

Project 1
Joseph Martinsen
Math 308-510

STATEMENT OF PROBLEM: Consider the following differential equations:

I. $\frac{dy}{dx} = xy$

II. $\frac{dy}{dx} = y^2$

III. $x \frac{dy}{dx} + \sin(x)y = x^2$

1. Graph the direction field in the neighborhood of the origin
 2. Graph approximate integral curves for 10 different initial conditions (you can use `dfield8.m`)
 3. Solve the differential equation using Euler's method (e.g. with `euler.m`), for three initial conditions and two values of the stepsize, h
 4. If possible compute the error, either using the exact solution (if available) or a very accurate numerical solution (from Matlab)
- I. Using Matlab, we investigate the following differential equation

$$\frac{dy}{dx} = xy$$

- a) **Direction Field.** The direction field near the origin, $-5 \leq x \leq 5$. $-5 \leq y \leq 5$ is shown below

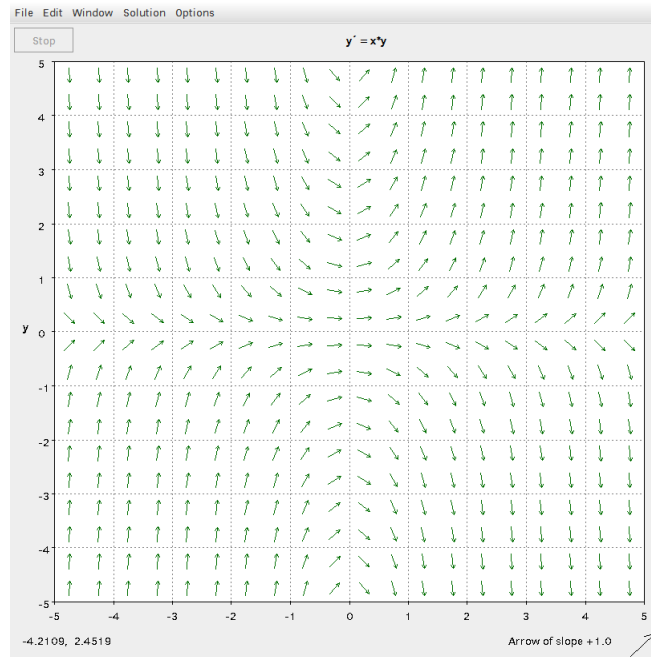


Figure 1: Direction field of $y' = xy$

b) **Integral curves.** 10 different initial conditions are shown in the figure below

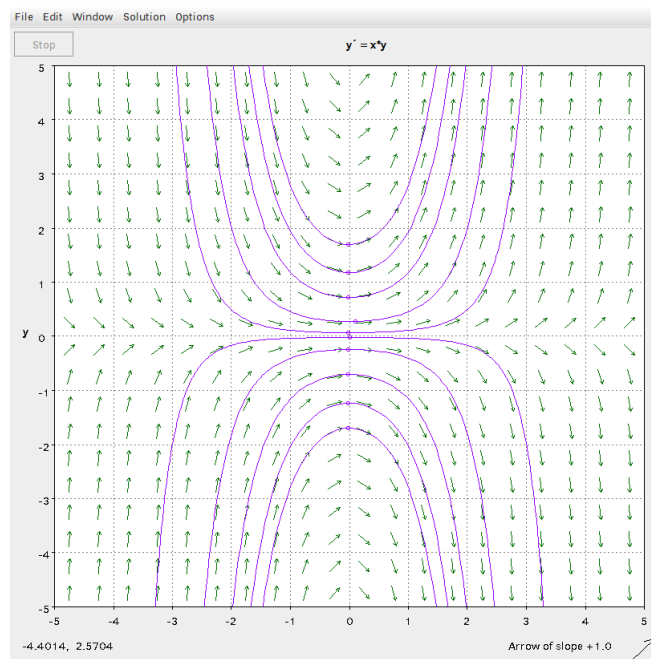


Figure 2: Integral curves of $y' = xy$

c) **Numerical solution.**

The exact solution for $y(0) = 1$ is using Matlab is given by

$$y = e^{x^2/2}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$ is compared to the solution before.

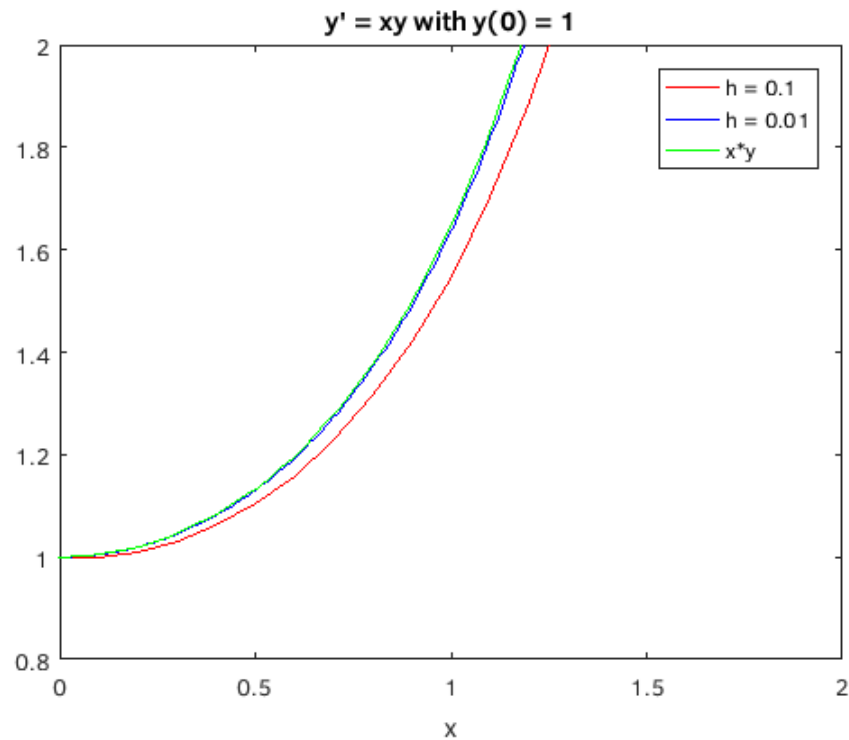


Figure 3: With $y(0) = 1$

The exact solution for $y(0) = 2$ found using Matlab is given by

$$y = 2e^{x^2/2}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$, is compared to the exact solution below

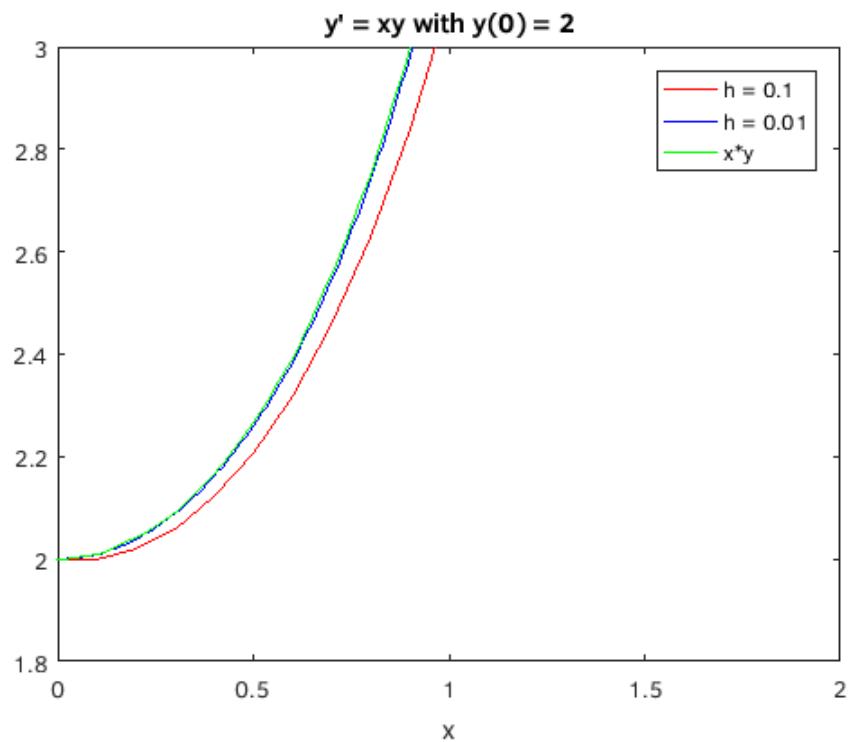


Figure 4: With $y(0) = 2$

The exact solution for $y(0) = -1$ found using Matlab is given by

$$y = -e^{x^2/x}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$, is compared to the exact solution below

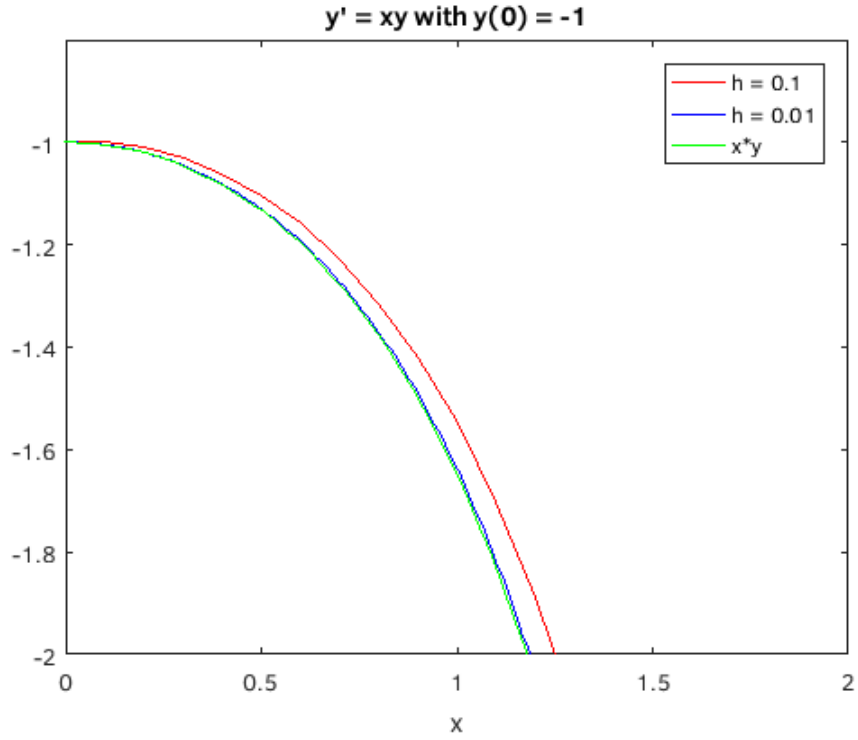


Figure 5: With $y(0) = -1$

d) **Error.** Error, as a function of h . We fix the initial condition, and the interval $[0,1]$, and then compute the error (defined as the maximum of the absolute value of the pointwise error over the interval) versus the step size h . The initial condition being analyzed is for $y(0) = 2$

Table 1: Error Analysis of $y = 2e^{x^2/2}$

Step	Max Abs Difference
.1	0.20322
.05	0.10556
.01	0.00218
.005	0.10946
.001	0.00220

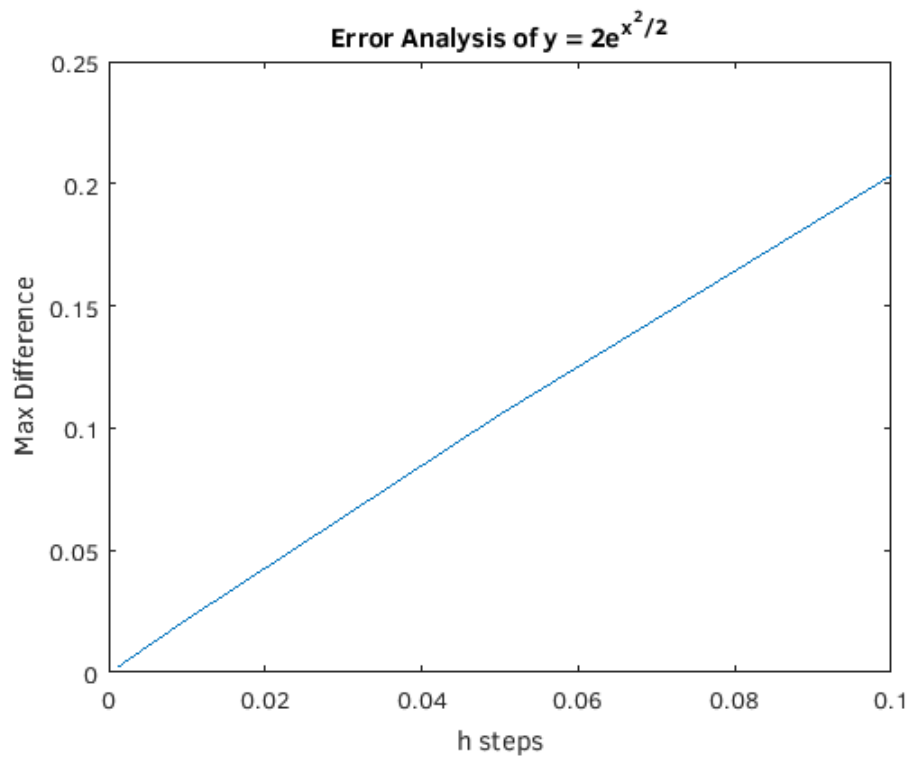


Figure 6: Error Analysis of $y = 2e^{x^2/2}$

II. Using Matlab, we investigate the following differential equation

$$\frac{dy}{dx} = y^2$$

- a) **Direction Field.** The direction field near the origin, $-5 \leq x \leq 5$. $-5 \leq y \leq 5$ is shown below

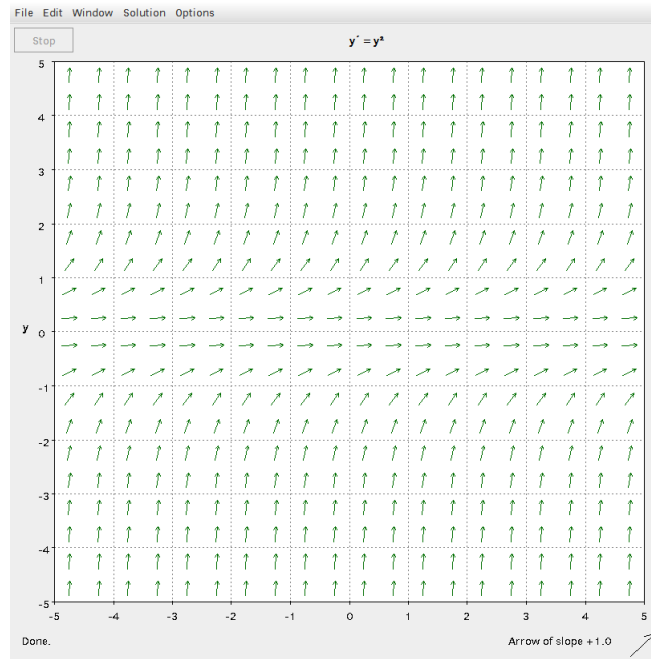


Figure 7: Direction field of $y' = y^2$

b) **Integral curves.** 10 different initial conditions are shown in the figure below

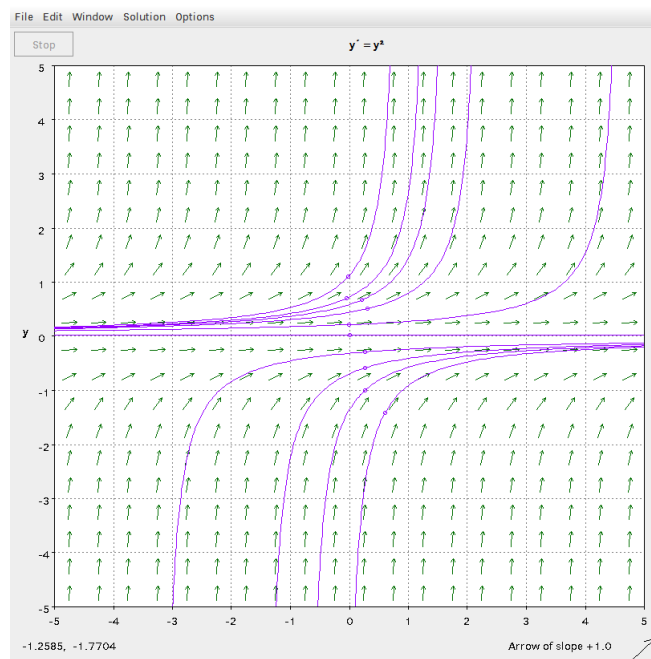


Figure 8: Integral curves of $y' = y^2$

c) **Numerical solution.**

The exact solution for $y(0) = 1$ found using Matlab is given by

$$y = -\frac{1}{x-1}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$, is compared to the exact solution below

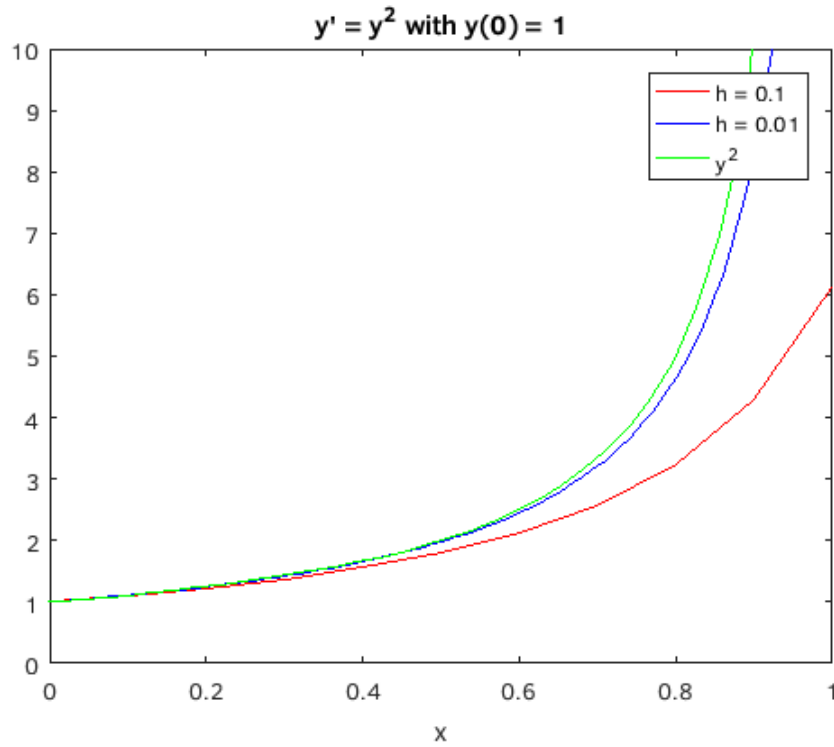


Figure 9: With $y(0) = 1$

The exact solution for $y(0) = 2$ found using Matlab is given by

$$y = -\frac{1}{x - \frac{1}{2}}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$, is compared to the exact solution below

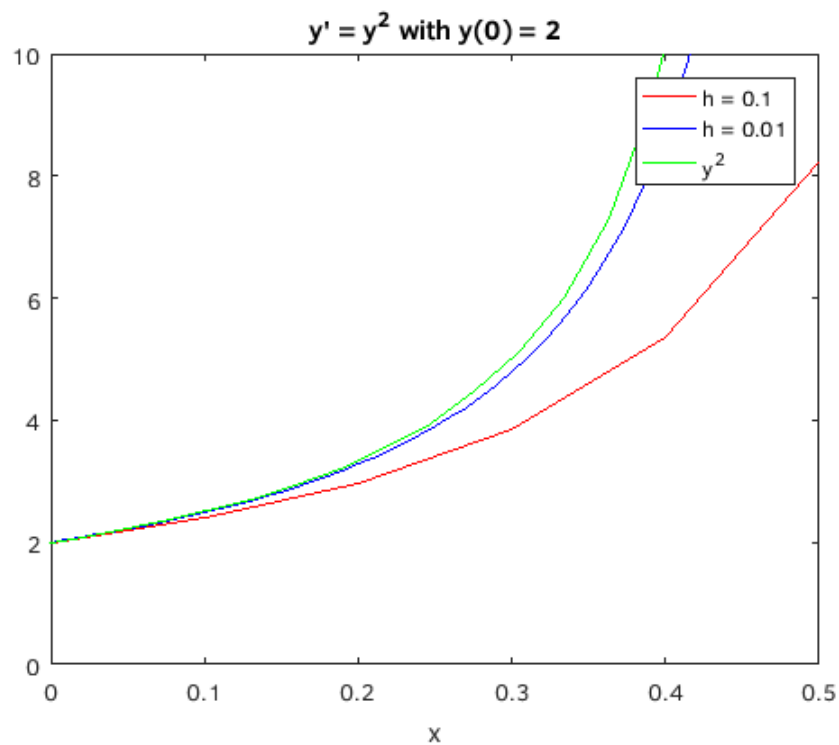


Figure 10: With $y(0) = 2$

The exact solution for $y(0) = -1.5$ found using Matlab is given by

$$y = -\frac{1}{x + \frac{2}{3}}$$

The graph of the results of Euler's method, with $h = 0.1$ and $h = 0.01$, is compared to the exact solution below

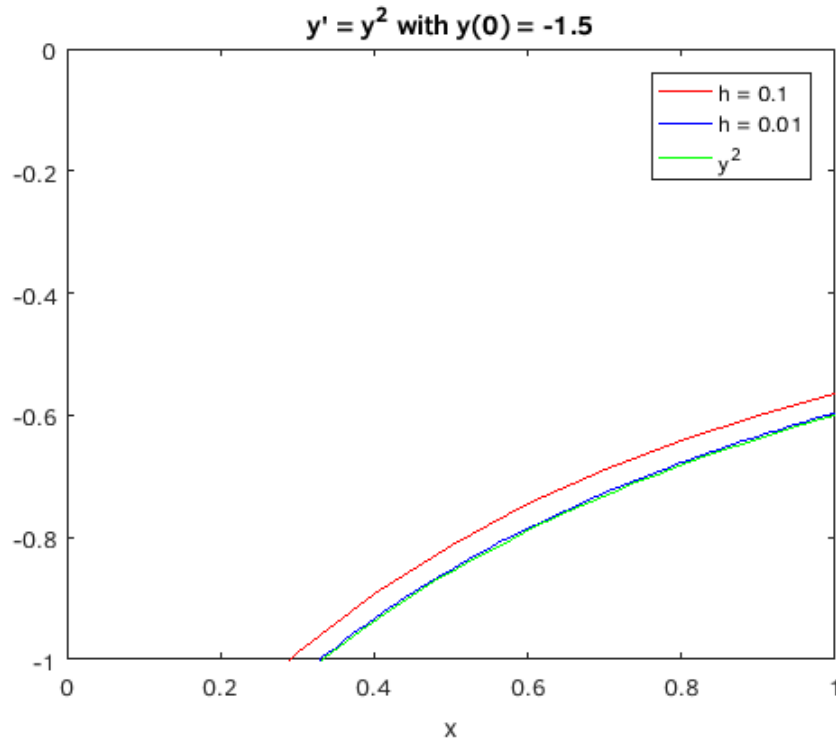


Figure 11: With $y(0) = -1.5$

d) **Error.** Error, as a function of h . We fix the initial condition, and the interval $[0,1]$, and then compute the error (defined as the maximum of the absolute value of the pointwise error over the interval) versus the step size h . The initial condition being analyzed is for $y(0) = -1.5$

Table 2: Error Analysis of $y = -\frac{1}{x + \frac{2}{3}}$

Step	Max Abs Difference
.1	0.04656
.05	0.02187
.01	0.00418
.005	0.00208
.001	0.00041

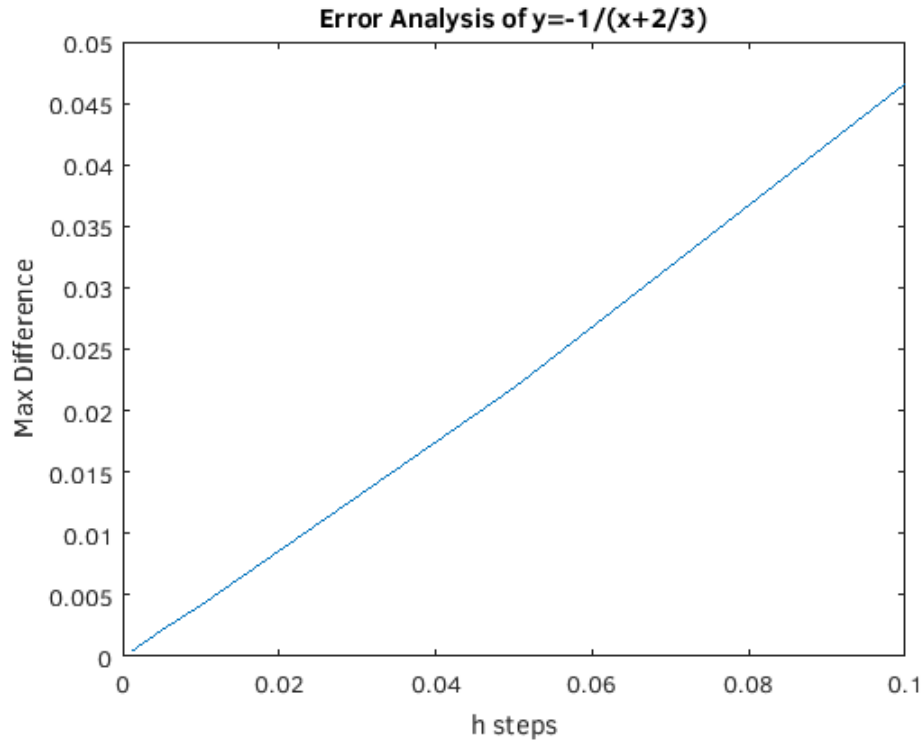


Figure 12: Error analysis with $y(0) = -1.5$

III. Using Matlab, we investigate the following differential equation

$$xy' + \sin(x)y = x^2$$

- a) **Direction Field.** The direction field near the origin, $-5 \leq x \leq 5$. $-5 \leq y \leq 5$ is shown below

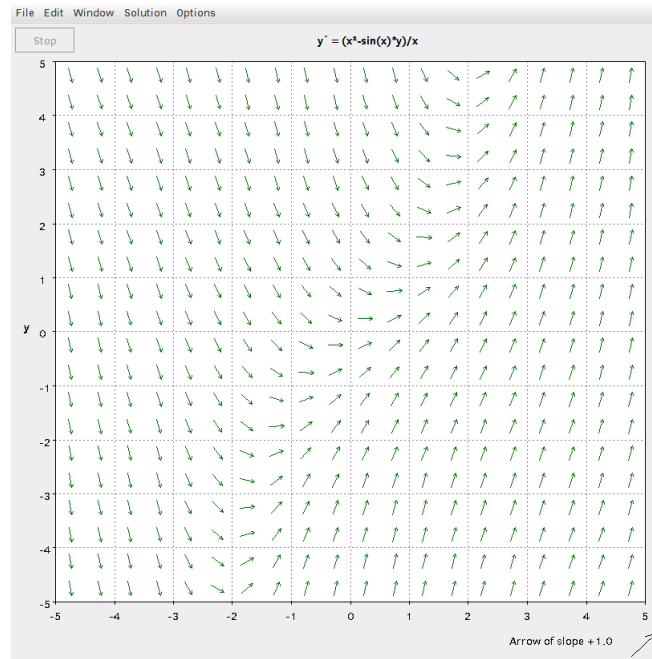


Figure 13: Direction field of $xy' + \sin(x)y = x^2$

b) **Integral curves.** 10 different initial conditions are shown in the figure below

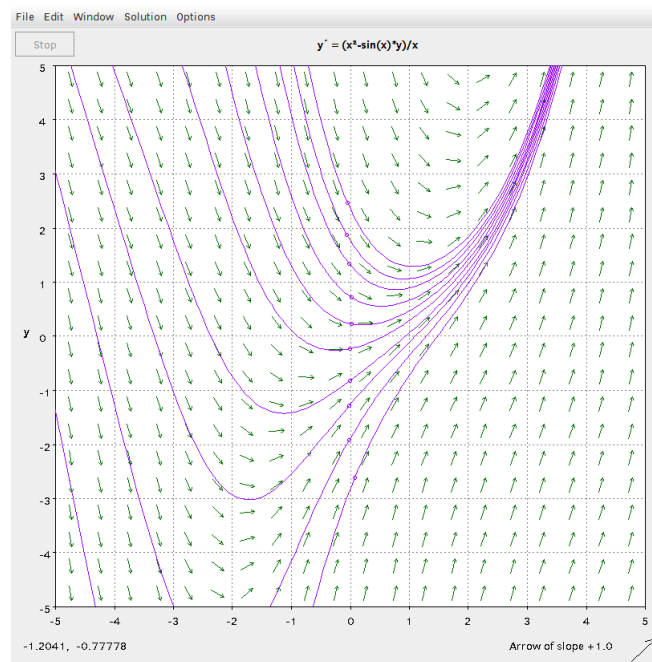


Figure 14: Integral curves of $xy' + \sin(x)y = x^2$

c)

d)

Conclusions

Appendix I: Program Listings

Code Block 1: euler.m

```
1 function [tout, yout] = euler(rhs, t0, y0, h, N)
2
3 % Usage: euler('rhs', t0, y0, h, N)
4 % Note: error will occur if you omit quotes, e.g.
5 % euler(rhs, t0, y0, h, N)
6 %
7 % integrates a single ordinary differential equation
8 % of the form
9 %       $y' = f(t, y)$ 
10 % where
11 %      the initial value of  $t = t0$ 
12 %      and  $y(t0) = y0$ 
13 %      the stepsize =  $h$ 
14 %      and the number of steps =  $N$ 
15 % Note: the right hand side must be defined in the file rhs.m
16
17 % allocate space and initialize to vectors to zero
18 tout = zeros(N+1,1);
19 yout = zeros(N+1,1);
20
21 % set initial values
22 tout(1) = t0;
23 yout(1) = y0;
24
25 % apply Euler's method
26 for i=1:N
```

```

27     tout(i+1) = tout(i) + h;
28     slope = feval(rhs,tout(i),yout(i));
29     yout(i+1) = yout(i) + slope*h;
30 end

```

Code Block 2: rhs.m

```

1 function yprime = rhs(x, y)
2
3 % rhs returns a function which is the right hand side
4 % of a first order differential equation
5 %  $dy/dx = y' = f(x, y)$ 
6
7 yprime = x.*y;

```

Code Block 3: rhs2.m

```

1 function yprime = rhs(x, y)
2
3 % rhs returns a function which is the right hand side
4 % of a first order differential equation
5 %  $dy/dx = y' = f(x, y)$ 
6
7 yprime = y.^2;

```

Code Block 4: main.m

```

1 %% PROJECT 01
2 % written by Joseph Martinsen
3 % Math 308–510
4 clear; clc;
5
6 %% Problem 1
7 clear; clc;

```

```

8 syms y(x)
9 eqn = diff(y,x) == x*y;
10 cond = y(0) == 1;
11 cond1 = y(0) == 2;
12 cond2 = y(0) == -1;
13 ySol1 = char(dsolve(eqn,cond));
14 ySol2 = char(dsolve(eqn,cond1));
15 ySol3 = char(dsolve(eqn,cond2));
16
17 fprintf('Problem 1: dy/dx = xy\n\n')
18 fprintf('Y solution for y(0) = 1: %s\n',ySol1)
19 fprintf('Y solution for y(0) = 2: %s\n',ySol2)
20 fprintf('Y solution for y(0) = -1: %s\n',ySol3)
21
22 % Plot for y(0) = 1
23 [x1, y1] = euler('rhs',0,1,0.1,100);
24 [x2, y2] = euler('rhs',0,1,0.01,200);
25
26 plot(x1,y1,'r')
27 hold on
28 plot(x2,y2,'b')
29 p = ezplot(ySol1);
30
31 set(p,'Color','g')
32 title('y' == xy with y(0) = 1')
33 axis([0 2 0.8 2])
34 legend('h = 0.1', 'h = 0.01', 'x*y')
35
36 % Plot for y(0) = 1
37 figure
38 [x3, y3] = euler('rhs',0,2,0.1,100);

```

```

39 [x4, y4] = euler('rhs',0,2,0.01,200);
40
41 plot(x3,y3,'r')
42 hold on
43 plot(x4,y4,'b')
44 p = ezplot(ySol2);
45
46 set(p,'Color','g')
47 title('y'' vs xy with y(0) = 2')
48 axis([0 2 1.8 3])
49 legend('h = 0.1','h = 0.01','x*y')
50
51
52 % Plot for y(0) = -1
53 figure
54 [x3, y3] = euler('rhs',0,-1,0.1,100);
55 [x4, y4] = euler('rhs',0,-1,0.01,200);
56
57 plot(x3,y3,'r')
58 hold on
59 plot(x4,y4,'b')
60 p = ezplot(ySol3);
61
62 set(p,'Color','g')
63 title('y'' vs xy with y(0) = -1')
64 axis([0 2 -2 -.8])
65 legend('h = 0.1','h = 0.01','x*y')
66
67 % Error Analysis fixing y(0) = 2
68 [x5, y5] = euler('rhs',0,2,0.1,10);
69 [x6, y6] = euler('rhs',0,2,0.05,20);

```



```

70 [x7, y7] = euler('rhs',0,2,0.01,100);
71 [x8, y8] = euler('rhs',0,2,0.005,200);
72 [x9, y9] = euler('rhs',0,2,0.001,1000);
73
74 eq5 = 2 .* exp(x5.^2/2);
75 eq6 = 2 .* exp(x6.^2/2);
76 eq7 = 2 .* exp(x7.^2/2);
77 eq8 = 2 .* exp(x8.^2/2);
78 eq9 = 2 .* exp(x9.^2/2);
79
80 maxdiff(1) = max(abs(y5-eq5));
81 maxdiff(2) = max(abs(y6-eq6));
82 maxdiff(3) = max(abs(y7-eq7));
83 maxdiff(4) = max(abs(y8-eq8));
84 maxdiff(5) = max(abs(y9-eq9));
85
86 fprintf('\nStep\tDifference\n')
87 fprintf('.1\t%d\n', maxdiff(1))
88 fprintf('.05\t%d\n', maxdiff(2))
89 fprintf('.01\t%d\n', maxdiff(3))
90 fprintf('.005\t%d\n', maxdiff(4))
91 fprintf('.001\t%d\n', maxdiff(5))
92
93 step = [.1 .05 .01 .005 .001];
94
95 figure
96 plot(step, maxdiff)
97 title('Error Analysis of  $y = 2e^{x^2/2}$ ')
98 xlabel('h_steps')
99 ylabel('Max_Difference')
100

```

```

101
102 %% Problem 2
103 syms y(x)
104 eqn = diff(y,x) == y^2;
105 cond = y(0) == 1;
106 cond1 = y(0) == 2;
107 cond2 = y(0) == -1.5;
108 ySol1 = dsolve(eqn,cond);
109 ySol2 = dsolve(eqn,cond1);
110 ySol3 = dsolve(eqn,cond2);
111
112 fprintf( '\n\nProblem 2: dy/dx = y^2\n\n' )
113 fprintf( 'Y solution for y(0) = 1: %s\n', ySol1 )
114 fprintf( 'Y solution for y(0) = 2: %s\n', ySol2 )
115 fprintf( 'Y solution for y(0) = -1.5: %s\n', ySol3 )
116
117
118 % Plot for y(0) = 1
119 [x1, y1] = euler( 'rhs2', 0, 1, 0.1, 200);
120 [x2, y2] = euler( 'rhs2', 0, 1, 0.01, 200);
121
122 figure
123 plot(x1,y1, 'r')
124 hold on
125 plot(x2,y2, 'b')
126 p = ezplot(ySol1);
127
128 set(p, 'Color', 'g')
129 title( 'y' == y^2 with y(0) = 1 )
130 axis([0 1 0 10])
131 legend( 'h = 0.1', 'h = 0.01', 'y^2' )

```

```

132
133 % Plot for y(0) = 2
134 [x1, y1] = euler('rhs2',0,2,0.1,200);
135 [x2, y2] = euler('rhs2',0,2,0.01,200);
136
137 figure
138 plot(x1,y1,'r')
139 hold on
140 plot(x2,y2,'b')
141 p = ezplot(ySol2);
142
143 set(p,'Color','g')
144 title('y'' = y^2 with y(0) = 2')
145 axis([0 .5 0 10])
146 legend('h = 0.1', 'h = 0.01', 'y^2')
147
148 % Plot for y(0) = 1
149 [x1, y1] = euler('rhs2',0,-1.5,0.1,200);
150 [x2, y2] = euler('rhs2',0,-1.5,0.01,200);
151
152 figure
153 plot(x1,y1,'r')
154 hold on
155 plot(x2,y2,'b')
156 p = ezplot(ySol3);
157
158 set(p,'Color','g')
159 title('y'' = y^2 with y(0) = -1.5')
160 axis([0 1 -1 0])
161 legend('h = 0.1', 'h = 0.01', 'y^2')
162

```

```

163 % Error Analysis fixing  $y(0) = 2$ 
164 [x5, y5] = euler('rhs2', 0, -1.5, 0.1, 10);
165 [x6, y6] = euler('rhs2', 0, -1.5, .05, 20);
166 [x7, y7] = euler('rhs2', 0, -1.5, 0.01, 100);
167 [x8, y8] = euler('rhs2', 0, -1.5, 0.005, 200);
168 [x9, y9] = euler('rhs2', 0, -1.5, 0.001, 1000);
169
170 eq5 = -1./(x5+2/3);
171 eq6 = -1./(x6+2/3);
172 eq7 = -1./(x7+2/3);
173 eq8 = -1./(x8+2/3);
174 eq9 = -1./(x9+2/3);
175
176 maxdiff(1) = max(abs(y5-eq5));
177 maxdiff(2) = max(abs(y6-eq6));
178 maxdiff(3) = max(abs(y7-eq7));
179 maxdiff(4) = max(abs(y8-eq8));
180 maxdiff(5) = max(abs(y9-eq9));
181
182 fprintf('\nStep\tDifference\n')
183 fprintf('.1\t%d\n', maxdiff(1))
184 fprintf('.05\t%d\n', maxdiff(2))
185 fprintf('.01\t%d\n', maxdiff(3))
186 fprintf('.005\t%d\n', maxdiff(4))
187 fprintf('.001\t%d\n', maxdiff(5))
188
189 step = [.1 .05 .01 .005 .001];
190
191 figure
192 plot(step, maxdiff)
193 title('Error Analysis of  $y=-1/(x+2/3)$ ')

```

```

194 xlabel( 'h_steps ' )
195 ylabel( 'Max_Difference ' )
196
197 %% Problem 3
198 % syms y(x)
199 % eqn = x*diff(y,x) + sin(x)*y == x^2;
200 % cond = y(0) == 1;
201 % cond1 = y(0) == 2;
202 % cond2 = y(0) == -1;
203 % ySol(x) = dsolve(eqn, cond)
204 % ySol(x) = dsolve(eqn, cond1)
205 % ySol(x) = dsolve(eqn, cond2)

```