Math 371 FINAL PROJECT
Part 2

Wyatt Kowall [wskowall]
Stefano DeBellis [stefdeb]
Bhairav Mehta [bhairavm]
Nirmal Patel [nnvpatel]

April 15, 2016

1 Part 2: Curvature Flow

This section is for Part 2 of the project. Here, we simulate the evolution of a curve parameterized by α, and described as a partial differential equation with respect to time. The Euler method is used to simulate the evolution of the curve as time progresses. Additionally, the computational cost of the algorithm per time step is discussed. Also, the length of the curve is calculated using the trapezoidal rule and the order of accuracy of the algorithm is analyzed. The evolution of the curve is then simulated with a larger of number of parameterization points and the results are presented.

1.1 Part 2 - Question 1

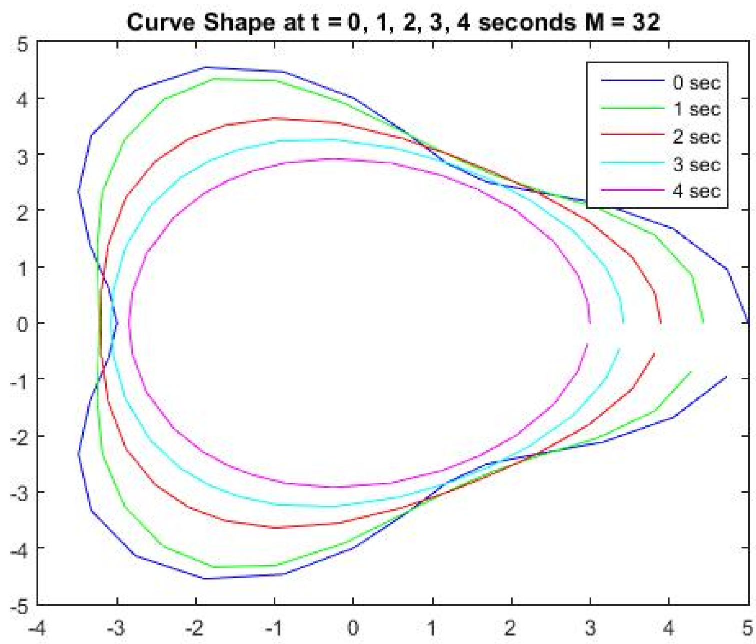

Figure 1: This is M= 32

Figure 1 shows the evolution of the curve, which is plotted at times of 0, 1, 2, 3, and 4 seconds. The positions of points on the curve are calculated using the Euler method with the provided differential equation.

1.2 Part 2 - Question 2

For each time step, the forward Euler method is ran twice, once for x1 and once for x2. Additionally, fft_diff is called four times. One each for the first and second partial derivatives of x1 with respect to α, and one each for the first and second partial derivatives of x2 with respect to α. fft_diff runs fft and ifft once each. Forward euler is O(M), and fft and ifft are O(MlogM). This means that for each time step, there are two O(M) algorithms for computing the forward euler method, and sixteen total O(MlogM) algorithms (two equations that call fft_diff four times, which contains fft and ifft). Additionally, there is basic arithmetic of the elements of the matrices in computing the differential equation values which is O(M).

1.3 Part 2 - Question 3

The Length of the curve at 4 seconds is calculated using the trapezoidal method with varying numbers of time steps (N). Table 1 shows the results of the calculations. Using the length with 6400 time steps as a reference value, the error for the length at time steps of 400, 800, and 1600 is calculated and can be seen in Table 2. When N is doubled the error is halved, therefore it can be deduced that this algorithm is first-order accurate.

Table 1: Length of curves at different time steps (N)

N	Length
400	251.9
800	266.5
1600	274.8
2400	281.7

Table 2: Error of lengths at N time steps, using the length at 6400 steps as a reference value

N	Error
400	29.8
800	15.2
1600	6.9

1.4 Part 2 - Question 4

When the simulation in Step 1 is rerun with M = 64, the output becomes extremely chaotic. When M is increased to 64, we double the amount of spatial discretization points from the previous simulation. Yet, by keeping the number of timesteps (N) the same, we dramatically increase the ratio of $\Delta t/\Delta x$. By increasing this ratio, the equations no longer get sampled enough, a problem that was not present when M was equal to 32. Therefore, we see a chaotic evolution graph as a result of undersampling of the new curve.

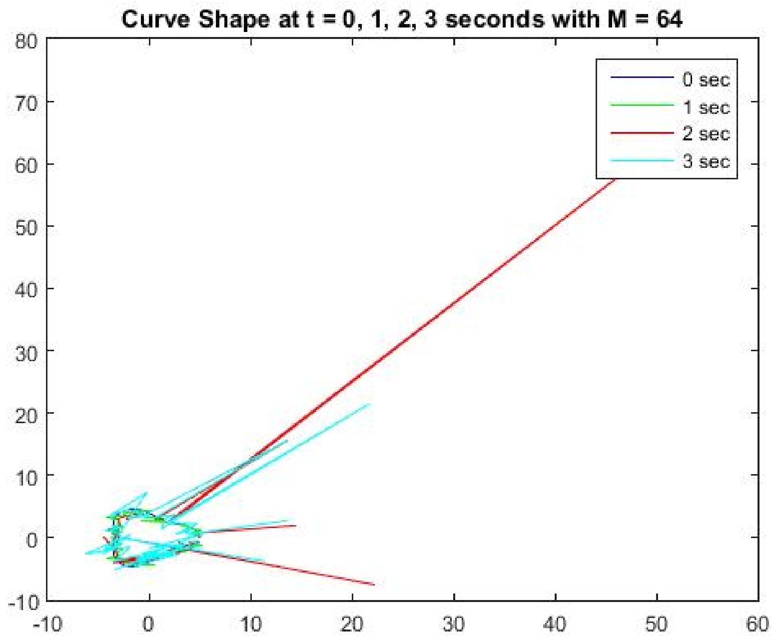

Figure 2: This is M = 64, at time t =4 a curve was not plotted as the curve's image would not allow the window to display the t = 0,1,2,3.