Linear Multi-step ODE Methods, Program 5

Colton Piper

December 13th, 2017

1 Executive Summary

This report focuses on some linear one-step and multi-step ODE Methods that we discuss and implement on a code. We will analyze the methods including absolute stability of the methods in context. We will graph the stable regions for some methods and discuss when methods are convergent and when they are not convergent.

2 Statement of the Problem

First we will analyze a family of methods defined by

$$y_n = y_{n-1} + h(\theta f_n + (1-\theta) f_{n-1})$$

where $\theta \in [0, 1]$. We will then identify some commonly known methods of the family determine for what values of θ that the methods are convergent and when they are of order 2. Then lastly we will determine what values of θ are A-Stable and if any produce a method with stiff decay.

Next we will implement code that produces graphs for A-Stable region's regions for Adams methods, both Bashforth and Moulton, and BDF methods. Also produce code for A-stable regions for different values of θ for the family of methods above. Then we will also graph the stability regions for a few other methods from homework 7.

After we will implement code to several different mehtods including

$$y_n = 4y_{n-1} + 5y_{n-2} + h(4f_{n-1} + 2f_{n-2}) \tag{1}$$
$$y_n = y_{n-2} + 2hf_{n-1} \tag{2}$$
$$y_n = y_{n-1} + \frac{h}{2}(3f_{n-1} - f_{n-2}) \tag{3}$$
$$y_n = y_{n-1} + \frac{h}{2}(f_n - f_{n-1}) \tag{4}$$
$$y_n = y_{n-1} + hf_n \tag{5}$$
$$y_n = \frac{4}{3}y_{n-1} - \frac{1}{3}y_{n-2} + \frac{2}{3}hf_n \tag{6}$$

We will use our model problem $f = \lambda(y - F'(t)) + F'(t)$ where $y(0) = y_0$ and $y(t) = (y_0 - F(0))e^{\lambda t} + F(t)$. But we will take $F(t) = \sin(t)$ and $y(0) = 1$ on the unit line for time. For the multi-step methods we will use both exact solutions for $y_0(0)$ and $y_1(h)$ for the step size we choose. We will analyze different step sizes and different $l\lambda$ values to see what works well with different methods.

3 Description of the Mathematics

3.1 Family of Linear One-Step Methods

First we will discuss the family of methods defined by

$$y_n = y_{n-1} + h(\theta f_n + (1 - \theta)f_{n-1})$$

where $\theta \in [0, 1]$. We can get three easily recognizable methods when we set θ to 0, 1, and $\frac{1}{2}$. From those we get the Forward Euler Method $y_n = y_{n-1} + hf_{n-1}$, Backward Euler Method $y_n = y_{n-1} + hf_n$, and Trapezoid $y_n = y_{n-1} + \frac{h}{2}(f_n - f_{n-1})$, respectively.

Next to show convergence we need the methods to be consistent and stable. First from class notes we know that a linear multi-step method with characteristic polynomials $\rho(\eta)$ and $\sigma(\eta)$ is consistent if and only if $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$. In our family of methods our characteristic polynomials are

$$\rho(\eta) = \eta - 1 \quad \text{and} \quad \sigma(\eta) = \theta\eta + 1 - \theta.$$

Thus we can see that indeed $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$. Thus all of the methods in this family are consistent. We know that the polynomials also satisfies the root condition. Then we know that all the methods are strongly stable as well because they strongly stable. Thus they will be convergent for all θ. Now how convergent will these methods be? we have a nice Theorem (22.3) that tells us that a strongly stable k-step method can be at most order $k+1$.

Next we will use the test problem $y' = \lambda y$ with $y(0) = C$. A method is called δ damping if

$$\lim_{\Re(h\lambda)\to-\infty} \frac{|y_n|}{|y_{n-1}|} \leq \delta$$

where $0 \leq \delta < 1$. So we can write that our method in terms of $h\lambda$ using our test problem. So we have

$$\begin{aligned}
y_n &= y_{n-1} + h\theta\lambda y_n + h(1-\theta)\lambda y_{n-1} \\
y_n(1 - \lambda h\theta) &= y_{n-1}(1 + h\lambda(1-\theta)) \\
\frac{|y_n|}{|y_{n-1}|} &= \frac{|1 + h\lambda(1-\theta)|}{|1 - h\theta\lambda|} \\
\lim_{\Re(h\lambda)\to-\infty} \frac{|(h\lambda)^{-1} + (1-\theta)|}{|(h\lambda)^{-1} - \theta|} &\leq \delta \\
\frac{|1-\theta|}{|\theta|} &\leq \delta.
\end{aligned}$$

Now we know that for absolute stability we will need for $\frac{|1-\theta|}{|\theta|} \leq 1$. This tells us that the sequence will not be unbounded as our step size gets smaller and smaller. Thus we can see that only the methods where $\theta \geq 1/2$ will be A-stable.

Lastly stiff decay is when the limit for delta damping is 0. And we can see that that only happens when $\theta = 0$. Thus that is the only method that has stiff decay in this family of methods.

3.2 Graphing Absolute Stability Regions

Here we are going to need the coefficients for the characteristic polynomials of the Adams methods and the BDF methods. To graph the methods we used the formula for the boundary of the absolute stability region for $h\lambda = z \in \mathbb{C}$ is $z = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}$. This is where $\rho(\eta)$ and $\sigma(\eta)$ are the characteristic polynomials of each method defined like

$$\rho(\eta) = \sum_{j=0}^{k} \alpha_j \eta^{k-j} \qquad\qquad \sigma(\eta) = \sum_{j=0}^{k} \beta_j \eta^{k-j}$$

So first for the Adams-Bashforth methods we have that for all k, $\beta_0 = 0$, $\alpha_0 = 1$, and $\alpha_1 = -1$. Other alphas will be zero. Then for higher step methods we have

Adams-Bashforth Methods

k		1	2	3	4	5	6
1	β_j	1					
2	$2\beta_j$	3	-1				
3	$12\beta_j$	23	-16	5			
4	$24\beta_j$	55	-59	37	-9		
5	$720\beta_j$	1901	-2774	2616	-1274	251	
6	$1440\beta_j$	4277	-7923	9982	-7298	2877	-475

Table 1: Coefficients for Adams-Bashforth methods.

Next we have Adams-Moultons methods which have the same alphas as Adams-Bashforth methods but their β_0 is non zero due to them being an implicit family. For higher step methods we have

Adams-Moultons Methods

k		0	1	2	3	4	5
1	β_j	1	0				
1	$2\beta_j$	1	1				
2	$12\beta_j$	5	8	-1			
3	$24\beta_j$	9	19	-5	1		
4	$720\beta_j$	251	646	-264	106	-19	
5	$1440\beta_j$	475	1427	-798	482	-173	27

Table 2: Coefficients for Adams-Moulton methods.

Lastly we have the BDF methods and the coefficients for those are first for all k $\alpha_0 = 1$. For higher step methods we have

BDF Methods

k	β_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1	-1					
2	$\frac{2}{3}$	$-\frac{4}{3}$	$\frac{1}{3}$				
3	$\frac{6}{11}$	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$			
4	$\frac{12}{25}$	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$		
5	$\frac{60}{137}$	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$	
6	$\frac{60}{147}$	$-\frac{360}{147}$	$\frac{450}{137}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$

Table 3: Coefficients for BDF methods.

Then using characteristic polynomials we can determine the boundaries for absolute stability using the characteristic polynomials.

4 Description of the Algorithms and Implementation

4.1 Graphing Absolute Stability Regions

The algorithms and implementation of those algorithms were not to difficult in this program. First we will discuss the routines used to graph the absolute stability. Now as we said above in 'Description of Mathematics' the boundary for the absolute region is for $h\lambda = z \in \mathbb{C}$ is $z = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}$. Thus each routing is just using for loops to produce many points that will represent the boundary.

First we can notice that in each family of methods one of the two characteristic polynomials ρ or σ is very simple due to only one or two of the coefficients representing that characteristic polynomial being zero. Seeing this each routine for this only needs a for loop for the more complex characteristic polynomial in each method. The Adams methods have a simple ρ polynomial only being $e^{i*k*\theta} - e^{i*(k-1)*\theta}$. While the BDF has a simple σ polynomial being $\beta_0 e^{i*k*\theta}$. Then the family of linear one-step methods we need to graph for different values of θ is just a one line function because the characteristic polynomials are very simple.

Now for each multi-step Adams method you need to pass in how many steps the method is and the array filled with the β_i and then the amount of points you want graphed, i.e a vector filled with mesh from $[0, 2\pi]$. Then for the BDF methods we pass in the amount of steps for the method, the β_0, a vector filled with α_i, and the mesh for points graphed. Lastly for the linear one-step method you only need to pass in the θ for the method within the family and the mesh for the points to be graphed.

4.2 Multi-Step Methods Implemented

All of our methods for this were also fairly simple to implement. Even for the implicit methods we do to the simplicity of the function f we could still get an explicit function for an approximate y_n from an implicit method. This was very nice because it meant we did not have to use any root finding methods to determine what y_n is. Thus in each method we could just implement a for loop which determines all of the vector of y_i for as many steps we want to take. Then if they are multi-step we just use the exact to get the amount of steps needed to start the method in context.

For each method you pass in the amount of steps going to be taken which determines the step-size. Then you pass in an empty array of size N that you will fill up with each y_n determined from each iteration of the method. Then lastly you need to pass in the starting time and the time you want to end at.

5 Description of the Experimental Design and Results

5.1 Graphing Absolute Stability Regions

From the code implemented which we talked about in the last section we can produce many graphs for the absolute stability regions. First we have our family of one-step methods determined by θ. We get this graph

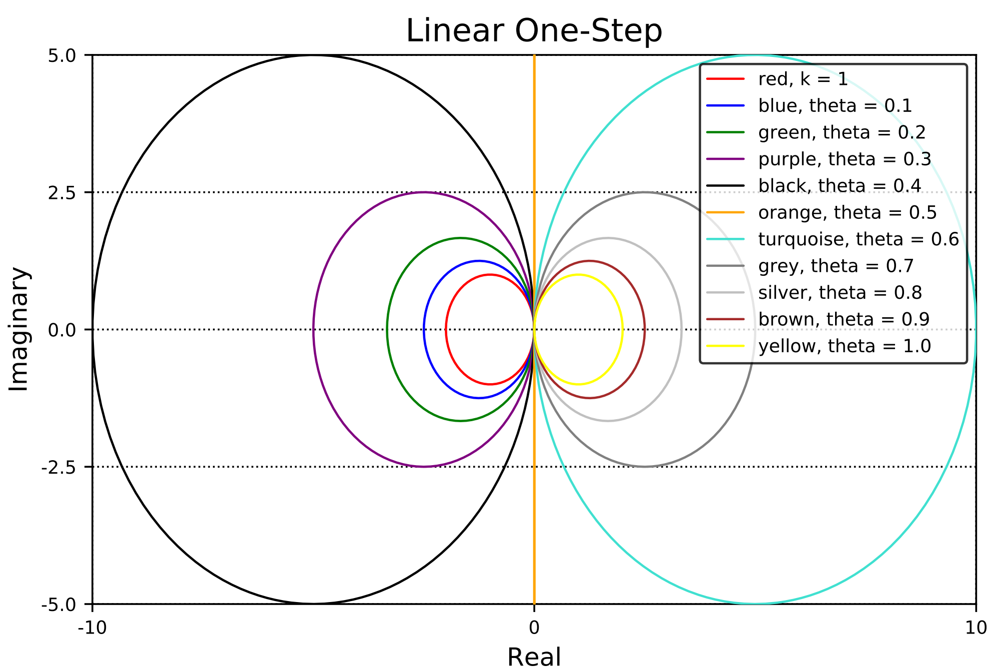

Figure 1: Absolute stability regions for θ from $0 \to 1$ by 0.1.

Next we have the graph for the Adams-Bashforth methods

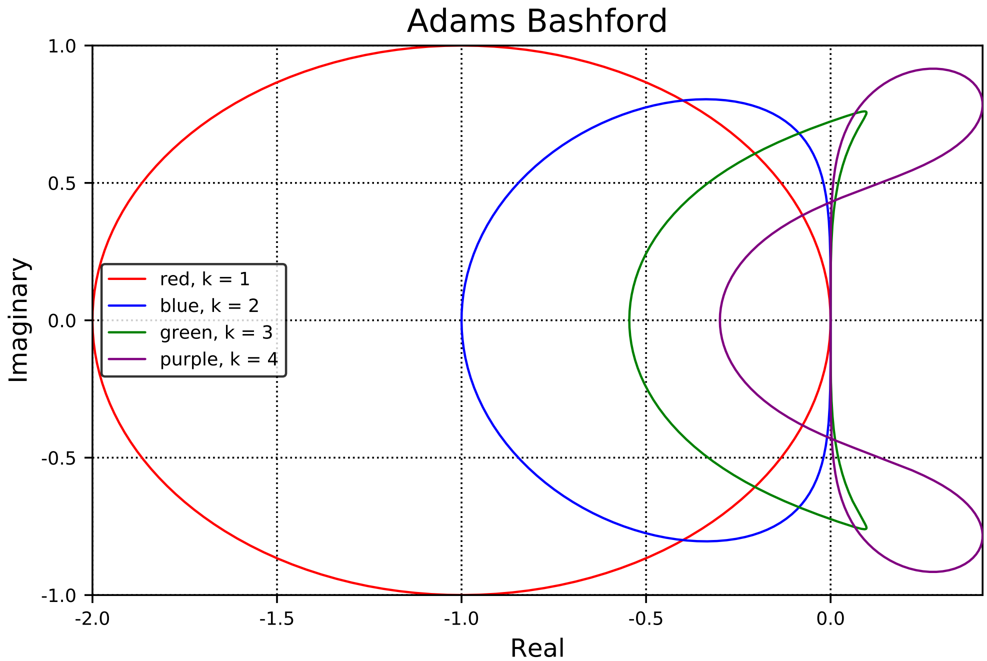

Figure 2: Absolute stability regions for Adams-Bashforth methods up to a 4 step method.

Now we have the graph for the other Adams Methods which are the Adams-Moulton methods

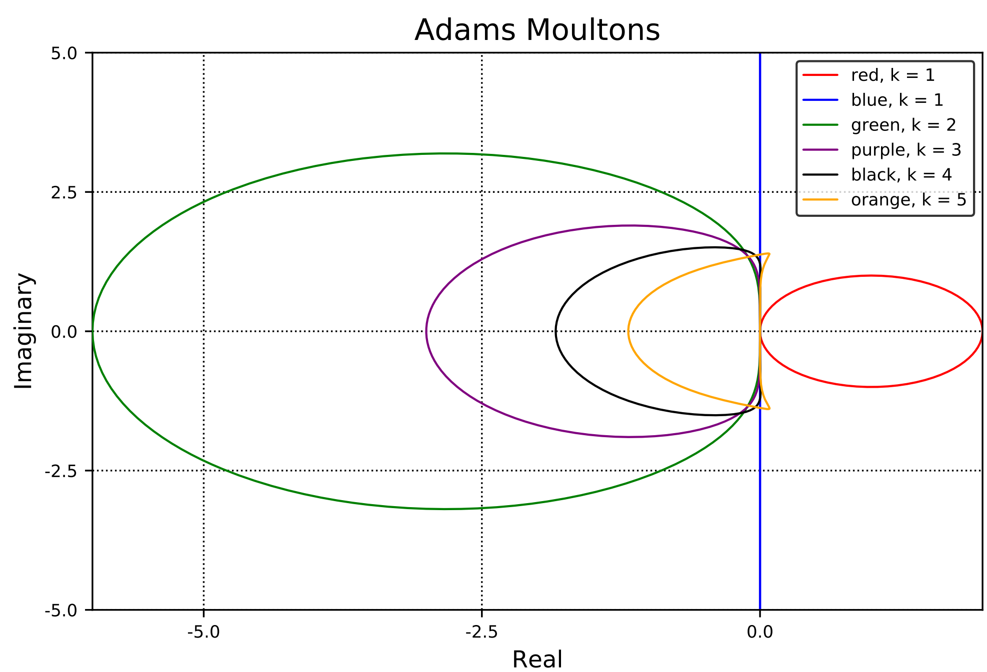

Figure 3: Absolute stability regions for Adams-Moulton methods up to a 5 step method.

Lastly we have the BDF methods

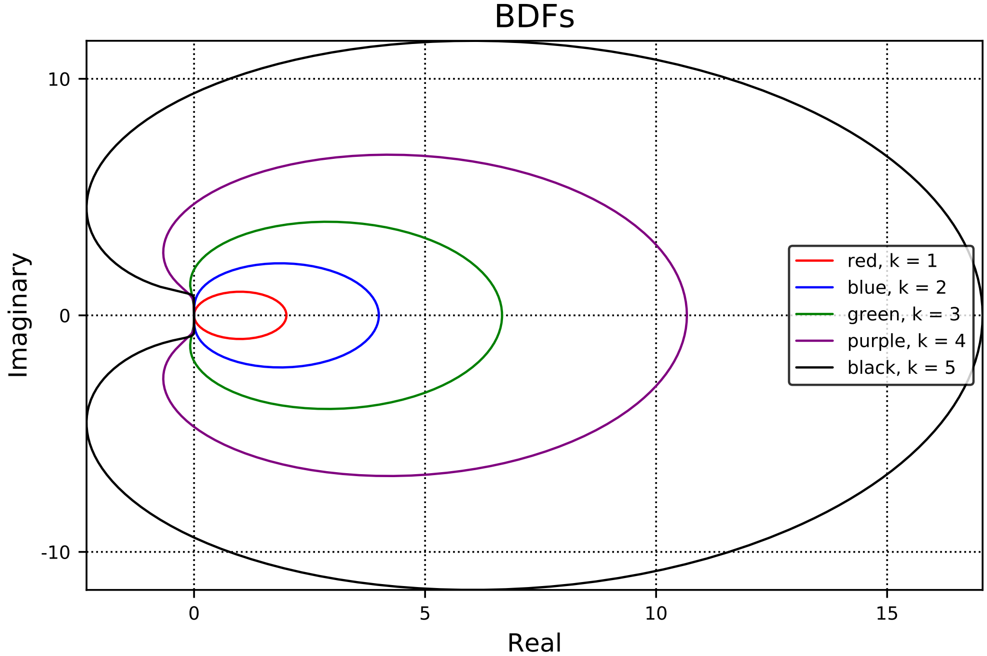

Figure 4: Absolute stability regions for BDF methods up to a 4 step method.

5.2 Multi-Step Methods

Now for the few methods we implemented to actually solve our program we get some of the methods work well for different λ values while others blow up and do not work very well for certain λ. So for certain methods it really depends on how First looking at the $\lambda = 10$ we have the last few steps for if the step size is 0.01 and then 0.001.

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
95	13360.5	4.19787e+57	13152.8	12880.5	142.378	22230.9	14973.6
96	14765.6	-1.97517e+58	14533.6	14229.5	150.172	24700.9	16553.5
97	16318.4	9.2935e+58	16059.4	15719.8	158.396	27445.4	18300.2
98	18034.6	-4.37275e+59	17745.4	17366.2	167.074	30494.8	20231.2
99	19931.2	2.05745e+60	19608.3	19185.1	176.231	33883	22365.9
100	22027.3	-9.68066e+60	21666.9	21194.5	185.893	37647.7	24725.8

Table 4: Last few steps for methods with $h = 0.01$ and $\lambda = 10$.

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
995	20953.1	-nan	20949.6	20944.4	202.707	22029.1	22826.7
996	21163.6	-nan	21160.1	21154.9	203.814	22251.6	23056.1
997	21376.3	-nan	21372.8	21367.5	204.928	22476.3	23287.8
998	21591.2	-nan	21587.6	21582.2	206.048	22703.3	23521.9
999	21808.1	-nan	21804.5	21799.1	207.174	22932.7	23758.3
1000	22027.3	-nan	22023.6	22018.2	208.306	23164.3	23997

Table 5: Last few steps for methods with $h = 0.001$ and $\lambda = 10$.

We can see that Method 1 fails hard in both sizes. We can hypothesize that the method probably not A-stable with the chosen λ and stepsize. The trapezoidal rule also does not fair well in this case as well. Now we can see that Method 5 was very bad with the smaller step but seems to converge with the higher order step size. But with both stepsizes. Methods 2 and 3 are okay for the smaller step size and are accurate up to three digits for a step size of 0.001. Lastly method 6 was not great with a step size of 0.01, but got better with a larger step size.

Next let us look at if $\lambda = -10$ with the same two step sizes. So we have

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
95	0.81349	2.78611e+62	-0.169901	0.813496	1.63622	0.81316	-0.00906594
96	0.819259	-1.47799e+63	1.9059	0.819265	1.65006	0.818922	-0.0104759
97	0.824947	7.84047e+63	-0.375772	0.824952	1.66378	0.824603	-0.0118841
98	0.830553	-4.15924e+64	2.15734	0.830558	1.67735	0.830203	-0.0132907
99	0.836076	2.20641e+65	-0.629999	0.836081	1.6908	0.83572	-0.0146954
100	0.841516	-1.17046e+66	2.46152	0.841521	1.7041	0.841155	-0.0160982

Table 6: Last few steps for methods with $h = 0.01$ and $\lambda = -10$.

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
995	0.838807	-nan	0.837077	0.838807	1.68988	0.83877	-0.0153961
996	0.83935	-nan	0.841098	0.83935	1.69121	0.839314	-0.0155364
997	0.839893	-nan	0.838128	0.839893	1.69254	0.839857	-0.0156766
998	0.840435	-nan	0.842218	0.840435	1.69387	0.840398	-0.0158169
999	0.840976	-nan	0.839176	0.840976	1.69519	0.84094	-0.0159571
1000	0.841516	-nan	0.843335	0.841516	1.69652	0.84148	-0.0160973

Table 7: Last few steps for methods with $h = 0.001$ and $\lambda = -10$.

We can see that with a change in lambda by sign it greatly changes the exact solution and changes how well the methods approximate it. Method 1 still is terrible here and blows up in both step size cases. Method 2 does not do great with the smaller step size, but does do well for the larger step size. Method 3 in this case approximates very well even with the small stepsize. The trapezoidal rule seems to be close to double the actual solution. Then Method 5 does very well with both step sizes. Method 6 does not approximate well with either step size.

Now we will change $\lambda = -500$ and see how the methods fair. So we have

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
95	0.813416	5.259e+128	-2.307e+93	-1.480e+78	-8.567e+16	0.81341	-0.00045788
96	0.819192	-1.251e+130	2.330e+94	1.015e+79	1.291e+17	0.81918	-0.0004858
97	0.824886	2.976e+131	-2.353e+95	-6.973e+79	-1.947e+17	0.82488	-0.00051361
98	0.830497	-7.080e+132	2.376e+96	4.786e+80	2.935e+17	0.83049	-0.0005414
99	0.836026	1.684e+134	-2.400e+97	-3.285e+81	-4.425e+17	0.83602	-0.00056913
100	0.841471	-4.007e+135	2.423e+98	2.255e+82	6.670e+17	0.84146	-0.00059681

Table 8: Last few steps for methods with $h = 0.01$ and $\lambda = -500$.

Step	Exact	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
995	0.838759	-nan	-4.508e+205	0.838759	1.67923	0.838758	-0.00058298
996	0.839303	-nan	7.295e+205	0.839303	1.68032	0.839302	-0.00058574
997	0.839846	-nan	-1.180e+206	0.839846	1.68141	0.839845	-0.0005885
998	0.840389	-nan	1.909e+206	0.840389	1.6825	0.840388	-0.0005913
999	0.84093	-nan	-3.090e+206	0.84093	1.68359	0.840929	-0.0005940
1000	0.841471	-nan	5.000e+206	0.841471	1.68468	0.84147	-0.0005968

Table 9: Last few steps for methods with $h = 0.001$ and $\lambda = -500$.

So we can see that for a small step size only method 5, which is backward Euler fairs well. When we increase the make the step size smaller Method 3 and the trapezoidal method get closer. Method 3 looks like it converges on the exact solution and the trapezoidal looks to converge to about double the exact solution. So we can see that changes in the step size and our parameter can make methods not converge at all. This tells us that ODE methods depend largely on the problem in context and the step size.

6 Conclusions

In this report we analyzed different linear methods to approximate solutions to ODEs. First we focused on a family of linear one-step methods and analyzed which methods in the family were convergent and different types of stabilities for each method and if any were stiff decay. Next we created code to graph the absolute stability regions for different families of methods including the family of linear one-step methods we analyzed. Then lastly we coded some methods and looked at how they performed for different problems. The biggest conclusion that we talked about is how convergence of methods change based on step size and the problem being discussed.