MATH 308
Spring 2016
Matlab Project

Ethan Vieira

Partners:
Akash Kundu, Aisya Abir

Introduction to the Code

Most sections are comprised into a single function that will take inputs that include the initial conditions and step size(s). For example:

```
% 1.1 Euler's Method

yE(0, 2, 1, .01) % (T initial, T final, Y initial, step size(s) (if entering an array, [h1 h2 h3...])
```

This function, yE, will compute Euler's method with those inputs for the differential equation given in the assignment, printing the results for t values with no more than 1 decimal place.

```
        if numel(num2str(t(i))) <= 3
            fprintf('%-2g  \t %-8.6f \n', t(i), y(i))
```

1.1 Euler's Method

The results are accurate when compared with the table from the book.

	Euler		Improved Euler	
t	$h = 0.01$	$h = 0.001$	$h = 0.025$	$h = 0.01$
0	1.0000000	1.0000000	1.0000000	1.0000000
0.1	1.5952901	1.6076289	1.6079462	1.6088585
0.2	2.4644587	2.5011159	2.5020618	2.5047827
0.3	3.7390345	3.8207130	3.8228282	3.8289146
0.4	5.6137120	5.7754845	5.7796888	5.7917911
0.5	8.3766865	8.6770692	8.6849039	8.7074637
1.0	60.037126	64.382558	64.497931	64.830722
1.5	426.40818	473.55979	474.83402	478.51588
2.0	3029.3279	3484.1608	3496.6702	3532.8789

```
h = 0.010 Euler Method
----------------------------------------------
0        1.000000
0.1      1.595290
0.2      2.464459
0.3      3.739035
0.4      5.613712
0.5      8.376686
0.6      12.454558
0.7      18.478797
0.8      27.384136
0.9      40.554208
1        60.037126
1.1      88.864597
1.2      131.524291
1.3      194.659053
1.4      288.101916
1.5      426.408176
1.6      631.123219
1.7      934.139486
1.8      1382.665578
1.9      2046.581755
2        3029.327877
```

1.2 Improved Euler's Method

Comparing with the same table above, the results are accurate.

```
h = 0.010 Improved Euler Method
----------------------------------------------
0        1.000000
0.1      1.608858
0.2      2.504783
0.3      3.828915
0.4      5.791791
0.5      8.707464
0.6      13.044394
0.7      19.501373
0.8      29.120764
0.9      43.457432
1        64.830722
1.1      96.700331
1.2      144.226989
1.3      215.108799
1.4      320.828796
1.5      478.515883
1.6      713.720730
1.7      1064.556518
1.8      1587.875542
1.9      2368.482683
2        3532.878861
```

1.3 Runge-Kutta Method

The results are accurate.

	Improved Euler	Runge–Kutta		
t	$h = 0.025$	$h = 0.2$	$h = 0.1$	$h = 0.05$
0	1.0000000	1.0000000	1.0000000	1.0000000
0.1	1.6079462		1.6089333	1.6090338
0.2	2.5020618	2.5016000	2.5050062	2.5053060
0.3	3.8228282		3.8294145	3.8300854
0.4	5.7796888	5.7776358	5.7927853	5.7941197
0.5	8.6849039		8.7093175	8.7118060
1.0	64.497931	64.441579	64.858107	64.894875
1.5	474.83402		478.81928	479.22674
2.0	3496.6702	3490.5574	3535.8667	3539.8804

```
---------------------------------------------
h = 0.050 Runge-Kutta Method
---------------------------------------------
0        1.000000
0.1      1.609034
0.2      2.505306
0.3      3.830085
0.4      5.794120
0.5      8.711806
0.6      13.052167
0.7      19.514901
0.8      29.143827
0.9      43.496137
1        64.894875
1.1      96.805601
1.2      144.398301
1.3      215.385649
1.4      321.273555
1.5      479.226738
1.6      714.851839
1.7      1066.349301
1.8      1590.707232
1.9      2372.941516
2        3539.880374
---------------------------------------------
```

1.4 Convergence Rates for Euler Methods

Function for the exact solution:

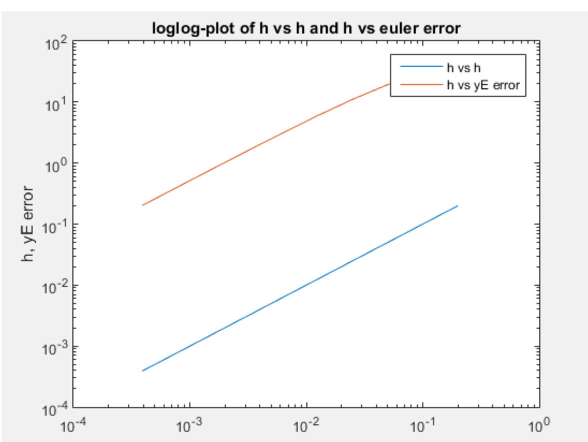
```
function f = yexact(t)
f = 1/4*t - 3/16 + 19/16*exp(4*t);
end
```

Computing the error:

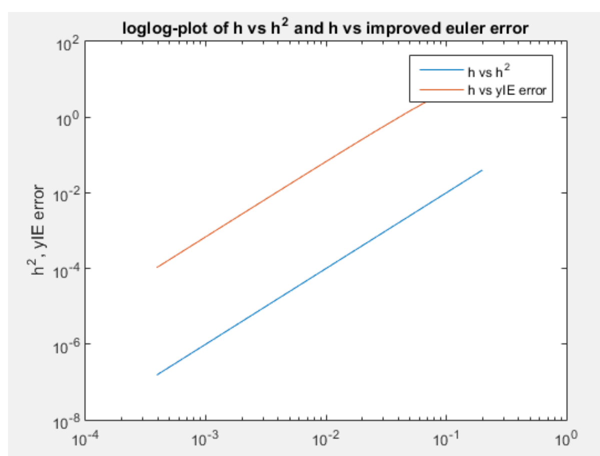
```
    yEh(m) = y(end-1);
    yEerror(m) = abs(yexact(1) - yEh(m));
    setGlobalx(yEerror);
```

The setGlobalx function allows values for variables to be used over different functions.

Plot 1



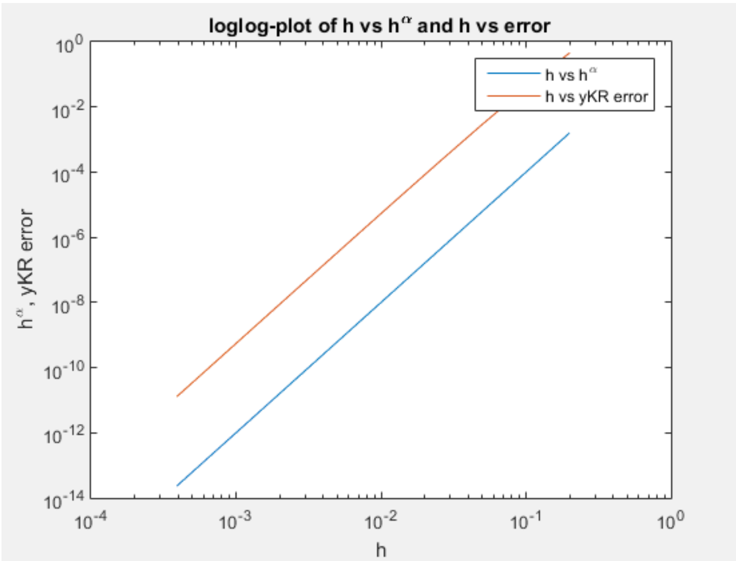
Plot 2



Since the Euler method error curve is similar to the h^1 curve and the improved Euler method error curve is similar to the h^2 curve, the improved method converges faster to the exact solution than the normal Euler method does.

1.5 Convergence Rates for Runge-Kutta Method

Plot 3



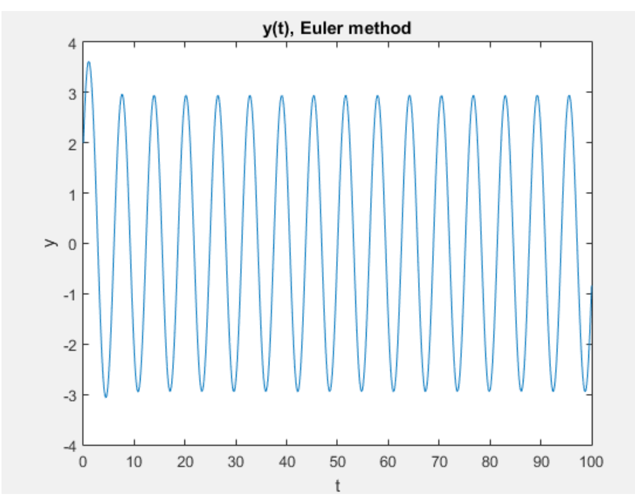
This graph has α = 4. This was determined to be the best alpha value by calculating the slope of the yRK error curve.

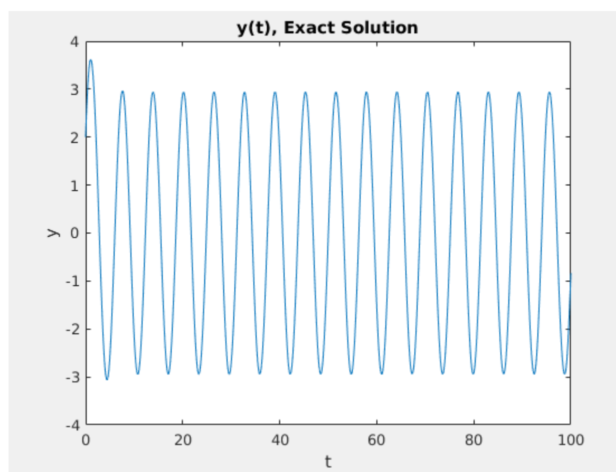
```
loglog(N, yKRerror)
CE = polyfit(log(N), log(yKRerror), 1);
slope = CE(1)
```

```
slope =

    3.9186
```

The closest integer to 3.9186 is 4. The RK4 method converges to the exact solution even faster than the previous two methods because of its high α value.

2.1 Test Problem

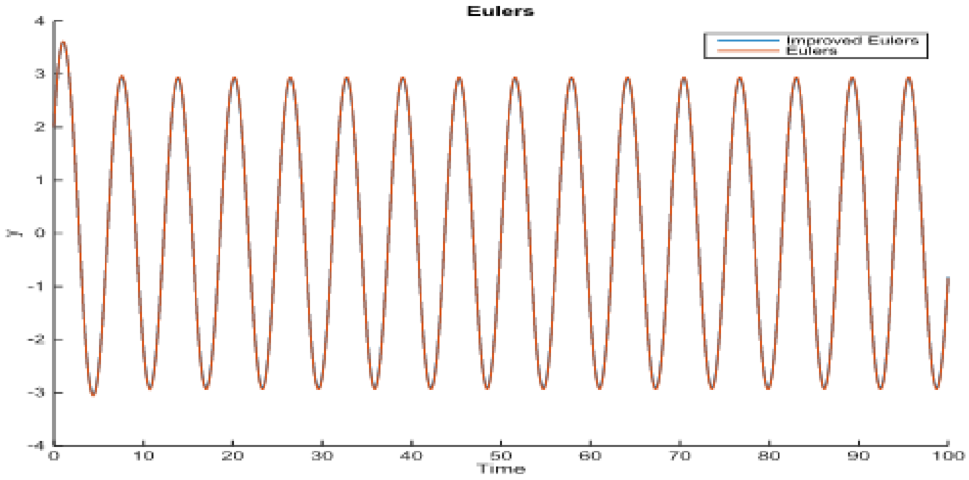




The two results are identical. Euler's method is a valid method to solve similar initial value problems.

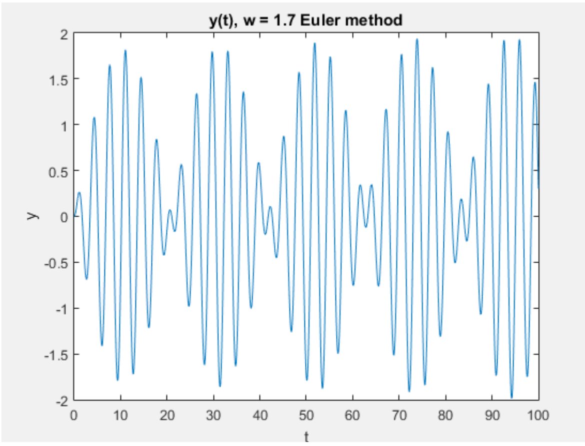
2.2 Improved Euler

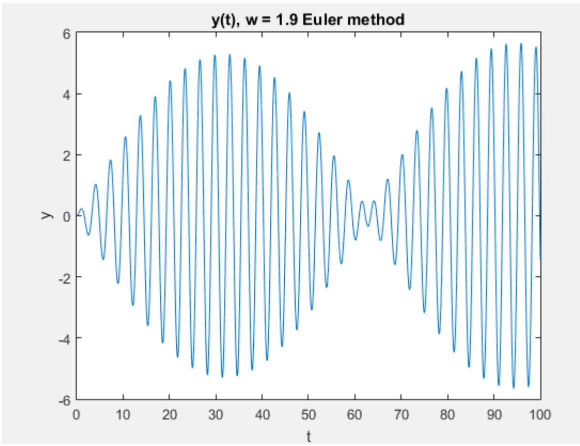
The graph of the improved Euler's method seems identical to the previous two. To show that it does agree with the Euler method, a graph of them plotted on top of each other is produced. With a low enough h value, h = .01, there is no visible difference,

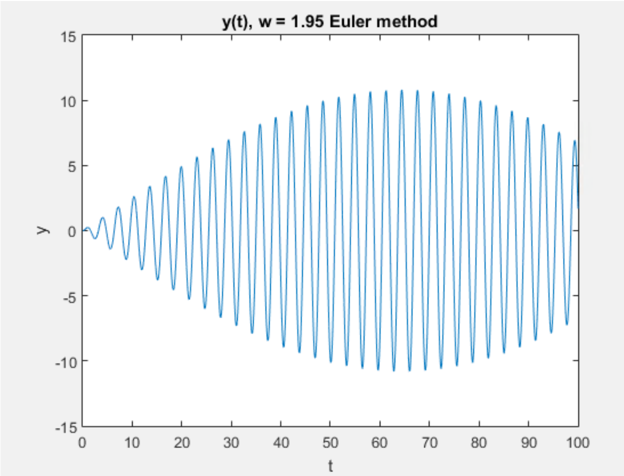


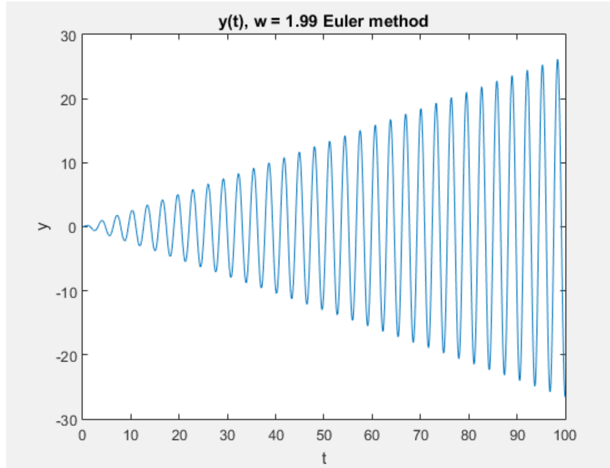
2.3 Beats and Resonance

Here are some graphs of the analytical solution with various w values and h = .001.







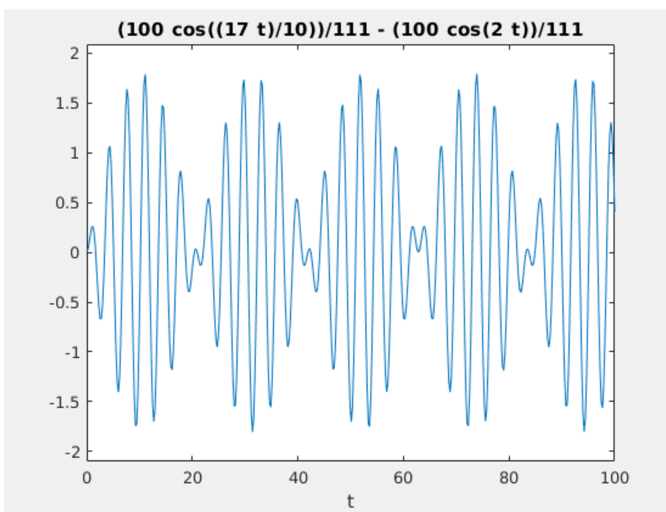


The natural frequency occurs when the amplitude is at its highest. This appears to be w = 2.

In order to determine which h values are good enough to produce a valid solution, the graphs of the analytical solution of Euler's and improved Euler's with varying h values were compared to graphs of a solution found with the function dsolve.

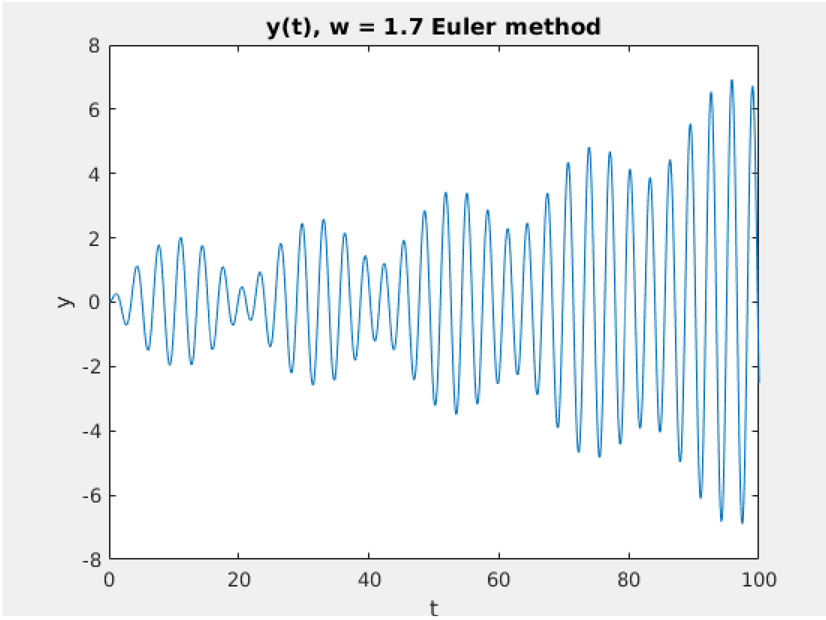
```
sol = dsolve('Dy1=y2','Dy2=-4*y1+cos(1.99*t)','y1(0)=0','y2(0)=0','t');

figure(1)
ezplot(sol.y1, [0, 100])
% un-comment next line if you want the graph of sum of both solutions
%ezplot(sol.y1+sol.y2, [0, 100])
```

Here is the graph of the exact solution when w = 1.7.



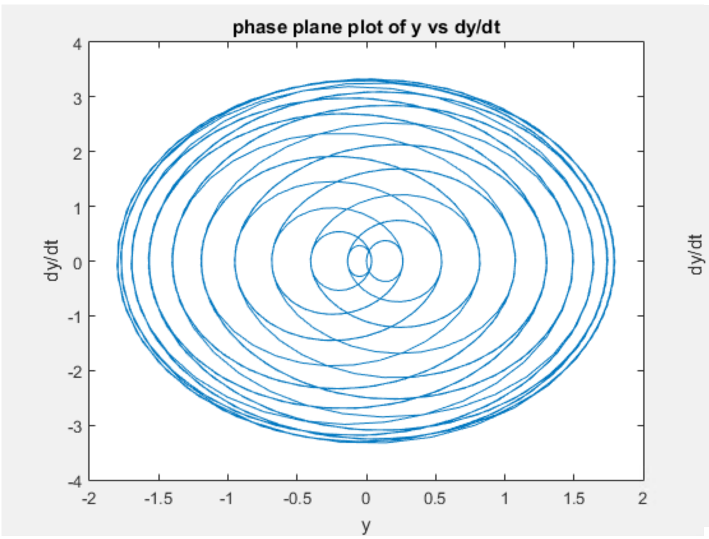
The solutions amplitude differs based on which parts of dsolve solutions are graphed, so the accuracy of the h value is based off of the shape of the graph.

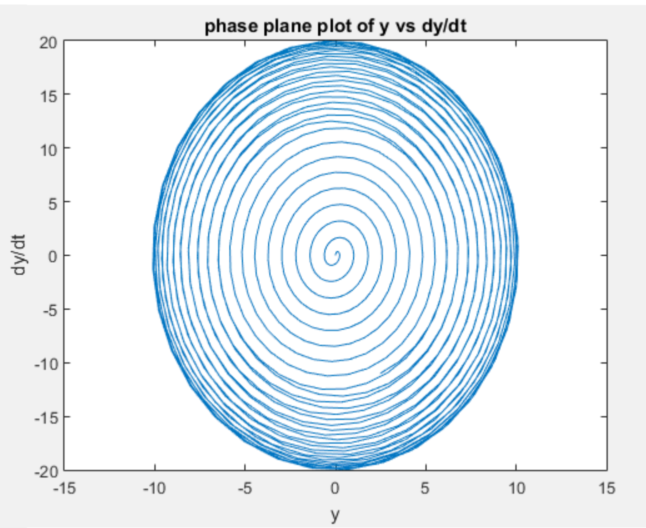
The Euler solution did not agree until h was lowered to .001. For the improved solution, it was best at .01. Here is the graph of the Euler method at h = .01 and w = 1.7 to show the difference.



Here are some phase plots at various w values.

w = 1.7 **w = 1.95**





The phase plots seem to be stable spirals about the center. This means that the real part λ in the general solution is nonzero and is less than zero.