0.7692308	0.01178416
1.35	-0.7274145	-0.7407407	0.01332622
1.4	-0.699495	-0.7142857	0.01479072
1.45	-0.6734675	-0.6896552	0.01618769
1.5	-0.6491412	-0.6666667	0.01752549
1.55	-0.6263501	-0.6451613	0.01881116
1.6	-0.6049494	-0.625	0.02005064
1.65	-0.5848116	-0.6060606	0.02124898
1.7	-0.5658248	-0.5882353	0.02241047
1.75	-0.5478898	-0.5714286	0.02353881
1.8	-0.5309184	-0.5555556	0.02463716
1.85	-0.5148323	-0.5405405	0.02570826
1.9	-0.4995613	-0.5263158	0.02675448
1.95	-0.4850426	-0.5128205	0.0277779
2	-0.4712197	-0.5	0.0287803

Code can be found attached at the end or in the GitHub.

The error is stable because each step increases the error by roughly 0.002.

In order to find the Lipschitz constant, take the derivative of $f(t,y)$ with respect to y:

$$\frac{df}{dy} = -\frac{1}{t} - 2y \qquad (1)$$

Because $y = -\frac{1}{t}$,

$$\frac{df}{dy} = -\frac{1}{t} - 2\left(-\frac{1}{t}\right) = \frac{1}{t} \qquad (2)$$

On the interval $t \in [1,2]$, the following inequality is true:

$$\frac{1}{t} \leq 1 \qquad (3)$$

The second derivative of y is:

$$y'' = -\frac{2}{t^3} \qquad (4)$$

On the interval $t \in [1,2]$, the following inequality is true:

$$|y''| < 2 \qquad (5)$$

Thus $L = 1$ and $M = 2$. Equation 5.10 gives:

$$0.05 = \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right] \qquad (6)$$

Plugging in and solving for h yields $h = 0.0291$.

Problem 4

The output of the code is:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2052405	0.2051118	0.0001287441
0.5	0.2774767	0.2773617	0.0001149945
0.6	0.3766981	0.3765957	0.0001023885
0.7	0.5001579	0.5000658	9.215388e-05
0.8	0.6461896	0.6461052	8.437549e-05
0.9	0.8137817	0.813703	7.870457e-05
1	1.002321	1.002246	7.468666e-05

Code can be found attached at the end or in Github.

Problem 5

RK4 output:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2052405	0.2051118	0.0001287441
0.5	0.2774767	0.2773617	0.0001149945
0.6	0.3766981	0.3765957	0.0001023885
0.7	0.5001579	0.5000658	9.215388e-05
0.8	0.6461896	0.6461052	8.437549e-05
0.9	0.8137817	0.813703	7.870457e-05
1	1.002321	1.002246	7.468666e-05

Adams-Bashforth 2-Step:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1739041	0.1626265	0.0112776
0.3	0.1740468	0.1643767	0.009670046
0.4	0.2144877	0.2051118	0.009375951
0.5	0.2846336	0.2773617	0.007271953
0.6	0.3822803	0.3765957	0.005684643
0.7	0.5042285	0.5000658	0.004162694
0.8	0.6491272	0.6461052	0.003021992
0.9	0.8158389	0.813703	0.002135924
1	1.003742	1.002246	0.00149555

Adams-Bashforth 3-Step:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1605261	0.1643767	0.003850613
0.4	0.2026399	0.2051118	0.002471824
0.5	0.2732179	0.2773617	0.004143734
0.6	0.3747011	0.3765957	0.00189459
0.7	0.4972078	0.5000658	0.002857948
0.8	0.645264	0.6461052	0.0008412219
0.9	0.8119618	0.813703	0.001741164
1	1.002089	1.002246	0.0001565466

Adams-Bashforth 4-Step:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2066057	0.2051118	0.001493983
0.5	0.2780929	0.2773617	0.0007312636
0.6	0.378768	0.3765957	0.002172342
0.7	0.4998405	0.5000658	0.0002253161
0.8	0.6487176	0.6461052	0.002612367
0.9	0.8116247	0.813703	0.002078325
1	1.006412	1.002246	0.004166082

Adams-Bashforth 5-Step:

t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2052405	0.2051118	0.0001287441
0.5	0.2769031	0.2773617	0.0004585888
0.6	0.3765206	0.3765957	7.503962e-05
0.7	0.4988777	0.5000658	0.001188087
0.8	0.6471458	0.6461052	0.001040593
0.9	0.8107178	0.813703	0.002985168
1	1.007335	1.002246	0.005088809

Adams-Bashforth 4-Step with Predictor:

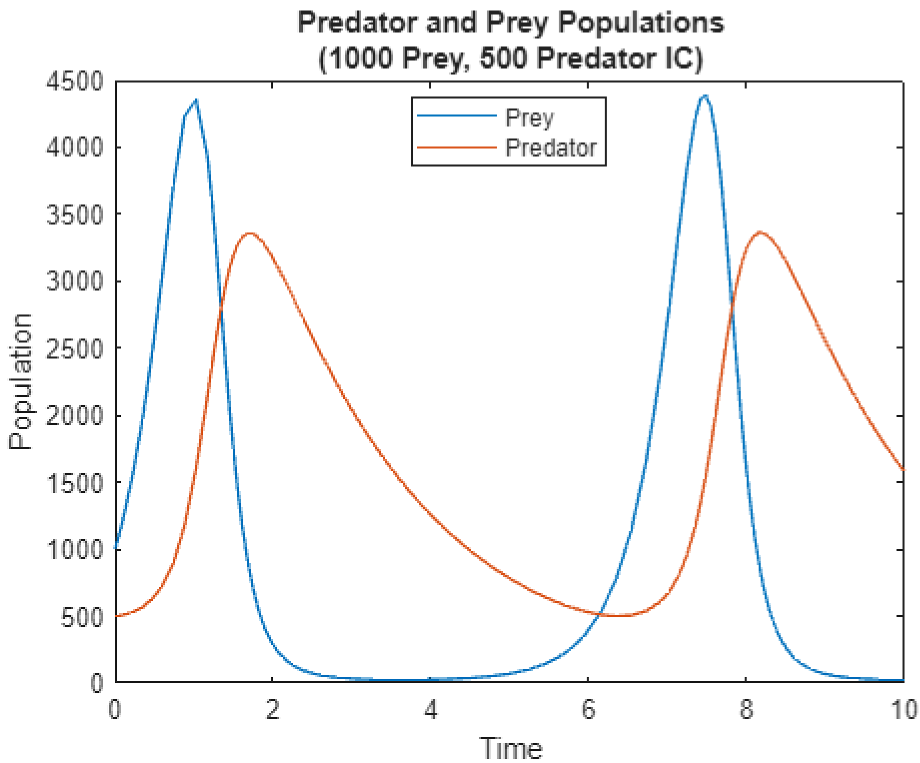
t_i	w_i	y_i	|y_i-w_i|
0	0.3333333	0.3333333	0
0.1	0.212283	0.2121769	0.0001060995
0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2048557	0.2051118	0.0002560885
0.5	0.2769896	0.2773617	0.0003721159
0.6	0.3762804	0.3765957	0.0003153045
0.7	0.4998012	0.5000658	0.0002645671
0.8	0.6458949	0.6461052	0.0002102884
0.9	0.8135498	0.813703	0.0001532097
1	1.002137	1.002246	0.0001087606

Code can be found attached at the end or in Github.

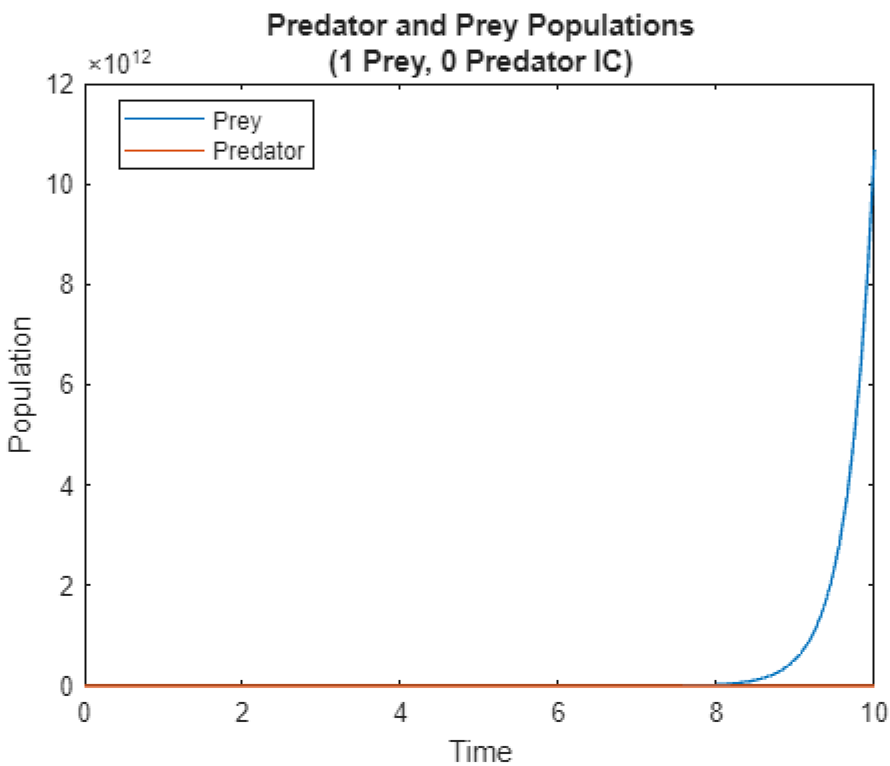
In this specific case, RK4 produced the best result with a final error of 7.47e-5.

Problem 6

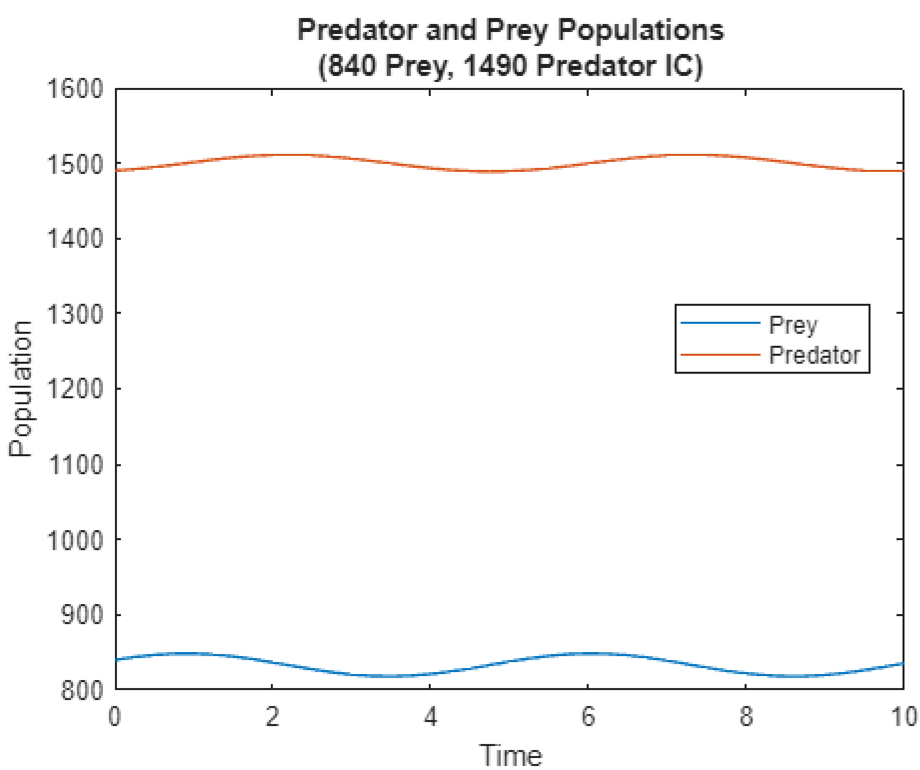
This is the graph for the 1000 prey, 500 predators case:



Note that, at least over the course of 10 units of time, the system does not reach any sort of equilibrium. To find the equilibrium points, solve for $x_1' = x_2' = 0$. This gives two potential points: $(0,0)$ and $(833.333,1500)$. To test if these equilibrium points are stable, run the solver with a small deviation to these conditions. This is the graph for 1 prey, 0 predators (representing a small deviation from the case $(0,0)$):



Note that the population blows up, and thus this is not a stable point. This is the graph for 840 prey, 1490 predators (representing a small deviation from the case $(833.333{,}1500)$):



Note that the populations of each species oscillate around a constant value, which indicates that this point is marginally stable (a truly stable point would drive both populations to a constant value).

Homework 5

Problem 2 (10a, 10c)

```
clear; clc;

syms t y

f(t, y) = 1/t^2 - y/t - y^2;
g(t) = -1/t;
a = 1;
b = 2;
N = 20;
alpha = -1;

[t, w] = EulerMethod(f, a, b, N, alpha);

fprintf("%-15s%-15s%-15s%-15s\n", "t_i", "w_i", "y_i", "|y_i-w_i|")
```

```
t_i            w_i            y_i            |y_i-w_i|
```

```
for i = 1:length(t)

    fprintf("%-15.7g%-15.7g%-15.7g%-15.7g\n", t(i), w(i), g(t(i)), abs(g(t(i)) -
w(i)))

end
```

```
1              -1             -1             0
1.05           -0.95          -0.952381      0.002380952
1.1            -0.9045354     -0.9090909     0.004555478
1.15           -0.8630071     -0.8695652     0.00655813
1.2            -0.8249169     -0.8333333     0.008416415
1.25           -0.7898476     -0.8           0.01015245
1.3            -0.7574466     -0.7692308     0.01178416
1.35           -0.7274145     -0.7407407     0.01332622
1.4            -0.699495      -0.7142857     0.01479072
1.45           -0.6734675     -0.6896552     0.01618769
1.5            -0.6491412     -0.6666667     0.01752549
1.55           -0.6263501     -0.6451613     0.01881116
1.6            -0.6049494     -0.625         0.02005064
1.65           -0.5848116     -0.6060606     0.02124898
1.7            -0.5658248     -0.5882353     0.02241047
1.75           -0.5478898     -0.5714286     0.02353881
1.8            -0.5309184     -0.5555556     0.02463716
1.85           -0.5148323     -0.5405405     0.02570826
1.9            -0.4995613     -0.5263158     0.02675448
1.95           -0.4850426     -0.5128205     0.0277779
2              -0.4712197     -0.5           0.0287803
```

```
diff(g, 2)
```

```
ans(t) =
```

$-\frac{2}{t^3}$

```
syms h

L = 1
```

```
L = 1
```

```
M = 2
```

```
M = 2
```

```
eq = 0.05 == (h*M)/(2*L)*(exp(L*(t(end) - a)) - 1);
h_sol = solve(eq, h)
```

h_sol = 0.0291

Problem 4

```
clear; clc;

syms t y

f(t, y) = -5*y + 5*t^2 + 2*t
```

f(t, y) = $5\,t^2+2\,t-5\,y$

```
g(t) = t^2 + (1/3)*exp(-5*t);
a = 0;
b = 1;
N = 10;
alpha = 1/3;

[t, w] = RK4(f, a, b, N, alpha);

fprintf("%-15s%-15s%-15s%-15s\n", "t_i", "w_i", "y_i", "|y_i-w_i|")
```

```
t_i            w_i            y_i            |y_i-w_i|
```

```
for i = 1:length(t)

    fprintf("%-15.7g%-15.7g%-15.7g%-15.7g\n", t(i), w(i), g(t(i)), abs(g(t(i)) -
w(i)))

end
```

```
0              0.3333333      0.3333333      0
0.1            0.212283       0.2121769      0.0001060995
```

```
0.2           0.1627655      0.1626265      0.0001389773
0.3           0.1645165      0.1643767      0.0001398207
0.4           0.2052405      0.2051118      0.0001287441
0.5           0.2774767      0.2773617      0.0001149945
0.6           0.3766981      0.3765957      0.0001023885
0.7           0.5001579      0.5000658      9.215388e-05
0.8           0.6461896      0.6461052      8.437549e-05
0.9           0.8137817      0.813703       7.870457e-05
1             1.002321       1.002246       7.468666e-05
```

Problem 5

```
clear; clc;

syms t y

f(t, y) = -5*y + 5*t^2 + 2*t
```

f(t, y) = $5\,t^2+2\,t-5\,y$

```
g(t) = t^2 + (1/3)*exp(-5*t);
a = 0;
b = 1;
N = 10;
alpha = 1/3;

[t, w_2step, w_3step, w_4step, w_5step] = AdamBashforthMethod(f, a, b, N, alpha);
w_step = [w_2step; w_3step; w_4step; w_5step]
```

```
w_step = 4×11
    0.3333    0.2123    0.1739    0.1740    0.2145    0.2846    0.3823    0.5042 ···
    0.3333    0.2123    0.1628    0.1605    0.2026    0.2732    0.3747    0.4972
    0.3333    0.2123    0.1628    0.1645    0.2066    0.2781    0.3788    0.4998
    0.3333    0.2123    0.1628    0.1645    0.2052    0.2769    0.3765    0.4989
```

```

for j = 1:4
    w = w_step(j,:);
    fprintf("%-15s%-15s%-15s%-15s\n", "t_i", "w_i", "y_i", "|y_i-w_i|")
    for i = 1:length(t)

        fprintf("%-15.7g%-15.7g%-15.7g%-15.7g\n", t(i), w(i), g(t(i)), abs(g(t(i))
- w(i)))

    end
end
```

```
t_i           w_i            y_i            |y_i-w_i|
0             0.3333333      0.3333333      0
0.1           0.212283       0.2121769      0.0001060995
0.2           0.1739041      0.1626265      0.0112776
0.3           0.1740468      0.1643767      0.009670046
0.4           0.2144877      0.2051118      0.009375951
0.5           0.2846336      0.2773617      0.007271953
0.6           0.3822803      0.3765957      0.005684643
```

```
0.7            0.5042285      0.5000658      0.004162694
0.8            0.6491272      0.6461052      0.003021992
0.9            0.8158389      0.813703       0.002135924
1              1.003742       1.002246       0.00149555
t_i            w_i            y_i            |y_i-w_i|
0              0.3333333      0.3333333      0
0.1            0.212283       0.2121769      0.0001060995
0.2            0.1627655      0.1626265      0.0001389773
0.3            0.1605261      0.1643767      0.003850613
0.4            0.2026399      0.2051118      0.002471824
0.5            0.2732179      0.2773617      0.004143734
0.6            0.3747011      0.3765957      0.00189459
0.7            0.4972078      0.5000658      0.002857948
0.8            0.645264       0.6461052      0.0008412219
0.9            0.8119618      0.813703       0.001741164
1              1.002089       1.002246       0.0001565466
t_i            w_i            y_i            |y_i-w_i|
0              0.3333333      0.3333333      0
0.1            0.212283       0.2121769      0.0001060995
0.2            0.1627655      0.1626265      0.0001389773
0.3            0.1645165      0.1643767      0.0001398207
0.4            0.2066057      0.2051118      0.001493983
0.5            0.2780929      0.2773617      0.0007312636
0.6            0.378768       0.3765957      0.002172342
0.7            0.4998405      0.5000658      0.0002253161
0.8            0.6487176      0.6461052      0.002612367
0.9            0.8116247      0.813703       0.002078325
1              1.006412       1.002246       0.004166082
t_i            w_i            y_i            |y_i-w_i|
0              0.3333333      0.3333333      0
0.1            0.212283       0.2121769      0.0001060995
0.2            0.1627655      0.1626265      0.0001389773
0.3            0.1645165      0.1643767      0.0001398207
0.4            0.2052405      0.2051118      0.0001287441
0.5            0.2769031      0.2773617      0.0004585888
0.6            0.3765206      0.3765957      7.503962e-05
0.7            0.4988777      0.5000658      0.001188087
0.8            0.6471458      0.6461052      0.001040593
0.9            0.8107178      0.813703       0.002985168
1              1.007335       1.002246       0.005088809
```

```
[t, w_4step] = AdamBashforthPredictorMethod(f, a, b, N, alpha);

w = w_4step;

fprintf("%-15s%-15s%-15s%-15s\n", "t_i", "w_i", "y_i", "|y_i-w_i|")
```

```
t_i            w_i            y_i            |y_i-w_i|
```

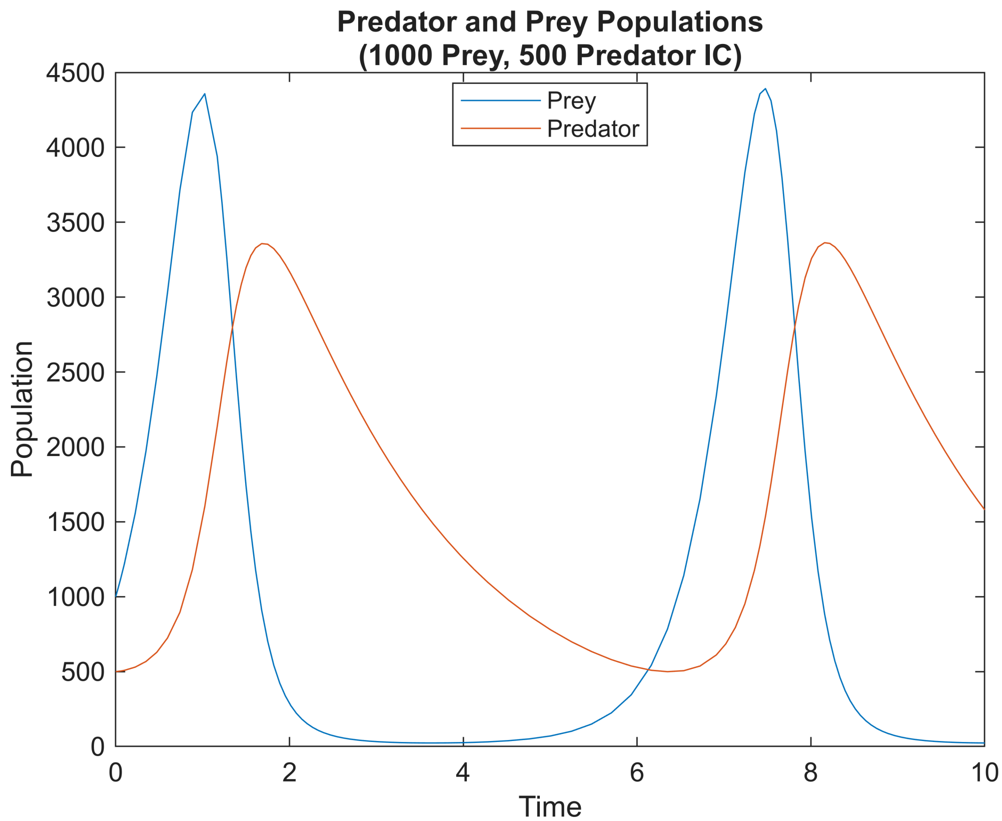
```
for i = 1:length(t)

    fprintf("%-15.7g%-15.7g%-15.7g%-15.7g\n", t(i), w(i), g(t(i)), abs(g(t(i)) -
w(i)))

end
```

```
0              0.3333333      0.3333333      0
0.1            0.212283       0.2121769      0.0001060995
```

0.2	0.1627655	0.1626265	0.0001389773
0.3	0.1645165	0.1643767	0.0001398207
0.4	0.2048557	0.2051118	0.0002560885
0.5	0.2769896	0.2773617	0.0003721159
0.6	0.3762804	0.3765957	0.0003153045
0.7	0.4998012	0.5000658	0.0002645671
0.8	0.6458949	0.6461052	0.0002102884
0.9	0.8135498	0.813703	0.0001532097
1	1.002137	1.002246	0.0001087606

Problem 6

```
clear; clc;

k1 = 3;
k2 = 0.002;
k3 = 0.0006;
k4 = 0.5;
tspan = [0 10];
x0 = [1000 500];

[t, x] = ode45(@(t,x) odefcn(t, x, k1, k2, k3, k4), tspan, x0);

figure
plot(t, x(:,1), t, x(:,2))
xlabel("Time")
ylabel("Population")
legend("Prey", "Predator", "Location", "best")
title(["Predator and Prey Populations", "(1000 Prey, 500 Predator IC)"])
```




```
syms x1 x2

eq1 = 0 == k1*x1 - k2*x1*x2;
eq2 = 0 == k3*x1*x2 - k4*x2;
[x1_sol, x2_sol] = solve([eq1, eq2], [x1, x2])
```

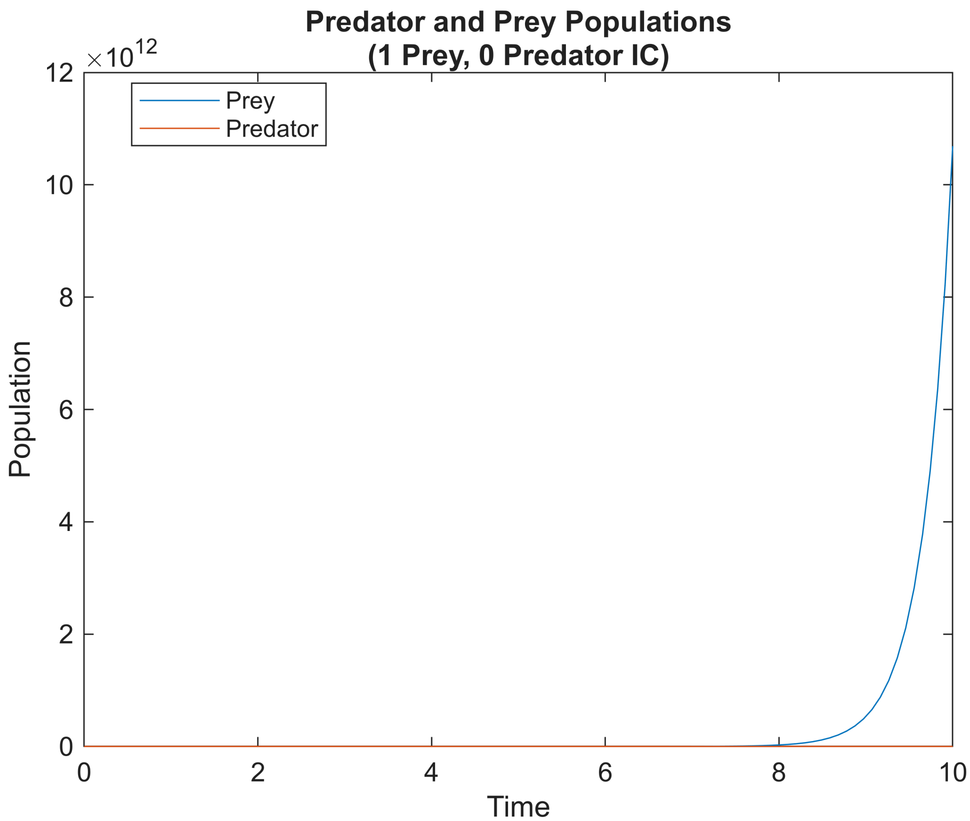
x1_sol =

$$\begin{pmatrix} 0 \\ 833.3333 \end{pmatrix}$$

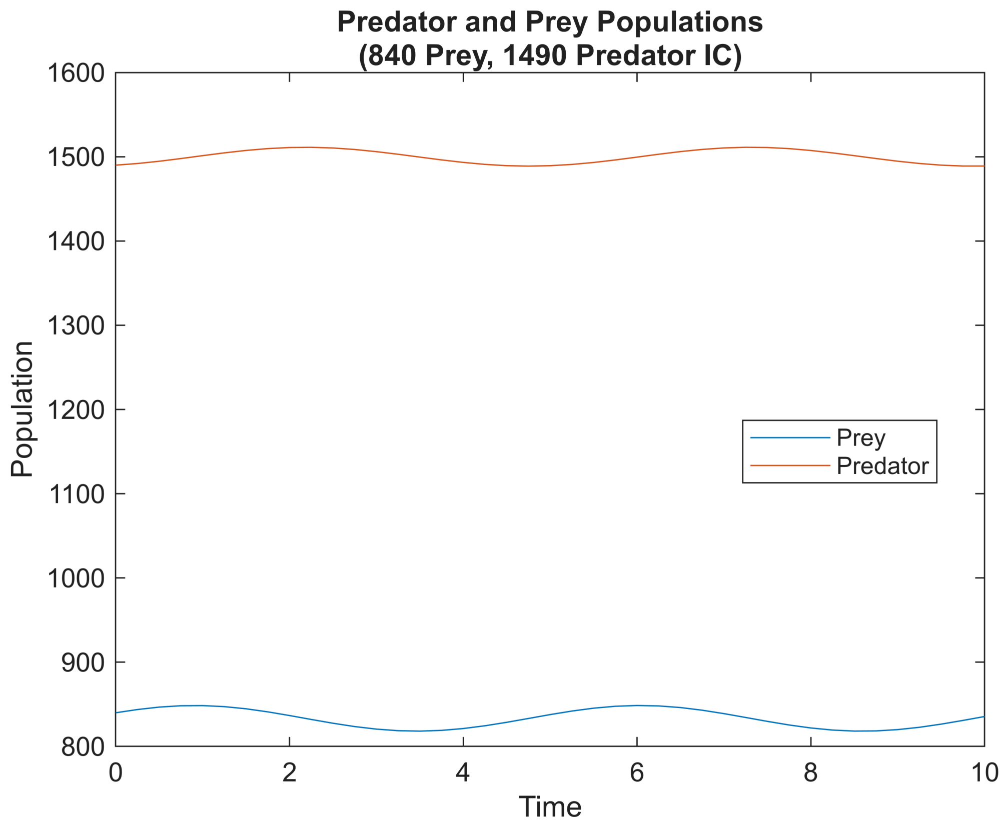
x2_sol =

$$\begin{pmatrix} 0 \\ 1500 \end{pmatrix}$$

```
x0 = [1 0];

[t, x] = ode45(@(t,x) odefcn(t, x, k1, k2, k3, k4), tspan, x0);

figure
plot(t, x(:,1), t, x(:,2))
xlabel("Time")
ylabel("Population")
legend("Prey", "Predator", "Location", "best")
title(["Predator and Prey Populations", "(1 Prey, 0 Predator IC)"])
```




```
x0 = [840 1490];

[t, x] = ode45(@(t,x) odefcn(t, x, k1, k2, k3, k4), tspan, x0);

figure
plot(t, x(:,1), t, x(:,2))
xlabel("Time")
ylabel("Population")
legend("Prey", "Predator", "Location", "best")
title(["Predator and Prey Populations", "(840 Prey, 1490 Predator IC)"])
```




```
function dxdt = odefcn(t, x, k1, k2, k3, k4)

    dxdt = zeros(2,1);
    dxdt(1) = k1*x(1) - k2*x(1)*x(2);
    dxdt(2) = k3*x(1)*x(2) - k4*x(2);

end
```

```
function [t, w] = EulerMethod(f, a, b, N, alpha)

    syms t y

    h = (b - a)/N;
    t(1) = a;
    w(1) = alpha;

    for i = 1:N

        w(i+1) = w(i) + h*f(t(i), w(i));
        t(i+1) = a + i*h;

    end

end
```

```
function [t, w] = RK4(f, a, b, N, alpha)

    syms t y

    h = (b - a)/N;
    t(1) = a;
    w(1) = alpha;

    for i = 1:N



        K1 = h*f(t(i), w(i));
        K2 = h*f(t(i) + h/2, w(i) + K1/2);
        K3 = h*f(t(i) + h/2, w(i) + K2/2);
        K4 = h*f(t(i) + h, w(i) + K3);

        w(i+1) = w(i) + (K1 + 2*K2 + 2*K3 + K4)/6;
        t(i+1) = a + i*h;

    end

end
```

```
function [t, w_2step, w_3step, w_4step, w_5step] = AdamBashforthMethod(f, a, b, N, alpha)

    syms t y

    h = (b - a)/N;
    t(1) = a;
    w(1) = alpha;

    for i = 1:4

        K1 = h*f(t(i), w(i));
        K2 = h*f(t(i) + h/2, w(i) + K1/2);
        K3 = h*f(t(i) + h/2, w(i) + K2/2);
        K4 = h*f(t(i) + h, w(i) + K3);

        w(i+1) = w(i) + (K1 + 2*K2 + 2*K3 + K4)/6;
        t(i+1) = a + i*h;

    end

    w_2step = w;
    w_3step = w;
    w_4step = w;
    w_5step = w;

    for i = 2:N

        t(i+1) = a + i*h;
        w_2step(i+1) = w_2step(i) + h*(3*f(t(i), w_2step(i)) - f(t(i-1), w_2step(i-1)))↙
/2;

    end

    for i = 3:N

        t(i+1) = a + i*h;
        w_3step(i+1) = w_3step(i) + h*(23*f(t(i), w_3step(i)) - 16*f(t(i-1), w_3step(i-↙
1)) + 5*f(t(i-2), w_3step(i-2)))/12;

    end

    for i = 4:N

        t(i+1) = a + i*h;
        w_4step(i+1) = w_4step(i) + h*(55*f(t(i), w_4step(i)) - 59*f(t(i-1), w_4step(i-↙
1)) + 37*f(t(i-2), w_4step(i-2)) - 9*f(t(i-3), w_4step(i-3)))/24;

    end

    for i = 5:N
```

```
        t(i+1) = a + i*h;
        w_5step(i+1) = w_5step(i) + h*(1901*f(t(i), w_5step(i)) - 2774*f(t(i-1), w_5step↙
(i-1)) + 2616*f(t(i-2), w_5step(i-2)) - 1274*f(t(i-3), w_5step(i-3)) + 251*f(t(i-4),↙
w_5step(i-4)))/720;

    end

end
```

```
function [t, w_4step] = AdamBashforthPredictorMethod(f, a, b, N, alpha)

    syms t y

    h = (b - a)/N;
    t(1) = a;
    w(1) = alpha;

    for i = 1:3

        K1 = h*f(t(i), w(i));
        K2 = h*f(t(i) + h/2, w(i) + K1/2);
        K3 = h*f(t(i) + h/2, w(i) + K2/2);
        K4 = h*f(t(i) + h, w(i) + K3);

        w(i+1) = w(i) + (K1 + 2*K2 + 2*K3 + K4)/6;
        t(i+1) = a + i*h;

    end

    w_4step = w;

    for i = 4:N

        t(i+1) = a + i*h;

        w_temp = w_4step(i) + h*(55*f(t(i), w_4step(i)) - 59*f(t(i-1), w_4step(i-1)) +↙
37*f(t(i-2), w_4step(i-2)) - 9*f(t(i-3), w_4step(i-3)))/24;
        w_4step(i+1) = w_4step(i) + h*(9*f(t(i+1), w_temp) + 19*f(t(i), w_4step(i)) - 5*f↙
(t(i-1), w_4step(i-1)) + f(t(i-2), w_4step(i-2)))/24;

    end

end
```