Differential Equations and Moving Systems

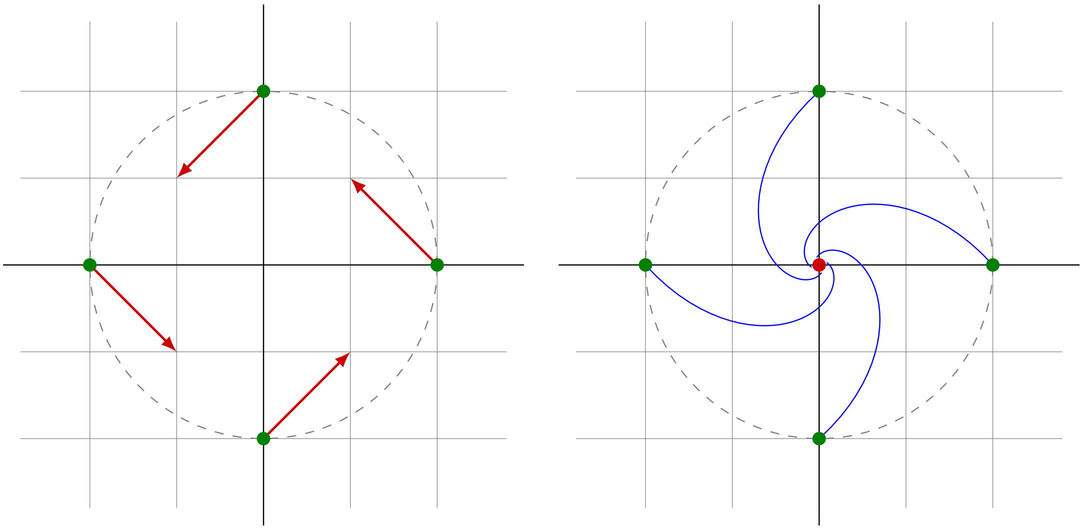
How can differential equations be used to describe a system of 4 points which move towards each other?

Math

Word count: 3994

1 Introduction

In this paper, I will investigate what happens when there are 4 evenly-spaced points on a circle of radius 1 unit, and each point moves towards another one of the 4 points. An example of this is shown in Fig. 1 below. In diagram (a) below, the green dots are the initial positions of the four points, and the red arrows represent the direction these points will start moving. The blue lines in diagram (b) represent the paths the points will travel along.



(a) Initial positions

(b) Paths of the points

Figure 1: One possible arrangement of 4 points.

When a point A "moves towards" another point B, this means the velocity of A is equal to the vector from A to B, in units per second. I chose 4 points because, after exploring several different starting conditions, the 4 point case seemed to me to be the most interesting.

I first came up with this scenario after I was playing around with the setup of the 3-body problem, a well-known physics problem. I chose to investigate this problem not so much for the literal meaning of the question, but because it lends itself to several creative problem-solving strategies and demonstrates how computers can be used to aid mathematicians. The choice to use 4 points instead of 3 or 5 is also for this reason, as there are some interesting solutions possible with 4 points that do not apply for any other number of points.

I mostly used complex numbers, differential equations, coordinate geometry, and integrals to solve the various problems proposed throughout this investigation.

2 Consecutive Order

First, I will explore the case when each point moves towards the next consecutive point counterclockwise, as shown previously in Fig. 1. My goal for this section was to find four expressions in terms of t, the time in seconds, which described the position of the four points. Once I had explicit expressions for the positions of the four points, I also calculated the distance travelled by each point to show how the expressions can be used to analyze any characteristic of the moving points.

At the start of my investigation, I wrote a computer program that could simulate this situation to get an idea of how the points will behave (the full code can be found in Appendix H). The computer generated the shape in Fig. 2 below, where the blue lines are the paths the 4 points travelled along.

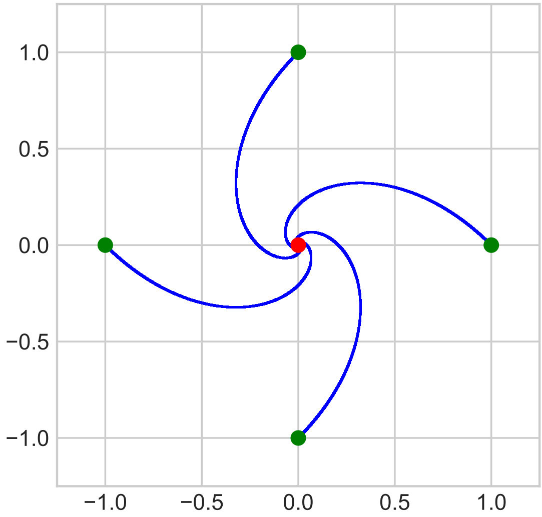


Figure 2: Simulated paths of the 4 points

To generate this image, I plotted the four points at the coordinates $(1, 0)$, $(0, 1)$, $(-1, 0)$, and $(0, -1)$. Then I had each point take a tiny step towards its target repeatedly until the four points reached the center. I used a step size of 0.0001, which means that each point moved 0.0001 of the way to its target point during every step. This can also be thought of as a 0.0001 second jump in time. The simulation stopped when the distance from each point to the origin was less than 0.0005.

2.1 Expressions for the Positions of the Four Points

Once I knew what the four curves roughly looked like, I proceeded to find four expressions which described the positions of the four points at time t.

The graphs of the four curves are rotations of each other. I knew that complex numbers provide a useful algebraic framework to describe rotations in a plane. Therefore, I decided to place the four points on the complex plane, such that the starting points would be at 1, i, -1, and $-i$. I first focused on a single point – the point starting at 1 on the complex plane. Let $f(t)$ equal the complex number representing the position of this point at time t. I proceeded to find an expression for the velocity of $f(t)$. Since the velocity of a position function (such as f) is its derivative, an expression for the velocity would let us set up a differential equation.

Recall that the velocity of the first point is equal to the vector from the first point to this next one. We can find an expression for a vector by taking (tip) – (tail). The next point is a rotation of the first point by $\frac{\pi}{2}$ radians (90 degrees) counterclockwise at all times. This means that the next point is equal to $f(t) \cdot i$, because multiplication by i in the complex plane is equivalent to a counterclockwise rotation by $\frac{\pi}{2}$ radians. Thus, this velocity vector is

$$f(t) \cdot i - f(t) = (i-1)f(t).$$

Since the velocity is equal to the derivative of $f(t)$, I was able to set up the differential equation

$$f'(t) = (i-1)f(t). \tag{1}$$

This differential equation tells us that the derivative of f is proportional to itself. The only functions that have this property are exponential functions; the derivative of an exponential function ae^{bt} with respect to t is $b \cdot ae^{bt}$. Since $f'(t)$ is proportional to $f(t)$ by a factor of $i-1$, the exponential function should look like

$$f(t) = c \cdot e^{(i-1)t} \tag{2}$$

for some constant c. The coefficient c determines the initial position of the function, because if we substitute $t = 0$ into the expression we are just left with c. The function we are looking for has an initial position of 1. Thus, $c = 1$ and our solution is just $f(t) = e^{(i-1)t}$.

Next, I simplified the solution $e^{(i-1)t}$ into a form which is easier to interpret. Using exponent laws, I rewrote this expression in exponential form $re^{i\theta}$ as such:

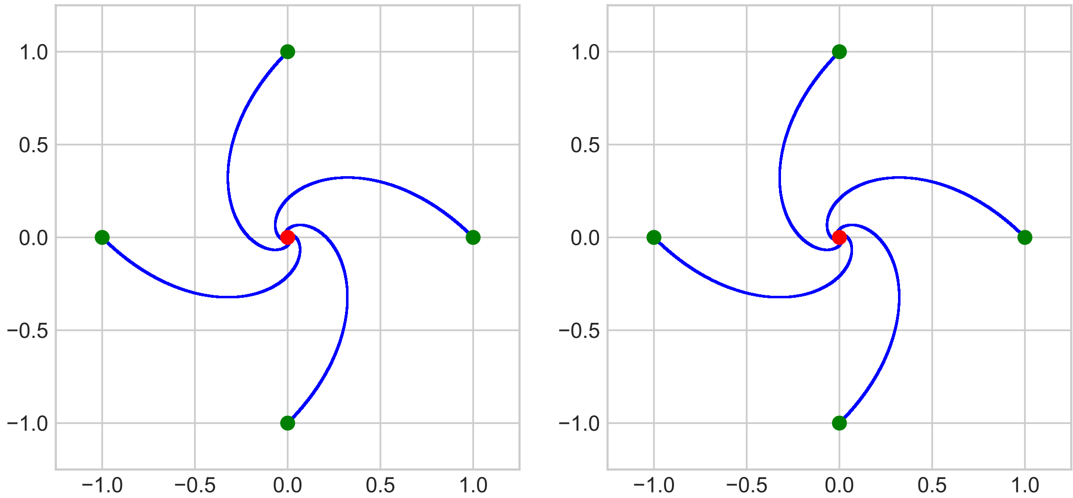
$$e^{(i-1)t} = e^{-t}e^{it}, \tag{3}$$

where $r = e^{-t}$ and $\theta = t$. I recognized this as the graph of a logarithmic spiral, which would typically be represented in polar coordinates as $r = e^{-\theta}$ [8].

The other three functions will be rotations of this function by multiples of $\frac{\pi}{2}$ radians. To achieve these rotation, we can multiply the final expression in (3) by powers of $e^{\frac{\pi}{2}i}$. Therefore, the four expressions for the positions of the four points in counterclockwise order are

$$\begin{aligned}
&e^{-t}e^{it} \\
&e^{-t}e^{i\left(t+\frac{\pi}{2}\right)} \\
&e^{-t}e^{i(t+\pi)} \\
&e^{-t}e^{i\left(t+\frac{3\pi}{2}\right)}.
\end{aligned}$$

To verify that this was correct, I plotted the four parametric functions with a computer next to the simulations from before, as shown in Fig. 3 below. To do this, I first had to express the functions in rectangular form, as shown in section 0.4.1 in Appendix H.



(a) Plot of parametrics

(b) Simulated paths

Figure 3: Plot of parametrics vs. Simulation

As we can see, the parametrics are indistinguishable from the simulation. However, the fact that their paths are the same does not mean they will be at the same position at any point in time. For example, the parametric function I calculated could travel the path very quickly, while in reality the simulation travels very slow. I therefore ran a computer program which tested the distance between the two functions at any point in time. The maximum discrepancy between the simulation and my solution was on the order of 10^{-9}, which supports the fact that my parametric matches the actual behaviour of the points.

2.2 Distance Travelled

Since the four expressions I derived above entirely describe the motion of the four points, we can use them to find any characteristic of the four points. To show an example of this, I decided to compute the total distance travelled by each point.

I first explored the arc length of a general polar function. This problem reminded me of calculating the area under a curve, so I employed the same strategies. We can calculate the area under a curve because we know how the area changes with respect to a small change in the horizontal value – in other words, we know the derivative of the area function. Therefore, I looked at how the arc length L changes with respect to a small change in the angle $d\theta$. Fig. 4 shows us how L changes when we nudge the angle θ by a small amount, $d\theta$.

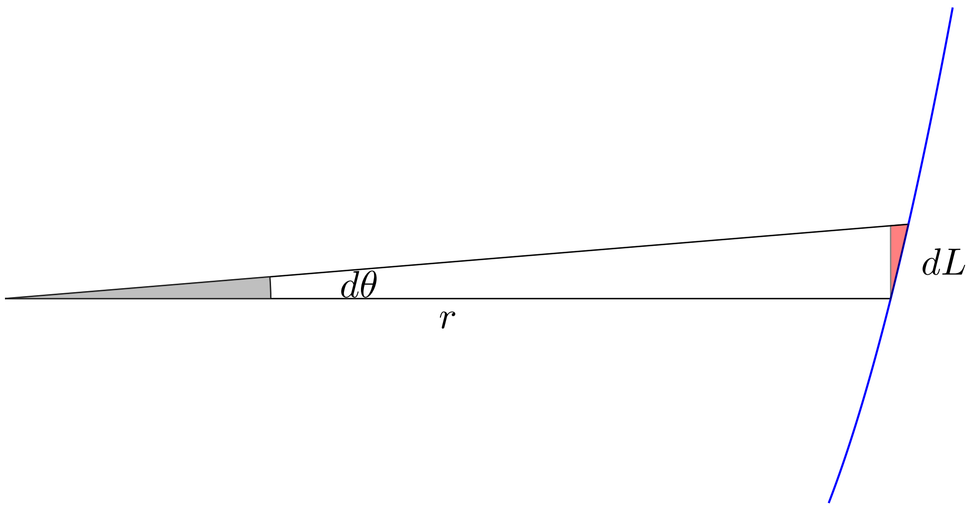


Figure 4: Arc length of a polar function

I attempted to find the small change in arc length dL in terms of $d\theta$. The region in red is formed by drawing a line perpendicular to the radius (the line labelled r). As $d\theta$ approaches 0, this shape will get closer and closer to a right triangle. The hypotenuse of this (almost) right triangle also becomes a better and better approximation for dL. The vertical leg of the right triangle is $r \cdot \sin(d\theta)$, which approaches $r \cdot d\theta$ as $d\theta$ gets closer to 0. The horizontal leg of the triangle is approximately dr, the small change in r. By the Pythagorean theorem, the hypotenuse of the triangle approaches

$$\sqrt{r^2\,d\theta^2 + dr^2}.$$

We can factor out a $d\theta$ from the square root to arrive at

$$dL = \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}\,d\theta.$$

We wish to find the sum of all dL's for infinitely small $d\theta$'s, which is an integral. This means that, for any polar function, the arc length between two angles α and β is given by

$$\int_\alpha^\beta \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}\,d\theta. \tag{4}$$

I was able to verify this formula for the arc length of a polar curve with other sources [1, 2, 9]. The sources I could find used a different method, converting the polar function into a parametric function in rectangular coordinates, to achieve the same result.

The curve our function f traces can be represented in polar coordinates as $r = e^{-\theta}$. The derivative $\frac{dr}{d\theta}$ of this function is $-e^{-\theta}$. This means the arc length of this function from $\theta = \alpha$ to $\theta = \beta$ is

$$\begin{aligned}\int_{\alpha}^{\beta} \sqrt{(e^{-\theta})^2 + (-e^{-\theta})^2}\, d\theta &= \int_{\alpha}^{\beta} \sqrt{2e^{-2\theta}}\, d\theta \\ &= \int_{\alpha}^{\beta} \sqrt{2}e^{-\theta}\, d\theta \\ &= -\sqrt{2}e^{-\beta} + \sqrt{2}e^{-\alpha}.\end{aligned}$$

Substiting $\alpha = 0$ and $\beta = \infty$, we can see the total distance travelled by a single point is $\sqrt{2}$. Although the distance travelled is a finite value, the points would take an infinite amount of time to reach the center of the spiral, as the velocity of the points would decrease exponentially as the points got closer to the middle.

2.3 Conclusion

In this section, I investigated what happens when four evenly spaced, consecutively-ordered points on a circle each move towards the next point. I found four expressions which described the positions of each of the four points at any point in time t, then analyzed the expressions to find the distance travelled by each point. I used the aid of computer simulations to motivate some problem-solving decisions, such as the idea to use the complex plane, as well as verify my answers. This solution also showcased the utility of complex numbers when describing rotations in a plane.

3 Switched Order

Next, I investigated what happens if I switched the targets of the points, so that the rightmost point moves towards the leftmost point, and the leftmost point moves towards the top point as shown in Fig. 5 below. This is the only other possible arrangement of 4 points (in which all 4 points are dependent on each other) excluding rotations and reflections [6]. Just as before, my goal was to find expressions in terms of t representing the positions of the four points, and then analyze the expressions by finding the distance travelled by the points.

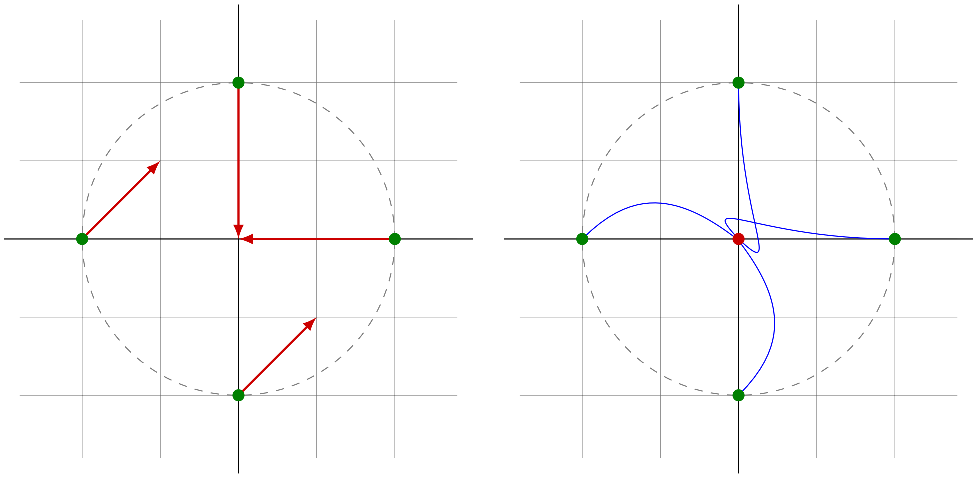


Figure 5: Switched order

I again started by simulating the paths of the four points again using the same process (step size, etc.). This yielded Fig. 6 below.

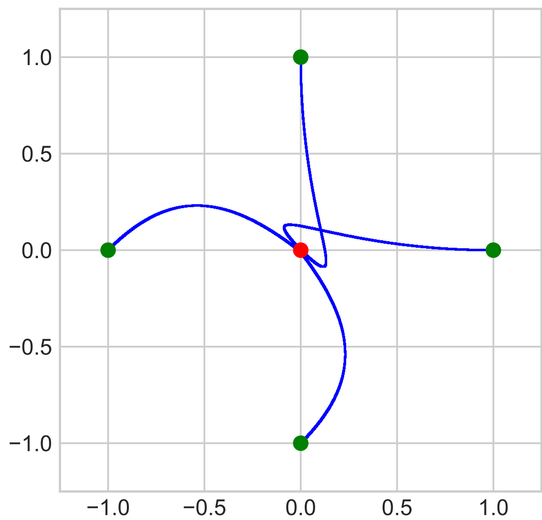


Figure 6: Simulated paths of the 4 points

3.1 Expressions for the Positions of the Four Points

Unlike the previous case, there initially seemed to be no need to describe rotations in the plane, so instead of using complex numbers I used the regular Cartesian coordinate system. Similar to before, I placed the four points at $(1,0)$, $(0,1)$, $(-1,0)$, and $(0,-1)$. As we can see in the simulation above, unlike the previous problem some of the four points follow a completely different path from others. Therefore, instead of defining a single function like before, I defined four seperate functions. Let $f_k(t)$, where k is an ingeter from 1 to 4, be a vector-valued parametric function containing the x- and y-coordinates of the kth point at time t. The first point is the point starting at $(1, 0)$, and each subsequent point is the target of the previous point. For instance, $f_2(t)$ represents the coordinates of the point starting at $(-1, 0)$.

Similar to before, I knew that the derivative of $f_1(t)$ is the vector from $f_1(t)$ to $f_2(t)$ (the derivative of a 2D vector-valued function is also a 2D vector). Therefore, I wrote

$$f_1'(t) = f_2(t) - f_1(t). \tag{5}$$

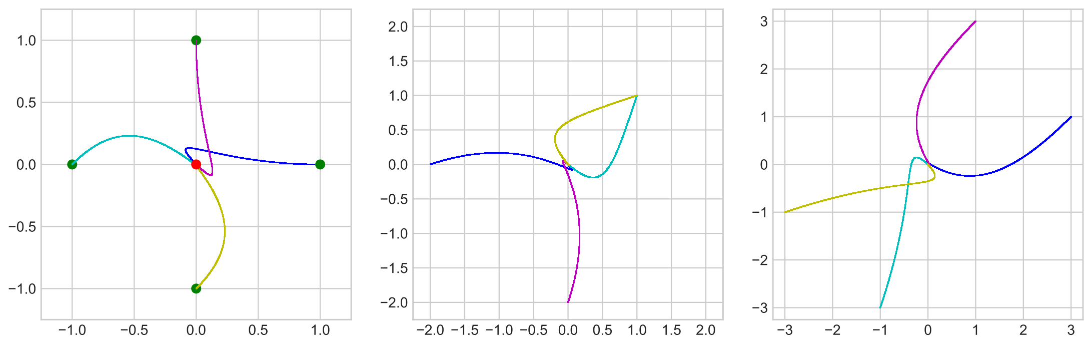
Likewise, the other three functions have derivatives defined by the equations

$$\begin{aligned} f_2'(t) &= f_3(t) - f_2(t) \\ f_3'(t) &= f_4(t) - f_3(t) \\ f_4'(t) &= f_1(t) - f_4(t). \end{aligned}$$

This is a system of 4 differential equations. I initially solved this system through my own method, which is outlined in the following paragraphs. However, later in my investigation, I came across a general tool to solve any system of differential equations (Appendix C) [5, 7].

I felt the derivatives of the functions were going to be important in some way. To allow me to better visualize the derivatives, I wrote a computer program that could take the derivative of my computer simulated functions. To calculate the derivative of one of the four points, I calculated each step the point took and divided that vector by the change in time, which is equivalent to the step size of 0.0001.

I proceeded to simulate the derivatives of the four functions. I didn't immediately notice anything useful so I also graphed their second derivatives, as shown in Fig. 7 (the colors remain consistent – the magenta curve in (b) is the derivative of the magenta curve in (a)).



(a) Original functions (b) First derivatives (c) Second derivatives

Figure 7: Simlations of the derivatives of the four functions

The shape of the second derivatives are somewhat similar to the shape of the original functions. The second derivatives seem to be a flipped, scaled, and "squished" version of the original graphs. I attempted to transform the original functions into the graph of the second derivatives (without worrying about matching the colors) so I could set up a differential equation.

To achieve the flipping and scaling transformations, we can scale the original graphs by a negative factor. The squishing transformation is more complicated. The graphs in (c) look as if they have been squished towards the line $y = x$. We can call this squishing transformation $T(a, b)$. Each point $T(a, b)$ is a certain fraction of the way along the perpendicular from (a, b) to the line $y = x$, as shown in Fig. 8.

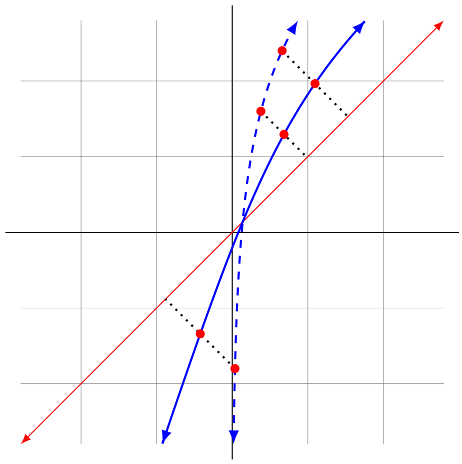


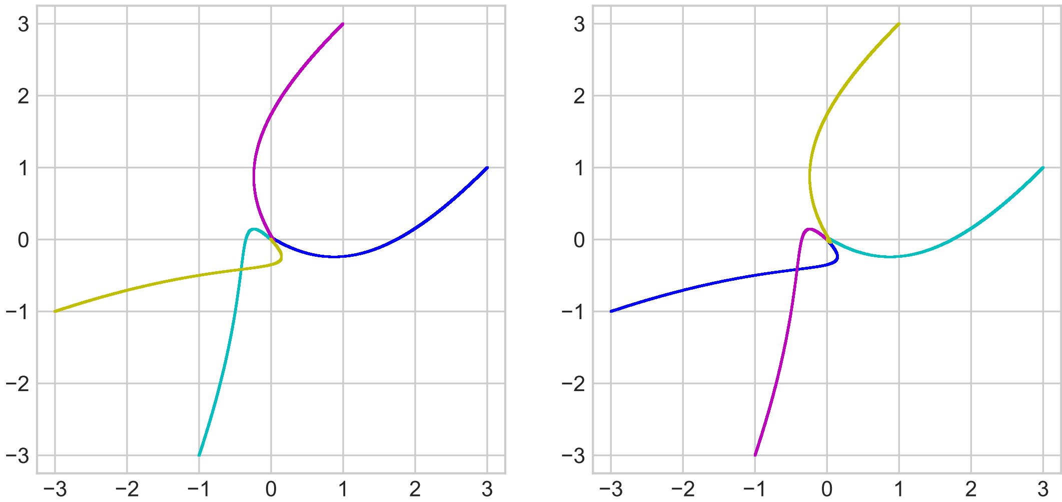
Figure 8: A curve $f(t)$ (dashed) and its transformation $T(f(t))$ (solid).

Next, I tried to determine exactly how much the graphs should be squished. To do this, I looked at the starting points of the second derivatives, which seem to be $(3, 1)$, $(1, 3)$, $(-3, -1)$, and $(-1, -3)$ from Fig. 7 (c). These values are proven to be correct in Appendix A. After flipping and scaling the four functions (before applying the squishing transformation) the starting points will be somewhere along the axes. If we extend lines from the four starting points of the second derivative perpendicular to $y = x$ until they hit the corresponding axes, we reach the four points $(4, 0)$, $(0, 4)$, $(-4, 0)$, and $(0, -4)$. Therefore, it makes sense that the scaling factor would be -4, where we use a negative number to flip the graphs across the origin. The points $(3, 1)$, $(1, 3)$, $(-3, -1)$, and $(-1, -3)$ are exactly halfway from the points $(4, 0)$, $(0, 4)$, $(-4, 0)$, and $(0, -4)$ to the line $y = x$. Thus, I assumed that $T(a, b)$ squished by a factor of $\frac{1}{2}$.

Now, we can determine an algebraic expression for the transformation function $T(a, b)$. The function should take an input point (a, b) and output the midpoint from (a, b) to the line $y = x$. I first found the coordinates of the foot of the perpendicular from (a, b) to this line, which we can call P. Moving our input point in the direction perpendicular to $y = x$, the sum of the x- and y-coordinates remains invariant. Since the sum of the coordinates of (a, b) is $a + b$, the sum of the coordinates of P must also be $a+b$. The point P also lies on the line $y=x$, which means that $P=\left(\frac{a+b}{2}, \frac{a+b}{2}\right)$. $T(a,b)$ is the midpoint of (a,b) and P, which means

$$T(a,b)=\left(\frac{a+\frac{a+b}{2}}{2}, \frac{b+\frac{a+b}{2}}{2}\right)=\left(\frac{3a+b}{4}, \frac{a+3b}{4}\right).$$

Now that I had an idea of what the entire transformation should look like (scaling the points by -4 and then applying T), I used a computer program to transform the four original paths accordingly, as shown in Fig. 9.



(a) Second derivative

(b) Transformed graph

Figure 9: Transformations of the four functions compared with the second derivative

The two graphs are indistinguishable. Just as before, I ran a program to verify the functions were at the same position at the same point in time. The maximum discrepancy between the position of any two corresponding functions at any time was on the order of 10^{-11} units, which is extremely small. From this point onwards, I will assume the second derivatives and the transformations of the four functions are identical. Reasoning off of this assumption will lead to a conjecture which would be difficult to reach otherwise, but is not too difficult to definitively prove once proposed.

First, we can equate matching functions. For example, the blue function in Fig. 9 (a) is identical to the cyan function in (b). From Fig. 7 (a), we know that the blue function represents f_1 and the cyan function represents f_2. Therefore, I wrote the equation

$$f_1''(t)=-4\cdot T(f_2(t)). \tag{6}$$

It is not entirely rigorous to input $f_2(t)$ into T because $f_2(t)$ is technically a vector containing the coordinates of the second point. However, we can imagine using the two coordinates in $f_2(t)$ as the two inputs of T.

We already have an expression for $T(a, b)$, so the next step is to find an expression for $f_1''(t)$. By differentiating (5), we find

$$f_1''(t) = f_2'(t) - f_1'(t) = f_3(t) - 2f_2(t) + f_1(t).$$

Next, I split the vector-valued functions into their x- and y-coordinates. Let

$$\begin{aligned} f_1(t) &= \begin{bmatrix} a_1(t) \\ b_1(t) \end{bmatrix} \\ f_2(t) &= \begin{bmatrix} a_2(t) \\ b_2(t) \end{bmatrix}. \end{aligned}$$

We can express $f_3(t)$ as $\left[\begin{smallmatrix} b_1(t) \\ a_1(t) \end{smallmatrix}\right]$ since it is a reflection of $f_1(t)$ over the line $y = x$. We have

$$\begin{aligned} f_1''(t) &= f_3(t) - 2f_2(t) + f_1(t) \\ &= \begin{bmatrix} b_1(t) \\ a_1(t) \end{bmatrix} - 2\begin{bmatrix} a_2(t) \\ b_2(t) \end{bmatrix} + \begin{bmatrix} a_1(t) \\ b_1(t) \end{bmatrix} \\ &= \begin{bmatrix} a_1(t) + b_1(t) - 2a_2(t) \\ a_1(t) + b_1(t) - 2b_2(t) \end{bmatrix}. \end{aligned}$$

We can also write

$$\begin{aligned} T(f_2(t)) &= T(a_2(t), b_2(t)) \\ &= \left(\frac{3a_2(t) + b_2(t)}{4}, \frac{a_2(t) + 3b_2(t)}{4} \right). \end{aligned}$$

Finally, I substituted the expressions for the x-coordinates of $T(a, b)$ and $f_1''(t)$ into (6):

$$\begin{aligned} a_1(t) + b_1(t) - 2a_2(t) &= -4\left(\frac{3a_2(t) + b_2(t)}{4} \right) \\ &= -3a_2(t) - b_2(t). \end{aligned} \tag{7}$$

Adding $2a_2(t)$ to both sides of (7), I got

$$a_1(t) + b_1(t) = -(a_2(t) + b_2(t)), \tag{8}$$

which seems significant (substituting the y-coordinates yield the same result). This is the aforementioned conjecture which was proposed through the use of computer simulations.

This result seemed relatively easy to verify with a computer, so I wrote a computer program to find the absolute value of the sum of all 4 coordinates of $f_1(t)$ and $f_2(t)$ at any time. If the statement were true, the computer would return a sum close to 0 every time. The program told me that the sum of the coordinates was less than 10^{-14}

away from 0, which gave me confidence that my result was indeed true. I proceeded to mathematically prove this conjecture, as demonstrated in Appendix B.

Thinking in terms of vectors, if we use a change of basis such that one of the new basis vectors is $\overrightarrow{v} = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, the component of any vector along $\overrightarrow{v}$ would be the sum of its original coordinates [3]. Therefore, the components of $f_1(t)$ and $f_2(t)$ along $\overrightarrow{v}$ should be equal and opposite.

I transformed the plane by applying the rotation-dilation matrix

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

as shown in Fig. 10, such that $\overrightarrow{v}$ is the horizontal basis vector. Since the components of the functions along this vector were equal and opposite, the x components of the new functions are equal and opposite. Below, I proceeded to calculate expressions for the four transformed functions, then apply the inverse of A to find the original functions.

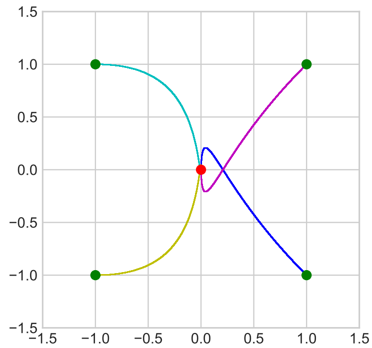


Figure 10: Transformed functions

If we let $g_k(t)$ represent the x component of the transformed function $A \cdot f_k(t)$ for $k = 1, \ldots, 4$, we can write

$$g_1'(t) = g_2(t) - g_1(t) = -2g_1(t).$$

This is a first-order differential equation. Similar to before, the derivative is proportional to the original function, so g_1 must be an exponential function. Specifically, it should look like

$$g_1(t) = c \cdot e^{-2t}$$

for some unknown coefficient c. As in the previous case, to find the exact value of c, we can use the fact that we know the position of this function at time $t = 0$. Moreover, $A \cdot f_1(0)$ represents the point $(1, -1)$, so

$$g_1(0) = c \cdot e^0 = c = 1.$$

This means that $g_1(t)$ is simply

$$g_1(t) = e^{-2t}. \tag{9}$$

Next, let $h_k(t)$ represent the vertical component of the transformed function $A \cdot f_k(t)$ for $k = 1, \ldots, 4$. By symmetry, the vertical components of $A \cdot f_1(t)$ (blue) and $A \cdot f_3(t)$ (magenta) are opposites of each other. Just like before, the derivative of $h_1(t)$ is given by

$$h_1'(t) = h_2(t) - h_1(t),$$

and the derivative of $h_2(t)$ is given by

$$h_2'(t) = h_3(t) - h_2(t).$$

Differentiating the first equation gives us an equation with $h_1(t)$ and $h_3(t)$, which cancel:

$$\begin{aligned} h_1''(t) &= h_2'(t) - h_1'(t) \\ &= h_3(t) - 2h_2(t) + h_1(t) \\ &= -2h_2(t). \end{aligned}$$

The four functions belong to a cyclic system of equations in the sense that each function could be replaced with the next function and the system would look identical. Differentiating this equation twice more will yield an expression with the second derivative of $h_2(t)$, which should contain an $h_3(t) = -h_1(t)$. We have

$$\begin{aligned} h_1^{(4)}(t) &= -2h_2''(t) \\ &= -2(-2h_3(t)) \\ &= 4h_3(t) \\ &= -4h_1(t). \end{aligned} \tag{10}$$

This is a higher-order ordinary differential equation. I solved this equation by paper, as shown in Appendix E, and I verified my answer by using WolframAlpha's differential equation solver [**?**]. The general answer to this differential equation is

$$h_1(t) = c_1 \cdot e^t \cos t + c_2 \cdot e^t \sin t + c_3 \cdot e^{-t} \cos t + c_4 \cdot e^{-t} \sin t \tag{11}$$

for four constants c_1, c_2, c_3, and c_4.

I used several facts I knew about the function h_1 to find the four constants, as shown in Appendix F. We have

$$\begin{aligned} c_1 &= 0 \\ c_2 &= 0 \\ c_3 &= -1 \\ c_4 &= 1. \end{aligned}$$

Thus, $h_1(t)$ is given by

$$h_1(t) = e^{-t}(-\cos t + \sin t). \tag{12}$$

This means we can fully construct the transformed function $A \cdot f_1(t)$ using (9) and (12):

$$\begin{aligned} A \cdot f_1(t) &= \begin{bmatrix} g_1(t) \\ h_1(t) \end{bmatrix} \\ &= \begin{bmatrix} e^{-2t} \\ e^{-t}(-\cos t + \sin t) \end{bmatrix}. \end{aligned} \tag{13}$$

The final step is to multiply this value by the inverse of A to return to the original function. Since A was a rotation by 45 degrees clockwise and a dilation by $\sqrt{2}$, its inverse

A^{-1} should be a rotation by 45 degrees counterclockwise and a dilation by $\frac{1}{\sqrt{2}}$, which is the matrix

$$\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Expanding the product of the vector from (13) and this matrix gives an expression for the original function $f_1(t)$:

$$\begin{aligned} f_1(t) &= \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} e^{-2t} \\ e^{-t}(-\cos t + \sin t) \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}e^{-2t} - \frac{1}{2}e^{-t}(-\cos t + \sin t) \\ \frac{1}{2}e^{-2t} + \frac{1}{2}e^{-t}(-\cos t + \sin t) \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t\right) \\ \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) \end{bmatrix}. \end{aligned}$$

Recall that $f_1(t)$ is a vector-valued parametric function, whose components represent the x- and y-coordinates of the first point (the point that starts at $(1, 0)$) at time t. An expression for $f_3(t)$ can be obtained by reflecting f_1 over the line $y = x$. This can be done by exchangning the x- and y-coordinates, which yields

$$f_3(t) = \begin{bmatrix} \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) \\ \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t\right) \end{bmatrix}$$

To find expressions for the two other functions f_2 and f_4, I first started by looking at f_2 with the differential equation

$$f_2'(t) = f_3(t) - f_2(t). \tag{14}$$

Since $f_2(t)$ and $f_3(t)$ are vector-valued functions, I split them up into their separate parts. Recall our earlier definition

$$f_2(t) = \begin{bmatrix} a_2(t) \\ b_2(t) \end{bmatrix}.$$

We can solve separate differential equations for a_2 and b_2 by taking the x- and y-components of (14):

$$a_2'(t) = \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) - a_2(t) \tag{15}$$

$$b_2'(t) = \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t\right) - b_2(t), . \tag{16}$$

I solved these equation by hand, as shown in Appendix G, and found that the general solutions are

$$a_2(t) = \frac{1}{2}e^{-t}\left(-e^{-t} - \sin t - \cos t - c_1\right) \tag{17}$$

$$b_2(t) = \frac{1}{2}e^{-t}\left(-e^{-t} + \sin t + \cos t - c_2\right). \tag{18}$$

I also verified my general answer using Wolfram Alpha, which yielded the same result [**?**].

First, I found the constant c_1 for $a_2(t)$. The point which $f_2(t)$ represents starts at $(-1,0)$, so $a_2(0) = -1$. Substituting $t = 0$ into (17), we get

$$\frac{1}{2} \cdot 1 \cdot (-1 - 0 - 1 - c) = -1 - \frac{c}{2},$$

which implies $c = 0$. Therefore, $a_2(t)$ is

$$a_2(t) = \frac{1}{2} e^{-t} \left(-e^{-t} - \sin t - \cos t \right).$$

We can follow a similar process with (18) to get

$$b_2(t) = \frac{1}{2} e^{-t} \left(-e^{-t} + \sin t + \cos t \right)$$

Putting these expressions for $a_2(t)$ and $b_2(t)$ together, $f_2(t)$ is given by

$$f_2(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(-e^{-t} - \sin t - \cos t \right) \\ \frac{1}{2} e^{-t} \left(-e^{-t} + \sin t + \cos t \right) \end{bmatrix}.$$

Finally, $f_4(t)$ is a reflection of $f_2(t)$ over the line $y = x$, so we can swap the x- and y-coordinates to get

$$f_4(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(-e^{-t} + \sin t + \cos t \right) \\ \frac{1}{2} e^{-t} \left(-e^{-t} - \sin t - \cos t \right) \end{bmatrix}.$$

In summary, the four parametric functions $f_1(t), \ldots, f_4(t)$ which describe the motion of the four points are

$$f_1(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(e^{-t} + \cos t - \sin t \right) \\ \frac{1}{2} e^{-t} \left(e^{-t} - \cos t + \sin t \right) \end{bmatrix}$$
$$f_2(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(-e^{-t} - \sin t - \cos t \right) \\ \frac{1}{2} e^{-t} \left(-e^{-t} + \sin t + \cos t \right) \end{bmatrix}$$
$$f_3(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(e^{-t} - \cos t + \sin t \right) \\ \frac{1}{2} e^{-t} \left(e^{-t} + \cos t - \sin t \right) \end{bmatrix}$$
$$f_4(t) = \begin{bmatrix} \frac{1}{2} e^{-t} \left(-e^{-t} + \sin t + \cos t \right) \\ \frac{1}{2} e^{-t} \left(-e^{-t} - \sin t - \cos t \right) \end{bmatrix}.$$

To verify my answer, I plotted the four parametrics I got next to the simulation from earlier, as shown in Fig. 11.

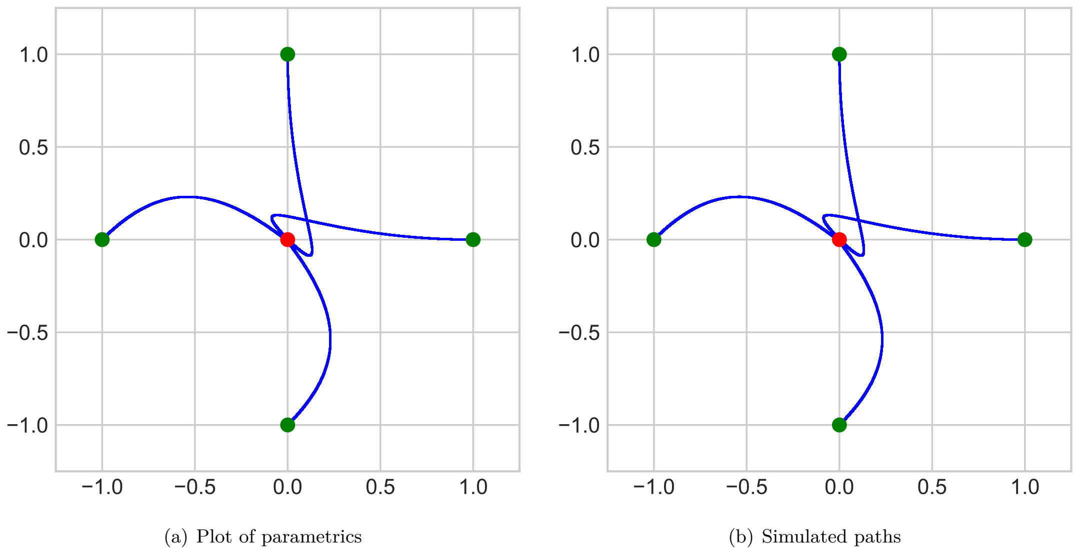


Figure 11: Plot of parametrics vs. Simulation

As we can see, the parametrics match the simulations almost exactly. The maximum discrepancy between the simulation and the parametrics at any point in time was on the order of 10^{-8}, which suggests my formula was correct.

3.2 Distance Travelled

Now that we have the four functions, we can calculate the distance travelled. The functions I found were parametrics in rectangular form, not polar form as before. We can use a similar argument, splitting the function up into many small line segments and taking the limit as these segments become infinitely small. Let us assume that a small nudge in time dt results in a small change in the real coordinate dx, a small change in the imaginary coordinate dy, and a small change in arc length dL.

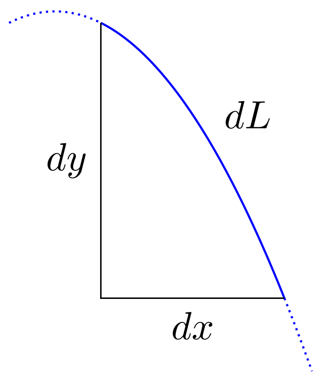


Figure 12: Arc length of a rectangular function

We can see that as dt goes to 0, the change in arc length dL approaches the hypotenuse of the right triangle formed by legs dx and dy. Therefore,

$$dL = \sqrt{dx^2 + dy^2} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\ dt.$$

It follows that the arc length between two values of time t_1 and t_2 is

$$\int_{t_1}^{t_2} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\ dt.$$

For the first parametric function f_1, we have:

$$\begin{aligned}
\frac{dx}{dt} &= \frac{d}{dt}\left(\frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t\right)\right) \\
&= \frac{1}{2}\left(e^{-t} \cdot \frac{d}{dt}\left(e^{-t} + \cos t - \sin t\right) + \frac{d}{dt}e^{-t} \cdot \left(e^{-t} + \cos t - \sin t\right)\right) \\
&= \frac{1}{2}\left(e^{-t}\left(-e^{-t} - \sin t - \cos t\right) - e^{-t}\left(e^{-t} + \cos t - \sin t\right)\right) \\
&= \frac{1}{2}e^{-t}\left(-2e^{-t} - 2\cos t\right) \\
&= -e^{-t}\left(e^{-t} + \cos t\right)
\end{aligned}$$

and

$$\begin{aligned}
\frac{dy}{dt} &= \frac{d}{dt}\left(\frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right)\right) \\
&= \frac{1}{2}\left(e^{-t} \cdot \frac{d}{dt}\left(e^{-t} - \cos t + \sin t\right) + \frac{d}{dt}e^{-t} \cdot \left(e^{-t} - \cos t + \sin t\right)\right) \\
&= \frac{1}{2}\left(e^{-t}\left(-e^{-t} + \sin t + \cos t\right) - e^{-t}\left(e^{-t} - \cos t + \sin t\right)\right) \\
&= \frac{1}{2}e^{-t}\left(-2e^{-t} + 2\cos t\right) \\
&= -e^{-t}\left(e^{-t} + \cos t\right).
\end{aligned}$$

Therefore, the distance travelled by the first point from $t = 0$ to $t = \infty$ is

$$\begin{aligned}
\int_0^\infty \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\ dt &= \int_0^\infty \sqrt{\left(-e^{-t}\left(e^{-t} - \cos t\right)\right)^2 + \left(-e^{-t}\left(e^{-t} + \cos t\right)\right)^2}\ dt \\
&= \int_0^\infty e^{-t}\sqrt{\left(e^{-t} + \cos t\right)^2 + \left(e^{-t} - \cos t\right)^2}\ dt \\
&= \int_0^\infty e^{-t}\sqrt{2e^{-2t} + 2\cos^2 t}\ dt \\
&\approx 1.264.
\end{aligned}$$

There was no closed form for this integral, so I used WolframAlpha to find a numerical approximation [**?**]. Similarly, we can calculate that the second point travels 1.188 units. By symmetry, the third and fourth points will also travel 1.264 and 1.188 units respectively.

3.3 Conclusion

In this section, I investigated what happens when four evenly spaced points, not ordered consecutively, move towards each other. I found four expressions for the positions of the points by using computer simulations and differential equations. As exemplified throughout this solution, computers can be vital to modern math research as they allow us to visualize various mathematical phenomena that would otherwise be impractical to plot or calculate by hand. This allows mathematicians to make better educated hypotheses, which can lead to new discoveries.

Although it wasn't true for my solution, computers can also be an integral part of mathematical proofs as well. A famous example of a proof that used computers is the Appel-Haken proof of the four color theorem.

Throughout this investigation, I explored how differential equations can help us find information about the world around us. Specifically, I explored a system of 4 moving points. I was able to find expressions for the positions of the 4 points for the two possible arrangements, then analyze those expressions to find characteristics of the moving points.

References

[1] "11.4: Area and Arc Length in Polar Coordinates." *Mathematics LibreTexts*, 11 July 2016, tinyurl.com/2zfp25qj.

[2] "Arc Length of Polar Curves." *UT Calculus*, UT Austin, tinyurl.com/2fujzdon. Accessed 9 Sept. 2022.

[3] *Change of Basis*. Directed by Grant Sanderson, vol. 13. *www.youtube.com*, tinyurl.com/223cn8e8. Accessed 17 Nov. 2022.

[4] Dawkins, Paul. "Calculus II - Arc Length with Polar Coordinates." *Paul's Online Notes*, tinyurl.com/yhzxm3wx. Accessed 9 Sept. 2022.

[5] Dawkins, Paul. "Differential Equations - Systems of Differential Equations." *Paul's Online Notes*, tinyurl.com/2g54mc3g. Accessed 9 Sept. 2022.

[6] Golomb, S. W., and L. R. Welch. "On the Enumeration of Polygons." *The American Mathematical Monthly*, vol. 67, no. 4, 1960, pp. 349–53. JSTOR, https://doi.org/10.2307/2308978. Accessed 17 Nov. 2022.

[7] *How (and Why) to Raise e to the Power of a Matrix*. Directed by 3blue1brown, vol. 6. *www.youtube.com*, tinyurl.com/27yz4kjy. Accessed 26 Oct. 2022.

[8] Weisstein, Eric W. "Logarithmic Spiral." *Wolfram MathWorld*, tinyurl.com/y8rremvv. Accessed 9 Sept. 2022.

[9] *WolframAlpha: Making the World's Knowledge Computable*. tinyurl.com/bs5ngw7. Accessed 10 Sept. 2022.

Appendices

A Proof of the Initial Positions of the Second Derivatives

For the switched problem, we claim that at time 0, the second derivatives of the four functions are given by

$$f_1''(0) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$
$$f_2''(0) = \begin{bmatrix} -1 \\ -3 \end{bmatrix}$$
$$f_3''(0) = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$
$$f_4''(0) = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

As shown in the paper, we know that

$$f_1''(0) = f_2'(0) - f_1'(0) = f_3(0) - 2f_2(0) + f_1(0).$$

Similarly, we have the equations

$$\begin{aligned} f_2''(0) &= f_4(0) - 2f_3(0) + f_2(0) \\ f_3''(0) &= f_1(0) - 2f_4(0) + f_3(0) \\ f_4''(0) &= f_2(0) - 2f_1(0) + f_4(0). \end{aligned}$$

Substituting the initial positions for the four functions, we get the desried result:

$$\begin{aligned}
f_1''(0) &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \\
f_2''(0) &= -\begin{bmatrix} 0 \\ 1 \end{bmatrix} - 2\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -3 \end{bmatrix} \\
f_3''(0) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\
f_4''(0) &= -\begin{bmatrix} 1 \\ 0 \end{bmatrix} - 2\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.
\end{aligned}$$

B Proof of Equation (8)

As explained in the paper, I applied the following rotation-dilation matrix to the four functions:

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

I wished to show the real components of the new transformed functions f_1 and f_2 were equal and opposite. Let $g_k(t)$ represent the x-coordinate of $f_k(t)$ for $k = 1, 2, 3$. Taking just the x component of (5), we get

$$g_1'(t) = g_2(t) - g_1(t).$$

Since $g_3(t) = g_1(t)$ by symmetry, we can also write

$$\begin{aligned}
g_2'(t) &= g_3(t) - g_2(t) \\
&= g_1(t) - g_2(t) \\
&= -g_1'(t).
\end{aligned}$$

Since the initial positions of g_1 and g_2 are equal and opposite, and their derivatives are equal and opposite, $g_1(t) = -g_2(t)$.

C Matrix Differential Equations

Matrix differential equations can be used to solve any system of differential equations, such as the ones I dealt with in my investigation (*How (and Why) to Raise e to the Power of a Matrix*). Below is a solution for the problem proposed in Section 3 using a matrix differential equation.

We start by defining the same 4 functions $f_1(t) \dots f_4(t)$, in the same way as the solution from the paper. We had the system of differential equations

$$\begin{aligned}
f_1'(t) &= (f_2(t) - f_1(t)) \\
f_2'(t) &= (f_3(t) - f_2(t)) \\
f_3'(t) &= (f_4(t) - f_3(t)) \\
f_4'(t) &= (f_1(t) - f_4(t)) \, .
\end{aligned}$$

We can express this system using a single equation by packing all four functions into a single 4-dimensional vector $\vec{\mathbf{v}}$:

$$\vec{\mathbf{v}}(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \\ f_4(t) \end{bmatrix} \tag{19}$$

Rewriting the system of equations above, we have

$$\begin{bmatrix} f_1'(t) \\ f_2'(t) \\ f_3'(t) \\ f_4'(t) \end{bmatrix} = \vec{\mathbf{v}}'(t) = \begin{bmatrix} f_2(t) - f_1(t) \\ f_3(t) - f_2(t) \\ f_4(t) - f_3(t) \\ f_1(t) - f_4(t) \end{bmatrix} .$$

We can "factor out" a matrix from the vector on the right hand side so that we have the original vector $\vec{\mathbf{v}}$:

$$\begin{bmatrix} f_2(t) - f_1(t) \\ f_3(t) - f_2(t) \\ f_4(t) - f_3(t) \\ f_1(t) - f_4(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \\ f_4(t) \end{bmatrix} .$$

Let A represent the 4 by 4 matrix in the above equation. We have

$$\vec{\mathbf{v}}'(t) = A\vec{\mathbf{v}}(t). \tag{20}$$

We can see that the derivative of $\vec{\mathbf{v}}$ is itself multiplied by a factor of A. Therefore, the original function could look something like

$$\vec{\mathbf{v}}(t) = e^{At}. \tag{21}$$

Although it may seem nonsensical to take raise e to the power of a matrix, this function is actually well-defined. To calculate the exponential of a matrix, we can insert it into the Taylor series for the exponential function. The notation $\exp(x)$ is often used to denote this process of inserting an input x into the Taylor series for the exponential. Another concern is that in the right hand side of (20), we have matrix multiplication, not standard multiplication. We will later see that this property – that the derivative of the exponential of a value is proportional to itself – still holds for matrix differential equations.

The matrix exponential is defined using the Taylor series for the exponential function, which is typically

$$\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots .$$

The matrix exponential looks very similar, except that instead of the leading 1 we have the identity matrix I. This is because the first term is actually $\frac{x^0}{0!}$, and a matrix raised to the power of 0 is always the identity matrix. Therefore, the matrix exponential is defined as

$$\exp(M) = I + \frac{M}{1!} + \frac{M^2}{2!} + \frac{M^3}{3!} + \cdots ,$$

which has a matrix output. We can verify that the derivative $\frac{d}{dt}\exp(tM)$ is indeed $M\exp(tM)$. We have

$$\begin{aligned}\frac{d}{dt}\exp(tM) &= \frac{d}{dt}\left(I + tM + \frac{1}{2!}t^2M^2 + \frac{1}{3!}t^3A^3 + \frac{1}{4!}t^4A^4 + \cdots\right)\\ &= M + tM^2 + \frac{1}{2!}t^2M^3 + \frac{1}{3!}t^3M^4 + \cdots\\ &= M\exp(Mt),\end{aligned}$$

as desired. This assuages our earlier concern that, since the right hand side of (20) has matrix multiplication, equation (21) may not be true. We can see that differentiating both sides of (21) with respect to t results in (20), so this is certainly one possible solution.

Returning to (21), we think of the exponential not as a solution but as something that acts on a given initial condition. In the matrix differential equation case, the exponential part is a matrix, and the initial condition is a vector, which I will call $\vec{\mathbf{v}}(0)$. In general, solutions which have the form

$$\vec{\mathbf{v}}(t) = \exp(tA) \cdot \vec{\mathbf{v}}(0) \tag{22}$$

will satisfy eqution (20).

Next, we can actually compute the matrix exponential $\exp(tA)$. The first few terms of this series are shown below.

$$\exp(tA) = \exp\left(\begin{bmatrix} -t & t & 0 & 0 \\ 0 & -t & t & 0 \\ 0 & 0 & -t & t \\ t & 0 & 0 & -t \end{bmatrix}\right)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -t & t & 0 & 0 \\ 0 & -t & t & 0 \\ 0 & 0 & -t & t \\ t & 0 & 0 & -t \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} -t & t & 0 & 0 \\ 0 & -t & t & 0 \\ 0 & 0 & -t & t \\ t & 0 & 0 & -t \end{bmatrix}^2 + \cdots$$

$$= I + \begin{bmatrix} -t & t & 0 & 0 \\ 0 & -t & t & 0 \\ 0 & 0 & -t & t \\ t & 0 & 0 & -t \end{bmatrix} + \frac{1}{2}\begin{bmatrix} t^2 & -2t^2 & t^2 & 0 \\ 0 & t^2 & -2t^2 & t^2 \\ t^2 & 0 & t^2 & -2t^2 \\ -2t^2 & t^2 & 0 & t^2 \end{bmatrix}$$

$$+ \frac{1}{6}\begin{bmatrix} -t^3 & 3t^3 & -3t^3 & t^3 \\ t^3 & -t^3 & 3t^3 & -3t^3 \\ -3t^3 & t^3 & -t^3 & 3t^3 \\ 3t^3 & -3t^3 & t^3 & -t^3 \end{bmatrix} + \frac{1}{24}\begin{bmatrix} 2t^4 & -4t^4 & 6t^4 & -4t^4 \\ -4t^4 & 2t^4 & -4t^4 & 6t^4 \\ 6t^4 & -4t^4 & 2t^4 & -4t^4 \\ -4t^4 & 6t^4 & -4t^4 & 2t^4 \end{bmatrix} + \cdots .$$

Now, let us focus on any single term in the answer – for example, the term in the top left. That term will be equal to $1 - t + \frac{1}{2}t^2 - \frac{1}{6}t^3 + \frac{1}{24}\left(2t^4\right)\cdots$. It is unclear whether any pattern emerges from only the first 5 terms, so I simplified the terms and calculated a few more:

$$1 - t + \frac{t^2}{2} - \frac{t^3}{6} + \frac{t^4}{12} - \frac{t^5}{20} + \frac{t^6}{45} - \frac{t^7}{140} + \frac{t^8}{560} - \cdots .$$

The pattern is still not obvious, but the coefficient of x^n is $\frac{1}{n!}$ times the sum of every fourth term in the nth row of Pascal's triangle, or $\frac{\binom{n}{0}+\binom{n}{4}+\binom{n}{8}+\binom{n}{12}+\cdots}{n!}$ Using a computer, I found that this is the Maclaurin series for

$$\frac{e^{-2t}}{4} + \frac{1}{2}e^{-t}\cos t + \frac{1}{4},$$

but there is no method of obtaining the closed form for such an infinite series without computational assistance and some guesswork.

We can go through each of the 16 positions of the 4 by 4 matrix and calculate the closed form of each one to find the answer to the matrix exponential. The actual answer is too large to fit on this page. With a few substitutions, the answer is

$$\begin{bmatrix} a+p+\frac{1}{4} & -a+q+\frac{1}{4} & a-p+\frac{1}{4} & -a-q+\frac{1}{4} \\ -a-q+\frac{1}{4} & a+p+\frac{1}{4} & -a+q+\frac{1}{4} & a-p+\frac{1}{4} \\ a-p+\frac{1}{4} & -a-q+\frac{1}{4} & a+p+\frac{1}{4} & -a+q+\frac{1}{4} \\ -a+q+\frac{1}{4} & a-p+\frac{1}{4} & -a-q+\frac{1}{4} & a+p+\frac{1}{4} \end{bmatrix}$$

where $a = \frac{e^{-2t}}{4}$, $p = \frac{1}{2}e^{-t}\cos t$, and $q = \frac{1}{2}e^{-t}\sin t$. Let this matrix be represented by M below.

There is another method of computing the matrix exponential, by first diagonalizing the original matrix, that is more powerful in the sense that it leaves us with the closed form solution without the need to use a computer. The full calculations are shown in Appendix D.

Now that we have our matrix exponential, we only need to find an initial condition, $\vec{\mathbf{v}}(t_0)$, for some known value of time t_0. I decided to use the complex plane so we would only have 1 matrix differential equation, but this could also be solved through normal coordinates using 2 separate matrix differential equations for the x- and y-components of the functions The four functions start, in order, at 1, -1, i, and $-i$, so we can set $t_0 = 0$ for the initial condition vector

$$\vec{\mathbf{v}}(0) = \begin{bmatrix} 1 \\ -1 \\ i \\ -i \end{bmatrix}.$$

Finally, we can substitute these two pieces into (22) and compute the actual function $\vec{\mathbf{v}}(t)$. We have

$$\vec{\mathbf{v}}(t) = \frac{1}{2}e^{-t} \cdot \begin{bmatrix} e^{-t} + \cos t - \sin t + \left(e^{-t} - \cos t + \sin t\right) i \\ -e^{-t} - \cos t - \sin t + \left(-e^{-t} + \cos t + \sin t\right) i \\ e^{-t} - \cos t + \sin t + \left(e^{-t} + \cos t - \sin t\right) i \\ -e^{-t} + \cos t + \sin t + \left(-e^{-t} - \cos t - \sin t\right) i \end{bmatrix}.$$

Recall that $\vec{\mathbf{v}}(t)$ contains the four functions $f_1(t), \ldots, f_4(t)$ in order (see (19)). Thus, the four functions are

$$\begin{aligned}
f_1(t) &= \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t + \left(e^{-t} - \cos t + \sin t\right) i\right) \\
f_2(t) &= \frac{1}{2}e^{-t}\left(-e^{-t} - \cos t - \sin t + \left(-e^{-t} + \cos t + \sin t\right) i\right) \\
f_3(t) &= \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t + \left(e^{-t} + \cos t - \sin t\right) i\right) \\
f_4(t) &= \frac{1}{2}e^{-t}\left(-e^{-t} + \cos t + \sin t + \left(-e^{-t} - \cos t - \sin t\right) i\right),
\end{aligned}$$

which is the exact same answer as the one given in the paper.

The value of this approach is that it can easily be extended for any set of initial conditions. If the four functions started out in four random locations on the complex plane, say z_1, z_2, z_3, and z_4, we could simply multiply the same matrix exponential by the new initial condition vector $(z_1\ z_2\ z_3\ z_4)$ to find the four corresponding functions. We can also generalize to any number of points by following the same process of setting up a system of differential equations and solving using the matrix exponential. This allows us to find explicit expressions for the paths of points in any system such as the one shown in Fig. 13. However, the complexity of the matrix exponential increases drastically from 5 points and onwards.

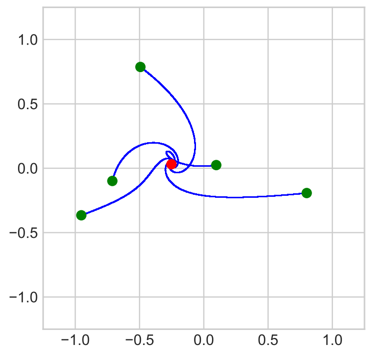


Figure 13: A general system

D Calculating the Matrix Exponential by Diagonalization

The closed form of the matrix exponential from Appendix C, and the exponential of any diagonalizable matrix in general, can be calculated by employing diagonalization (*WolframAlpha*). A square matrix M is diagonalizable if its eigenvectors, when used as basis vectors, can span the entire vector space that M applies to. For example, for a 2 by 2 matrix to be diagonalizable, its eigenvectors must span the 2D plane. Luckily, our matrix A from the solution in Appendix C is indeed diagonalizable. Even if our matrix wasn't diagonalizable, there is a general way to calculate the matrix exponential, but it has a few more steps (the process involves decomposing the non-diagonalizable matrix into a diagonalizable matrix and a nilpotent matrix).

We can first diagonalize the matrix using its eigenvalues. For our matrix tA, we have the eigenvalues $\lambda_1 = -2t$, $\lambda_2 = (-1+i)t$, $\lambda_3 = (-1-i)t$, and $\lambda_4 = 0$. The correspondng eigenvectors are

$$\begin{aligned} v_1 &= (-1, 1, -1, 1) \\ v_2 &= (-i, -1, i, 1) \\ v_3 &= (i, -1, -i, 1) \\ v_4 &= (1, 1, 1, 1). \end{aligned}$$

This means that our matrix tA can be diagonalized as $tA = PDP^{-1}$ where

$$P = \begin{bmatrix} -1 & -i & i & 1 \\ 1 & -1 & -1 & 1 \\ -1 & i & -i & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -2t & 0 & 0 \\ 0 & 0 & (-1-i)t & 0 \\ 0 & 0 & 0 & (-1+i)t \end{bmatrix}$$

$$P^{-1} = \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ \frac{i}{4} & -\frac{1}{4} & -\frac{i}{4} & \frac{1}{4} \\ -\frac{i}{4} & -\frac{1}{4} & \frac{i}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Thus, we have

$$\begin{aligned} \exp(tA) &= I + \frac{tA}{1} + \frac{(tA)^2}{2} + \frac{(tA)^3}{6} + \cdots \\ &= I + \frac{PDP^{-1}}{1} + \frac{PDP^{-1}PDP^{-1}}{2} + \frac{PDP^{-1}PDP^{-1}PDP^{-1}}{6} + \cdots \\ &= I + \frac{PDP^{-1}}{1} + \frac{PD^2P^{-1}}{2} + \frac{PD^3P^{-1}}{6} + \cdots \\ &= P\exp(D)\,P^{-1}. \end{aligned}$$

To find the exponential of a diagonal matrix, we can just exponentiate all the terms on the major diagonal. This means

$$\exp(D) = \begin{bmatrix} e^0 & 0 & 0 & 0 \\ 0 & e^{-2t} & 0 & 0 \\ 0 & 0 & e^{(-1-i)t} & 0 \\ 0 & 0 & 0 & e^{(-1+i)t} \end{bmatrix}.$$

Finally, we can compute the matrix exponential by computing the matrix product

$$\begin{bmatrix} -1 & -i & i & 1 \\ 1 & -1 & -1 & 1 \\ -1 & i & -i & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} e^0 & 0 & 0 & 0 \\ 0 & e^{-2t} & 0 & 0 \\ 0 & 0 & e^{(-1-i)t} & 0 \\ 0 & 0 & 0 & e^{(-1+i)t} \end{bmatrix} \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ \frac{i}{4} & -\frac{1}{4} & -\frac{i}{4} & \frac{1}{4} \\ -\frac{i}{4} & -\frac{1}{4} & \frac{i}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

With a few substitutions, the answer to this multiplication is

$$\begin{bmatrix} a+p+\frac{1}{4} & -a+q+\frac{1}{4} & a-p+\frac{1}{4} & -a-q+\frac{1}{4} \\ -a-q+\frac{1}{4} & a+p+\frac{1}{4} & -a+q+\frac{1}{4} & a-p+\frac{1}{4} \\ a-p+\frac{1}{4} & -a-q+\frac{1}{4} & a+p+\frac{1}{4} & -a+q+\frac{1}{4} \\ -a+q+\frac{1}{4} & a-p+\frac{1}{4} & -a-q+\frac{1}{4} & a+p+\frac{1}{4} \end{bmatrix}$$

where $a = \frac{e^{-2t}}{4}$, $p = \frac{1}{2}e^{-t}\cos t$, and $q = \frac{1}{2}e^{-t}\sin t$. As we can see, the answer is equal to the one outlined in the paper. Even though the diagonalization of tA contained some complex terms, the final exponential is real. This serves as a rudimentary check that all the calculations were done correctly, as all the terms in the final answer can be represented by infinite series of real numbers as shown before, which means they should not be complex.

E Solution for Equation (10)

For reference, equation (10) is

$$h_1^{(4)}(t) = -4h_1(t).$$

Below is a solution to the equation without the aid of WolframAlpha. If we let $y = h_1(t)$, then the differential equation we wish to solve is

$$y^{(4)} + 4y = 0. \tag{23}$$

Similar to the argument from Section 2, the only functions whose derivative is proportional to itself are exponential functions. Differentiating an exponential multiple times will also yield an expression proportional to the original function. Equation 23 shows that the function y is proportional to its fourth derivative. Therefore, I assumed that one solution to this differential equation was an exponential,

$$y = e^{\lambda t}.$$

Substituting this back into the left-hand side of the differential equation, I got

$$\begin{aligned} y^{(4)} + 4y &= \frac{d^4y}{dt^4}e^{\lambda t} + 4e^{\lambda t} \\ &= \lambda^4 e^{\lambda t} + 4e^{\lambda t} \\ &= e^{\lambda t}\left(\lambda^4 + 4\right). \end{aligned}$$

I knew that this expression was equal to 0. Since $e^{\lambda t}$ cannot be 0, I deduced that $\lambda^4 + 4$ must be equal to 0. This means that $\lambda^4 = -4$. However, this implies λ must be complex, but $h_1(t)$ must be real-valued. To overcome this, we can consider complex values for λ, then add or subtract solutions to find new, entirely real or imaginary solutions, as outlined in the paragraphs below.

By DeMoivre's theorem, if we let λ be the complex point (r, θ) in polar coordinates, λ^4 equals the point $(r^4, 4\theta)$. The point -4 in polar coordinates is $(4, \pi)$. This means that $r^4 = 4$ and thus $r = \sqrt{2}$ (r must be a positive real number). However, 4θ is not necessarily π, as it could also equal 3π or 5π and so on, as these angles are all the same. Because $\theta < 2\pi$, we know that $4\theta < 8\pi$. Therefore, the only possibilities are π, 3π, 5π, and 7π. The corresponding values of θ are $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, and $\frac{7\pi}{4}$. This means the four possible values for λ are (in polar coordinates) $\left(\sqrt{2}, \frac{\pi}{4}\right)$, $\left(\sqrt{2}, \frac{3\pi}{4}\right)$, $\left(\sqrt{2}, \frac{5\pi}{4}\right)$, and

$\left(\sqrt{2}, \frac{7\pi}{4}\right)$. These convert nicely into the complex numbers $1+i$, $-1+i$, $-1-i$, and $1-i$. For simplicity, let λ_1, λ_2, λ_3, and λ_4 respectively represent these four complex numbers. The corresponding solutions to 23 are:

$$\begin{aligned}
e^{\lambda_1 t} &= e^t e^{it} = e^t(\cos t + i \sin t) \\
e^{\lambda_2 t} &= e^{-t} e^{it} = e^{-t}(\cos t + i \sin t) \\
e^{\lambda_3 t} &= e^{-t} e^{-it} = e^{-t}(\cos(-t) + i \sin(-t)) = e^{-t}(\cos t - i \sin t) \\
e^{\lambda_4 t} &= e^t e^{-it} = e^t(\cos(-t) + i \sin(-t)) = e^t(\cos t - i \sin t).
\end{aligned}$$

As stated earlier, $h_1(t)$ must be real-valued, while these solutions are clearly complex. We can get around this issue by adding or subtracting two solutions together (any linear combination of solutions to a differential equation is also a solution). I noticed the imaginary components in the second and third solutions are opposites of each other. Therefore, their sum is a solution with no imaginary part. We have

$$e^{-t}(\cos t + i \sin t) + e^{-t}(\cos t - i \sin t) = 2e^{-t} \cos t.$$

It is slightly cleaner to add half of the second and third solutions, which yields $e^{-t} \cos t$. Similarly, adding half the first and fourth solutions yields another real solution:

$$\frac{1}{2} e^t(\cos t + i \sin t) + \frac{1}{2} e^t(\cos t - i \sin t) = e^t \cos t.$$

Another way we can get valid solutions is by taking some linear combination of solutions that is purely imaginary. To achieve a purely imaginary result, we can subtract the same pairs of equations as before. Taking the second solution minus the third solution, we get

$$e^{-t}(\cos t + i \sin t) - e^{-t}(\cos t - i \sin t) = 2ie^{-t} \sin t.$$

Now, we notice that there is an extra $2i$ in the solution. Hence, multiplying the two original solutions by $\frac{1}{2i}$, we get

$$\frac{1}{2i} e^{-t}(\cos t + i \sin t) - \frac{1}{2i} e^{-t}(\cos t - i \sin t) = e^{-t} \sin t.$$

Finally, working through the the same process with the first and last solutions, we get

$$\frac{1}{2i} e^t(\cos t + i \sin t) - \frac{1}{2i} e^t(\cos t - i \sin t) = e^t \sin t.$$

Therefore, our general real-valued solution to this differential equation is any linear combination of these four real solutions. In other words,

$$h_1(t) = c_1 \cdot e^t \cos t + c_2 \cdot e^t \sin t + c_3 \cdot e^{-t} \cos t + c_4 \cdot e^{-t} \sin t.$$

.

F Calculating the Constants in Equation (11)

For reference, equation (11) is

$$h_1(t) = c_1 \cdot e^t \cos t + c_2 \cdot e^t \sin t + c_3 \cdot e^{-t} \cos t + c_4 \cdot e^{-t} \sin t.$$

Recall that $h_1(t)$ represents the y value of $A \cdot f_1(t)$.

Generally, we would use the initial position of the function as well as the initial values of its first three derivatives to find the constants, but differentiating the expression above 3 times seemed like a handful. Instead, I used the fact that the four moving points approach the origin to find two of the constants quickly.

I knew that as t approaches infinity, $h_1(t)$ approaches 0. In other words, $\lim_{t\to\infty} h_1(t) = 0$. If t tends to infinity, any terms containing an e^{-t} approach 0, so we are left with

$$\lim_{t\to\infty} e^t \left(c_1 \cos t + c_2 \sin t\right) = 0.$$

Since e^t grows exponentially large as t approaches infinity, the value inside the parentheses must approach 0. I graphed a few examples of linear combinations of sine and cosine functions, and noticed that the result was always a sine wave. Indeed, after a bit of research, I found that $c_1 \sin t + c_2 \cos t$ can be written as a cosine wave with $t - X$ as its input (the phase shift X is a constant dependent on c_1 and c_2). Since a cosine wave has no limit as its input approaches infinity, for the value inside the parentheses to approach 0, we must have $c_1 = c_2 = 0$. Therefore, because the terms containing c_1 and c_2 are 0, $h_1(t) = e^{-t}\left(c_3 \cos t + c_4 \sin t\right)$.

Now, we can use the initial position $f_1(t)$ represents to find another one of the constants. We have $A \cdot f_1(0) = (1, -1)$, so $h_1(0) = -1$. I got

$$h_1(0) = e^0(c_3(1) + c_4(0)) = c_3 = -1.$$

Finally, to determine c_4, we can use the derivative of h_1 at time $t = 0$. We have

$$h_1'(0) = h_2(0) - h_1(0) = 1 - (-1) = 2.$$

I also computed the derivative of $h_1(t)$ from its components using the product rule:

$$\begin{aligned}
h_1'(t) &= \frac{d}{dt}\left(e^{-t}\left(-\cos t + c_4 \sin t\right)\right) \\
&= e^{-t} \cdot \frac{d}{dt}\left(-\cos t + c_4 \sin t\right) + \frac{d}{dt}\left(e^{-t}\right) \cdot \left(-\cos t + c_4 \sin t\right) \\
&= e^{-t}\left(\sin t + c_4 \cos t\right) - e^{-t}\left(-\cos t + c_4 \sin t\right) \\
&= e^{-t}\left(\left(-c_4 + 1\right)\sin t + \left(c_4 + 1\right)\cos t\right).
\end{aligned}$$

Substituting $t = 0$ yields

$$h_1'(0) = c_4 + 1.$$

We know that this expression is equal to 2, and subsequently $c_4 = 1$.

G Solutions to Equations (15) and (16)

For reference, the equations (15) and (16) are

$$a_2'(t) = \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) - a_2(t)$$
$$b_2'(t) = \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t\right) - b_2(t).$$

First, I solved the upper equation, a first-order linear ordinary differential equation. There exists a well-established method of solving this kind of differential equation, which I proceed to walk through below. For simplicity, from this point onwards let $y = a_2(t)$. The first step is to rearrange the differential equation so that all the terms containing $a_2(t)$ are on the left-hand side:

$$\frac{dy}{dt} + y = \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right). \tag{24}$$

Next, we multiply the whole equation by an integration factor, which in our case is e^t (the motivation for this choice will be explained in the next step). We have

$$e^t \cdot \frac{dy}{dt} + e^t \cdot y = e^t \cdot \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right). \tag{25}$$

Now, we might notice the left-hand side of this equation is a product rule for the derivative

$$\left(e^t \cdot y\right)'.$$

Indeed, the integration factor was chosen precisely so that the left hand side looks like a product rule. Substituting this into (25), we get

$$\left(e^t \cdot y\right)' = e^t \cdot \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right).$$

Since we are looking for y, we can integrate both sides to get rid of the derivative on the left-hand side:

$$\int \left(e^t \cdot y\right)' \, dt = \int e^t \cdot \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) \, dt$$
$$e^t \cdot y + c = \int e^t \cdot \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) \, dt.$$

Finally, subtracting c and dividing by e^t, we get

$$\begin{aligned} y &= \frac{\int e^t \cdot \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t\right) \, dt - c}{e^t} \\ &= \frac{\int e^{-t} - \cos t + \sin t \, dt - c}{2e^t} \\ &= \frac{-e^{-t} - \sin t - \cos t - c}{2e^t} \\ &= \frac{1}{2}e^{-t}\left(-e^{-t} - \sin t - \cos t - c_1\right). \end{aligned} \tag{26}$$

We can follow a similar process with (16), multiplying by e^t and doing some algebra to find

$$b_2(t) = \frac{1}{2}e^{-t}\left(-e^{-t} + \sin t + \cos t - c_2\right).$$

H Computer Code

Listed below is all the Python console code I wrote for the paper. The page numbers & numbering system are incorrect below, since I integrated a separate pdf into this document.

Code for Extended Essay

August 28, 2022

0.1 Code Initialization

[1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import math, cmath, random
import numpy as np
import sympy as sp
sp.init_printing(use_unicode=True)
```

0.2 Functions

[2]:
```
def reset_pts(n=4):

    points = []

    for k in range(n):
        arg = k*2*math.pi/n
        points.append(complex(math.cos(arg), math.sin(arg)))

    return points

def reorder(pts, new_order):
    return [pts[i] for i in new_order]

def convert_to_coords(pts):
    return [p.real for p in pts], [p.imag for p in pts]
```

[3]:
```
def simulate(start_pts, step_size=0.0001, stop_distance=0.0005,␣
 ↪return_center=False):

    n_pts = len(start_pts)
    current_pts = [p for p in start_pts]
    paths = [[start_pts[k]] for k in range(n_pts)]
    center = sum(start_pts) / n_pts

    while True:
        for k in range(n_pts):
```

```
            if k == 0:
                last_target = current_pts[0]
                # store pos of first point, else last point will step towards␣
↪updated pos of first point

            if k == n_pts - 1:
                target = last_target
            else:
                target = current_pts[k+1]

            step = (target - current_pts[k]) * step_size
            current_pts[k] += step

            paths[k].append(current_pts[k])

        if all(abs(p - center) < stop_distance for p in current_pts) or␣
↪any(abs(p - center) > 10 for p in current_pts):
            break

    path_coords = [convert_to_coords(paths[k]) for k in range(n_pts)]

    if return_center:
        center_coords = [center.real, center.imag]

        return path_coords, center_coords
    else:
        return path_coords
```

[4]:
```
def derivative(coords, degree=1, step=0.0001):

    for i in range(degree):
        deriv_coords = []

        for x, y in coords:
            x_deriv = [(x[j+1] - x[j]) / step for j in range(len(x)-1)]
            y_deriv = [(y[j+1] - y[j]) / step for j in range(len(x)-1)]

            deriv_coords.append((x_deriv, y_deriv))

        coords = [deriv_coords[k] for k in range(len(deriv_coords))]

    return deriv_coords
```

[5]:
```
def graph(
    paths=[],
    points=[],
    color='b',
```

```
    c_dot=[],
    axis_lim=[-1.25, 1.25],
    tick=6,
    save=''
):

    plt.axis('square')
    plt.xlim(*axis_lim)
    plt.ylim(*axis_lim)
    plt.locator_params(axis='both', nbins=tick, integer=False)

    for path in paths:
        plt.plot(*path, color+',')

    if points:
        plt.plot(*points, 'go')

    if c_dot:
        plt.plot(*c_dot, 'ro')

    if save:
        plt.savefig('Desktop/EE/Graphs/'+save, format='eps',␣
 ↪bbox_inches='tight')
```

[6]:
```
def parametric(x_func, y_func, t_init=0, t_fin=10, t_step=0.0001):

    coordinates = ([], [])

    for t in [t_init + i*t_step for i in range(math.ceil((t_fin-t_init)/
 ↪t_step))]:
        coordinates[0].append(x_func(t))
        coordinates[1].append(y_func(t))

    return coordinates
```

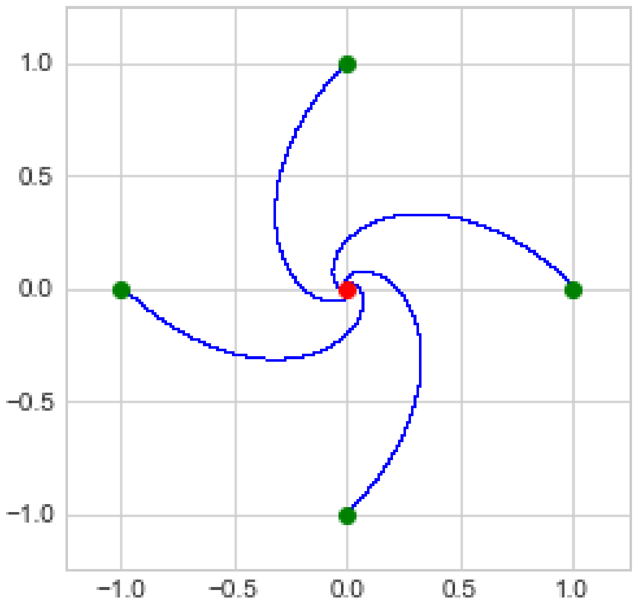
[7]:
```
def max_discrepancy(coords1, coords2):

    max_discr = 0

    for path1, path2 in zip(coords1, coords2):
        for n in range(min(len(path1), len(path2))):
            x_discr = abs(path1[0][n] - path2[0][n])
            y_discr = abs(path1[1][n] - path2[1][n])

            discr = math.sqrt(x_discr ** 2 + y_discr ** 2)
            if discr > max_discr:
                max_discr = discr
```

```python
    return max_discr
```

0.3 Simulations

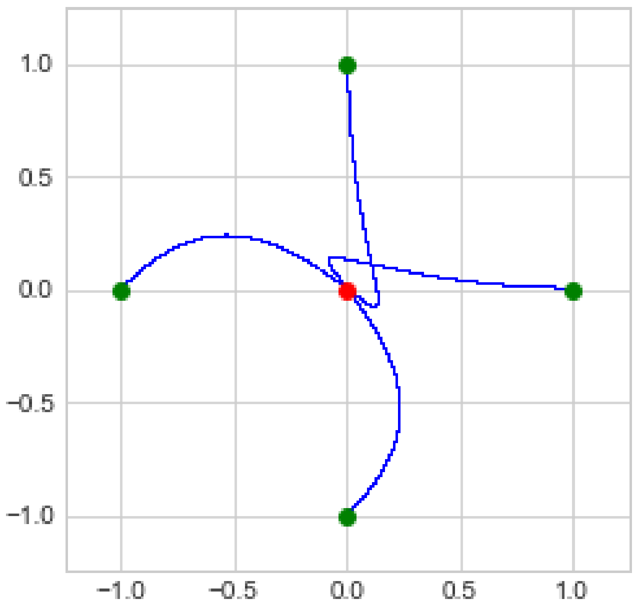
[8]:

```python
points = reset_pts()
pt_coords = convert_to_coords(points)
spi_pa_coords, c = simulate(points, return_center=True)

graph(spi_pa_coords, pt_coords, c_dot=c)
```

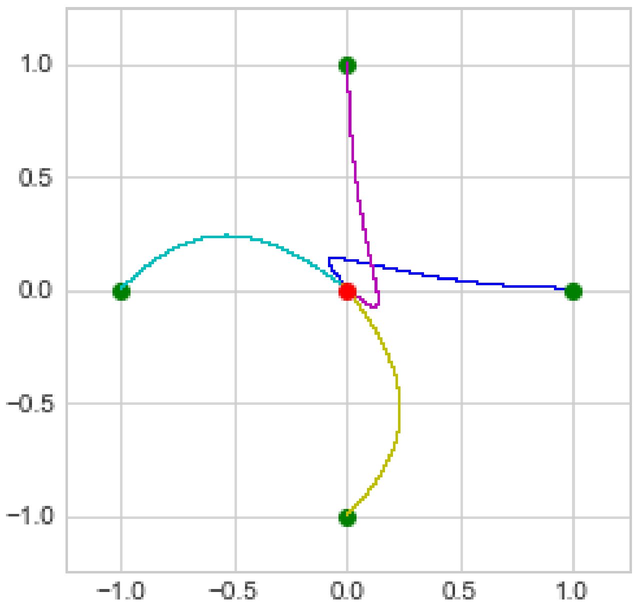


[9]:

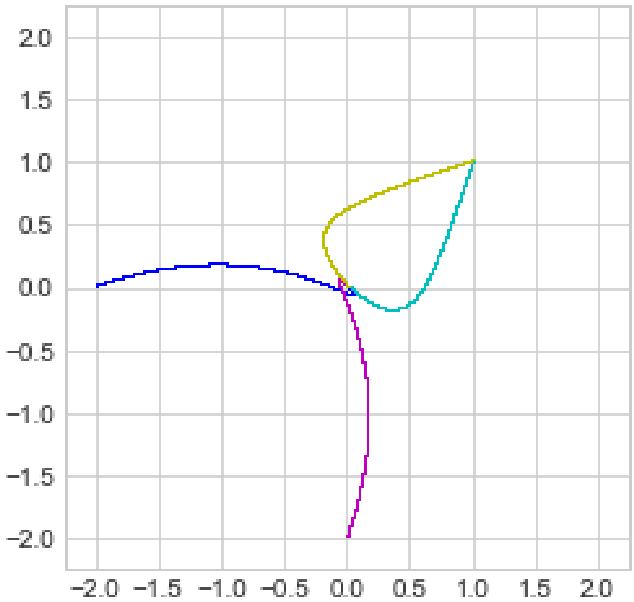
```python
points = reset_pts()
points = reorder(points, [0, 2, 1, 3])
pt_coords = convert_to_coords(points)
pa_coords, c = simulate(points, return_center=True)

graph(pa_coords, pt_coords, c_dot=c)
```



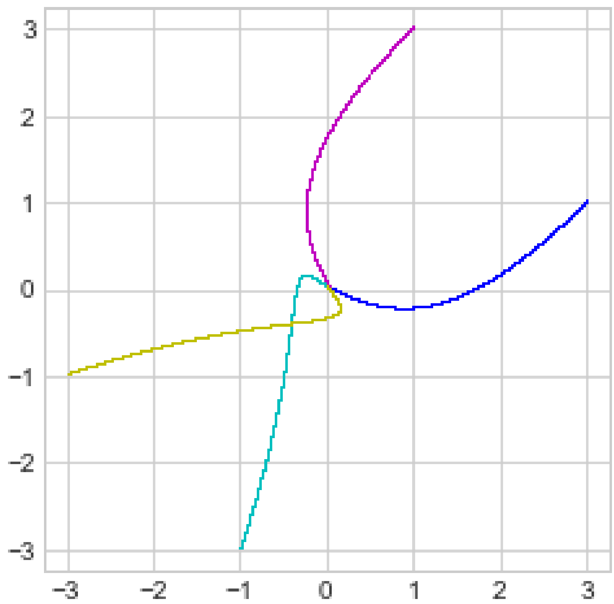
[10]:
```
graph([pa_coords[0]], pt_coords)
graph([pa_coords[1]], color='c')
graph([pa_coords[2]], color='m')
graph([pa_coords[3]], color='y', c_dot=c)
```



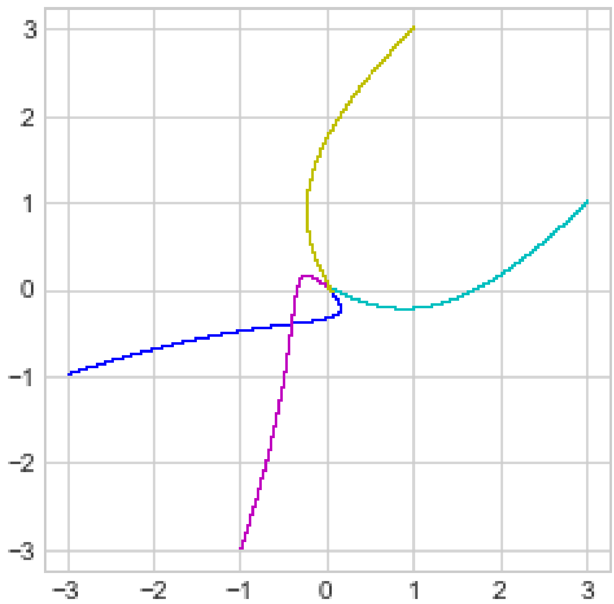
[11]:
```
deriv_coords = derivative(pa_coords)

graph([deriv_coords[0]])
graph([deriv_coords[1]], color='c')
graph([deriv_coords[2]], color='m')
graph([deriv_coords[3]], color='y', axis_lim=[-2.25, 2.25], tick=10)
```



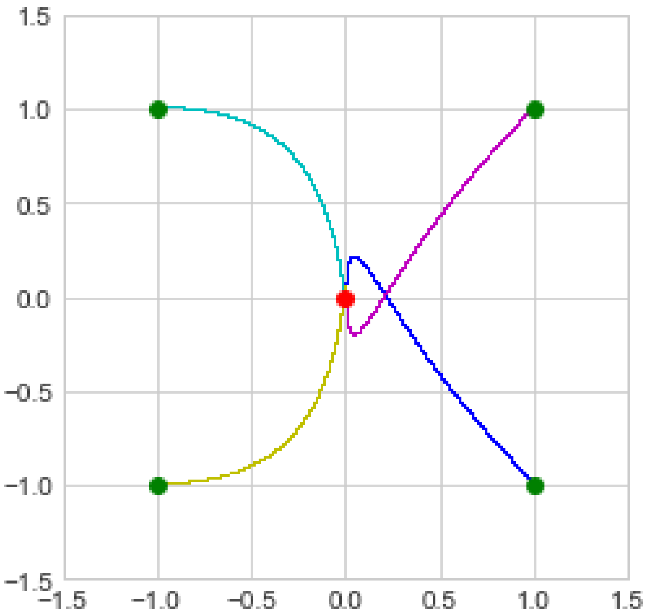
[12]:
```
deriv2_coords = derivative(deriv_coords)

graph([deriv2_coords[0]])
graph([deriv2_coords[1]], color='c')
graph([deriv2_coords[2]], color='m')
graph([deriv2_coords[3]], color='y', axis_lim=[-3.25, 3.25], tick=8)
```



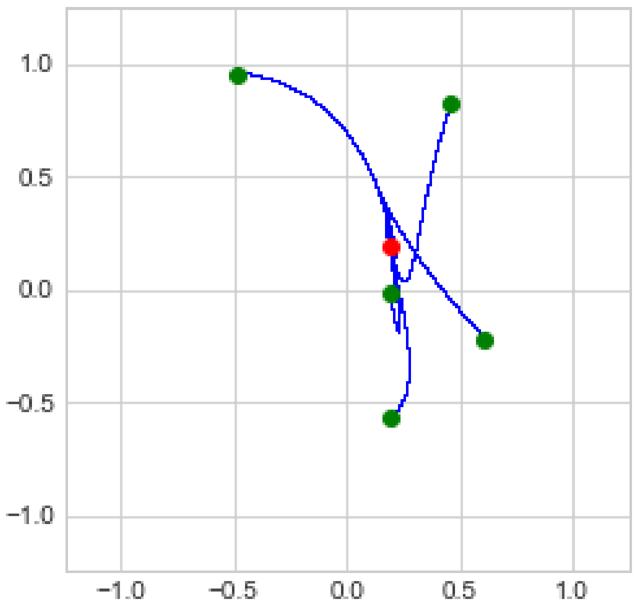
[13]:
```
new_pa_coords = [(
    [-3*x_coord-y_coord for x_coord, y_coord in zip(x, y)],
    [-x_coord-3*y_coord for x_coord, y_coord in zip(x, y)]
) for x, y in pa_coords]

graph([new_pa_coords[0]])
graph([new_pa_coords[1]], color='c')
graph([new_pa_coords[2]], color='m')
graph([new_pa_coords[3]], color='y', axis_lim=[-3.25, 3.25], tick=8)
```



[14]:
```
points = [complex(1, -1), complex(-1, 1), complex(1, 1), complex(-1, -1)]
rot_pt_coords = convert_to_coords(points)
rot_pa_coords, c = simulate(points, return_center=True)

graph([rot_pa_coords[0]])
graph([rot_pa_coords[1]], color='c')
graph([rot_pa_coords[2]], color='m')
graph([rot_pa_coords[3]], rot_pt_coords, color='y', c_dot=c, axis_lim=[-1.5, 1.
↪5])
```



[15]:
```
points = [complex(random.uniform(-1, 1), random.uniform(-1, 1)) for k in␣
 ↪range(5)]
rand_pa_coords, c = simulate(points, return_center=True)
rand_pt_coords = convert_to_coords(points)

graph(rand_pa_coords, rand_pt_coords, c_dot=c)
# image generated here may not the same as the one used in the paper, as it is␣
 ↪random each time
```



0.4 Calculations

0.4.1 Plots of Parametrics

Spiral parametrics:

$$
\begin{aligned}
r &= e^{-t}e^{it}\\
r &= e^{-t}e^{i\left(t+\frac{\pi}{2}\right)}\\
r &= e^{-t}e^{i(t+\pi)}\\
r &= e^{-t}e^{i\left(t+\frac{3\pi}{2}\right)}.
\end{aligned}
$$

In rectangular form:

$$
\begin{aligned}
r &= e^{-t}\cos t + ie^{-t}\sin t)\\
r &= e^{-t}\cos\left(t+\frac{\pi}{2}\right) + ie^{-t}\sin\left(t+\frac{\pi}{2}\right)\\
r &= e^{-t}\cos(t+\pi) + ie^{-t}\sin(t+\pi)\\
r &= e^{-t}\cos\left(t+\frac{3\pi}{2}\right) + ie^{-t}\sin\left(t+\frac{3\pi}{2}\right).
\end{aligned}
$$

Substitute trig functions:

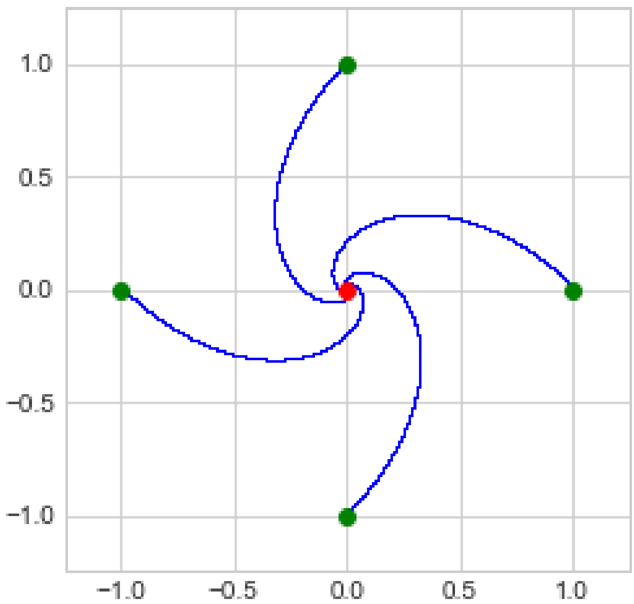
$$r = e^{-t}\cos t + ie^{-t}\sin t$$
$$r = -e^{-t}\sin t + ie^{-t}\cos t$$
$$r = -e^{-t}\cos t - ie^{-t}\sin t$$
$$r = e^{-t}\sin t - ie^{-t}\cos t.$$

[16]:
```
def x_1(t):
    return math.e ** (-t) * math.cos(t)

def y_1(t):
    return math.e ** (-t) * math.sin(t)

def x_2(t):
    return -math.e ** (-t) * math.sin(t)

def y_2(t):
    return math.e ** (-t) * math.cos(t)

def x_3(t):
    return -math.e ** (-t) * math.cos(t)

def y_3(t):
    return -math.e ** (-t) * math.sin(t)

def x_4(t):
    return math.e ** (-t) * math.sin(t)

def y_4(t):
    return -math.e ** (-t) * math.cos(t)
```

Plot parametrics and check discrepancy with simulation

[17]:
```
plot_coords = [
    parametric(x_1, y_1),
    parametric(x_2, y_2),
    parametric(x_3, y_3),
    parametric(x_4, y_4)
]

init_pts = convert_to_coords(reset_pts())

graph(plot_coords, init_pts, c_dot=[0, 0])

print("Maximum Discrepancy: ", max_discrepancy(plot_coords, spi_pa_coords))
```

```
Maximum Discrepancy:  9.999666672212565e-09
```



Switched parametrics:

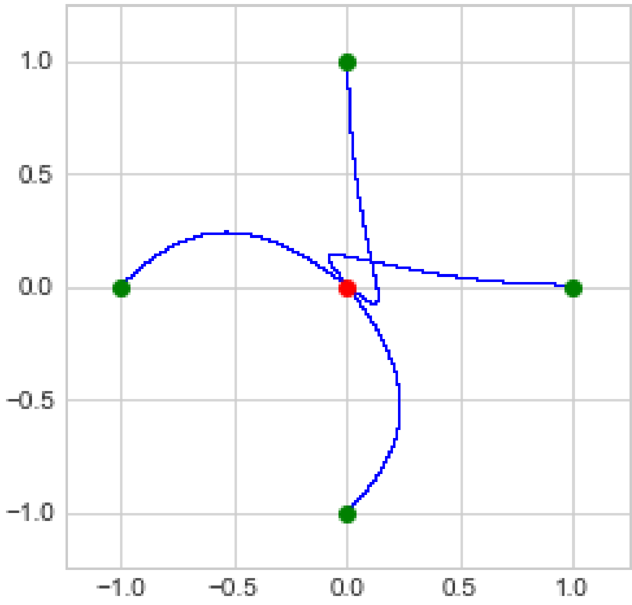
$$
\begin{aligned}
f_1(t) &= \frac{1}{2}e^{-t}\left(e^{-t} + \cos t - \sin t + \left(e^{-t} - \cos t + \sin t\right)i\right)\\
f_2(t) &= \frac{1}{2}e^{-t}\left(-e^{-t} - \cos t - \sin t + \left(-e^{-t} + \cos t + \sin t\right)i\right)\\
f_3(t) &= \frac{1}{2}e^{-t}\left(e^{-t} - \cos t + \sin t + \left(e^{-t} + \cos t - \sin t\right)i\right)\\
f_4(t) &= \frac{1}{2}e^{-t}\left(-e^{-t} + \cos t + \sin t + \left(-e^{-t} - \cos t - \sin t\right)i\right),
\end{aligned}
$$

[18]:
```
def x_1(t):
    return (math.e ** (-t) + math.cos(t) - math.sin(t)) * (math.e ** (-t)) / 2

def y_1(t):
    return (math.e ** (-t) - math.cos(t) + math.sin(t)) * (math.e ** (-t)) / 2

def x_2(t):
    return (-math.e ** (-t) - math.cos(t) - math.sin(t)) * (math.e ** (-t)) / 2

def y_2(t):
    return (-math.e ** (-t) + math.cos(t) + math.sin(t)) * (math.e ** (-t)) / 2

def x_3(t):
    return (math.e ** (-t) - math.cos(t) + math.sin(t)) * (math.e ** (-t)) / 2
```

```python
def y_3(t):
    return (math.e ** (-t) + math.cos(t) - math.sin(t)) * (math.e ** (-t)) / 2

def x_4(t):
    return (-math.e ** (-t) + math.cos(t) + math.sin(t)) * (math.e ** (-t)) / 2

def y_4(t):
    return (-math.e ** (-t) - math.cos(t) - math.sin(t)) * (math.e ** (-t)) / 2
```

[19]:
```python
plot_coords = [
    parametric(x_1, y_1),
    parametric(x_2, y_2),
    parametric(x_3, y_3),
    parametric(x_4, y_4)
]

init_pts = convert_to_coords(reset_pts())

graph(plot_coords, init_pts, c_dot=[0, 0])

print("Maximum Discrepancy: ", max_discrepancy(plot_coords, pa_coords))
```

```
Maximum Discrepancy:  1.5810545038958803e-08
```



Check that the transformed graph matches the second derivative

[20]:
```
reordered_new_pa_coords = reorder(new_pa_coords, [1, 2, 3, 0])

print("Maximum sum of diagonal components: ",␣
↪max_discrepancy(reordered_new_pa_coords, deriv2_coords))
```

```
Maximum sum of diagonal components:  9.449103277614763e-09
```

Verify the diagonal components of $f_1(t)$ and $f_2(t)$ are equal and opposite

[21]:
```
max_sum = 0

for a, b, c, d in zip(pa_coords[0][0], pa_coords[0][1], pa_coords[1][0],␣
↪pa_coords[1][1]):
    s = abs(a + b + c + d)

    if s > max_sum:
        max_sum = s

print("Maximum sum of diagonal components: ", max_sum / math.sqrt(2))
```

```
Maximum sum of diagonal components:  1.785980171752794e-15
```

Matrix exponential for 5 points (will be the same for all systems of 5 points, only the initial condition vector changes)

[ ]:
```
t = sp.symbols('t')
A = sp.Matrix(t*np.array([[-1, 1, 0, 0, 0],[0, -1, 1, 0, 0],[0, 0, -1, 1,␣
↪0],[0, 0, 0, -1, 1],[1, 0, 0, 0, -1]]))
print(sp.simplify(sp.exp(A)))
# Have not run here as output is extremely long
```