1 Euler's Method

Equation of Interest

$$\frac{dy}{dt} + 2y = 2 - e^{-4t}, y(0) = 1$$

a Here I solve the closed-form solution of the equation of interest.

Solve for the general equation

$$\frac{dy}{dt} + 2y = 2 - e^{-4t} \text{ given equation}$$

$$r(x) = f(x)^{\int p(x)dx} \text{ way to solve the first order linear differential equation}$$

let $f(x) = e^x$ and $p(x) = 2$

$$\begin{aligned}
r(x) &= e^{\int 2dt} \text{ plugging into the values} \\
r(x) &= e^{2t} \text{ calculate the integration factor} \\
\frac{dy}{dt}e^{2t} + 2ye^{2t} &= 2e^{2t} - e^{2t}e^{-4t} \text{ plug in integration factor} \\
\frac{d}{dt}(e^{2t}y) &= 2e^{2t} - e^{2t}e^{-4t} \text{ product rule} \\
\frac{d}{dt}(e^{2t}y) &= 2e^{2t} - e^{-2t} \text{ power rules} \\
\int \frac{d}{dt}(e^{2t}y) &= \int (2e^{2t} - e^{-2t})dt \text{ integrate both sides} \\
e^{2t}y &= e^{2t} + \frac{1}{2}e^{-2t} + C \text{ integration} \\
y &= \frac{e^{2t} + \frac{1}{2}e^{-2t} + C}{e^{2t}} \text{ divide both sides} \\
y &= 1 + \frac{1}{2}e^{-2t}e^{-2t} + Ce^{-2t} \text{ simplify} \\
y &= 1 + \frac{1}{2}e^{-4t} + Ce^{-2t} \text{ simplify}
\end{aligned}$$

Use the initial condition

$$\begin{aligned}
y(t) &= 1 + \frac{1}{2}e^{-4t} + Ce^{-2t} \text{ general equation} \\
y(0) &= 1 + \frac{1}{2}e^{-4(0)} + Ce^{-2(0)} \text{ use the initial condition} \\
1 &= 1 + 1 \cdot \frac{1}{2} + C \text{ plugging in the initial conditions} \\
1 &= 1 + \frac{1}{2} + C \text{ simplification} \\
C &= -\frac{1}{2} \text{ solving for } C
\end{aligned}$$

Therefore, the final solution with the given initial condition is $y(t) = 1 + \frac{1}{2}e^{-4t} - \frac{1}{2}e^{-2t}$.

b The next step is to compare the exact solution to an Euler's Method approximation at $t = \{1, 2, 3, 4, 5\}$. In this particular case, I used $h = .001$ which allowed the approximation to match very closely with the exact solution. Figures 1, 4, 3, **??**, and 5 show the results for each t value.

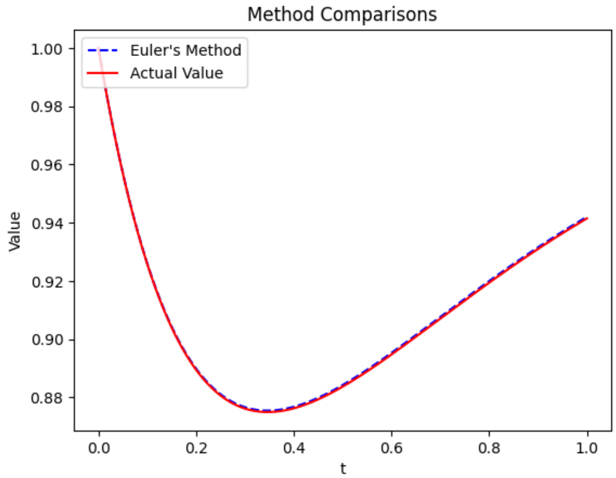

Figure 1: Exact vs Approximation for $t = 1$

The figure displays the exact solution and the approximation on the same plot for $t = 1$. Euler's Method yields a value of .9420 while the actual value yields a value of .9415.

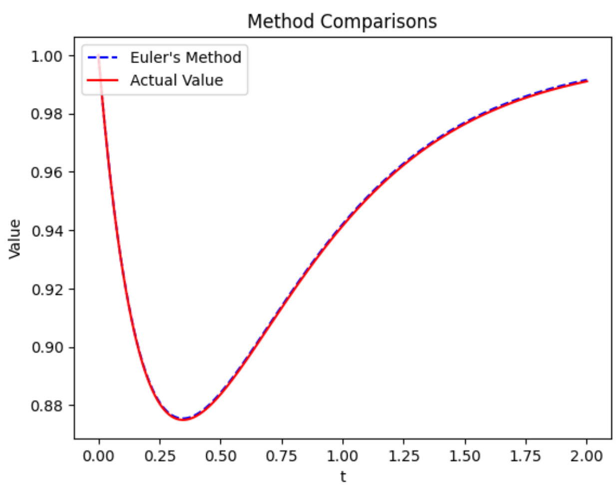

Figure 2: Exact vs Approximation for $t = 2$

The figure displays the exact solution and the approximation on the same plot for $t = 2$. Euler's Method yields a value of .9915 while the actual value yields a value of .9912.

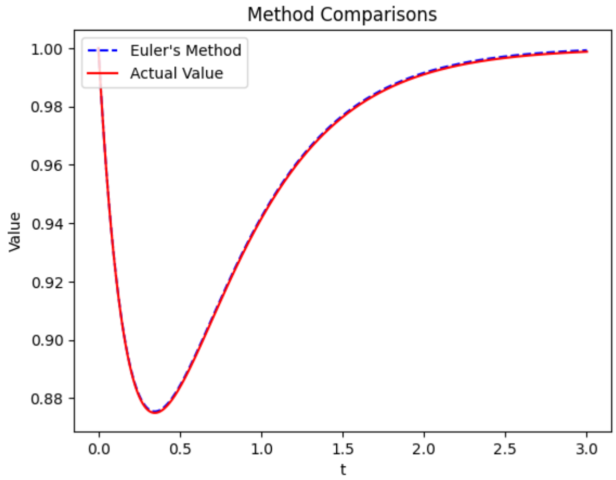

Figure 3: Exact vs Approximation for $t = 3$

The figure displays the exact solution and the approximation on the same plot for $t = 2$. Euler's Method yields a value of .9993 while the actual value yields a value of .9988.

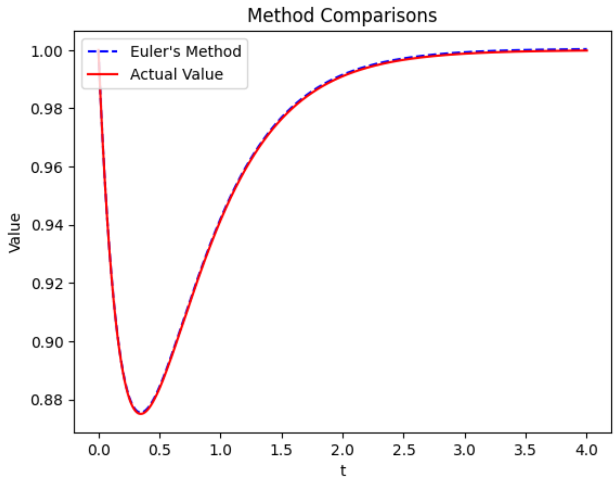

Figure 4: Exact vs Approximation for $t = 4$

The figure displays the exact solution and the approximation on the same plot for $t = 4$. Euler's Method yields a value of 1.0003 while the actual value yields a value of 0.9998.

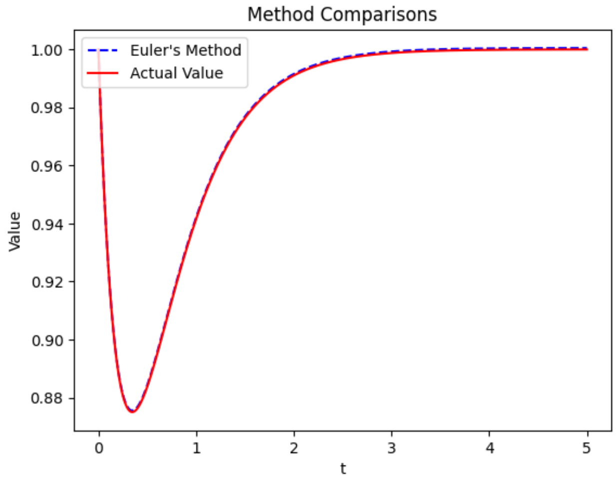

Figure 5: Exact vs Approximation for $t = 5$

The figure displays the exact solution and the approximation on the same plot for $t = 5$. Euler's Method yields a value of 1.0005 while the actual value yields a value of 1.0000.

The approximate vs exact values are provided in the descriptions of the figures for reference.

c To further work with Euler's method, I used different step sizes, $h = \{0.1, 0.05, 0.01, 0.005, 0.001\}$, to solve for each $t = \{1, 2, 3, 4, 5\}$ values. Figures 6, 7, 8, 9, and 10 display the results of a t value with each h value.

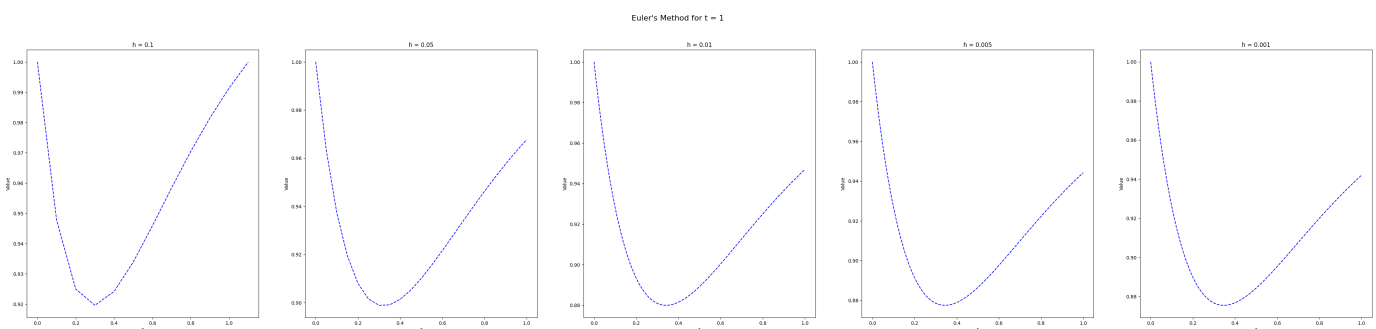

Figure 6: Approximation with all h values for $t = 1$

The figure displays the Euler's approximation for $t = 1$ for all h values. As evident in the figure, the smaller the h value, the smoother the graph looks.

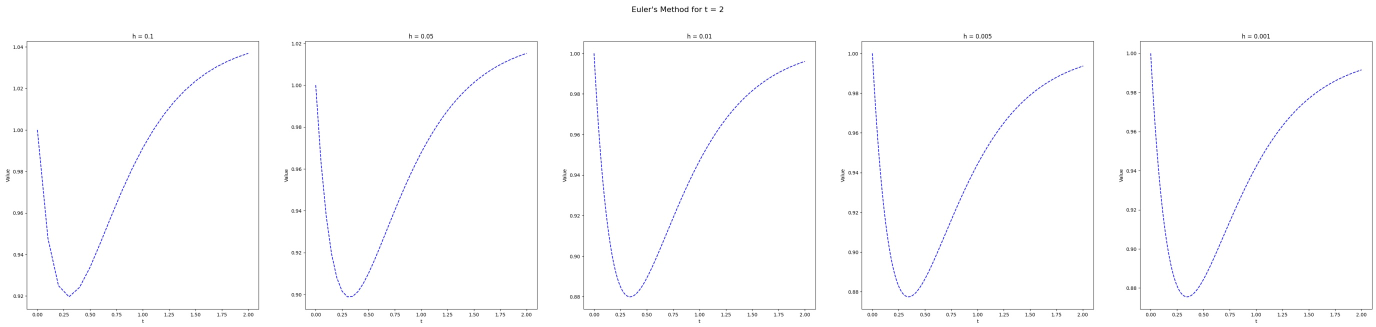

Figure 7: Approximation with all h values for $t = 2$

The figure displays the Euler's approximation for $t = 2$ for all h values. As evident in the figure, the smaller the h value, the smoother the graph looks.

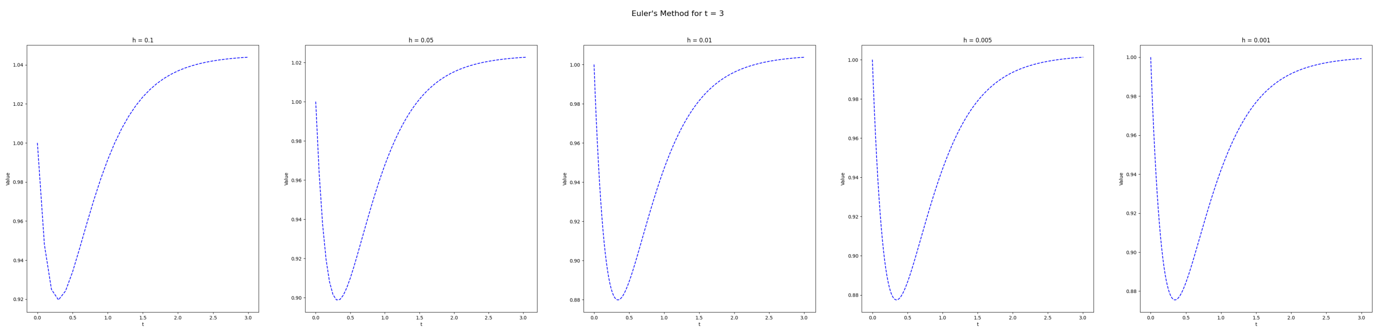

Figure 8: Approximation with all h values for $t = 3$

The figure displays the Euler's approximation for $t = 3$ for all h values. As evident in the figure, the smaller the h value, the smoother the graph looks.

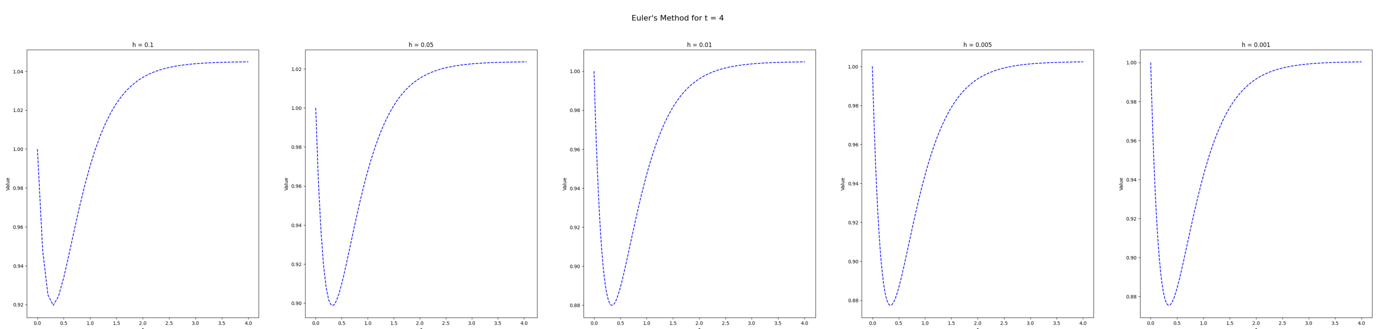

Figure 9: Approximation with all h values for $t = 4$

The figure displays the Euler's approximation for $t = 4$ for all h values. As evident in the figure, the smaller the h value, the smoother the graph looks.

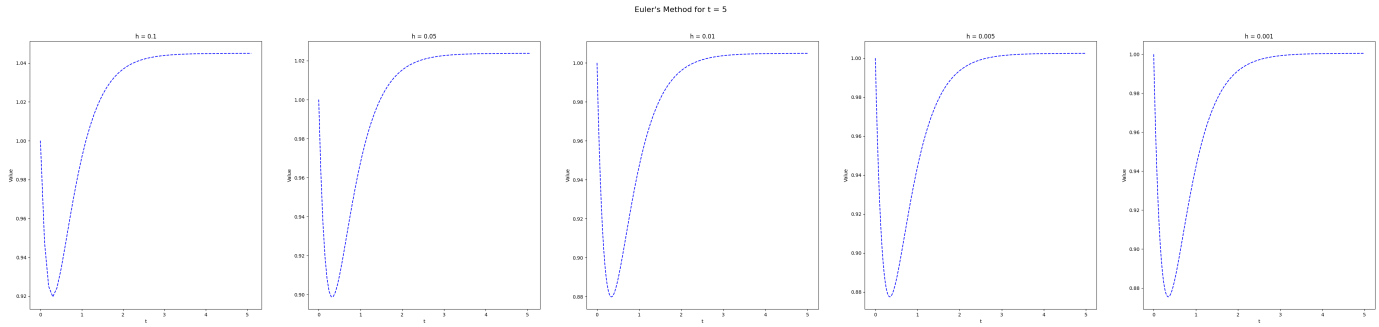

Figure 10: Approximation with all h values for $t = 5$

The figure displays the Euler's approximation for $t = 5$ for all h values. As evident in the figure, the smaller the h value, the smoother the graph looks.

Evidently, a smaller step value h results in a smoother curve but requires more time to calculate. For the upcoming experiment with the geodesic computing, an h value of .01 will be used for both smoothness and computability.

2 Geodesic Shooting

The purpose of this experiment is to generate time-sequential images with the geodesic shooting equations. The geodesic shooting equation $\frac{dv_t}{dt} = -K[(Dv_t)^T \cdot v_t + (Dv_t) \cdot v_t + v_t \cdot div(v_t)]$ and $\frac{d\phi_t}{dt} = v_t \circ \phi_t$ where K refers to a Gaussian smoothing kernel, D is a Jacobian matrix, div is a divergence operator, and $\circ$ refers to bicubic interpolation. Euler's method was used to find the integrals. Given a source image and initial velocity, geodesic shooting was used to compute the target image for $t = 1$ where $h = 0.1$. The final velocity was reported in the code and the transformation of the images are displayed below. Figures 11, 12, and 13 display the original image, the deformations from $t = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and final velocity respectively. The running time for the entire algorithm was reported to be $2.25s$.

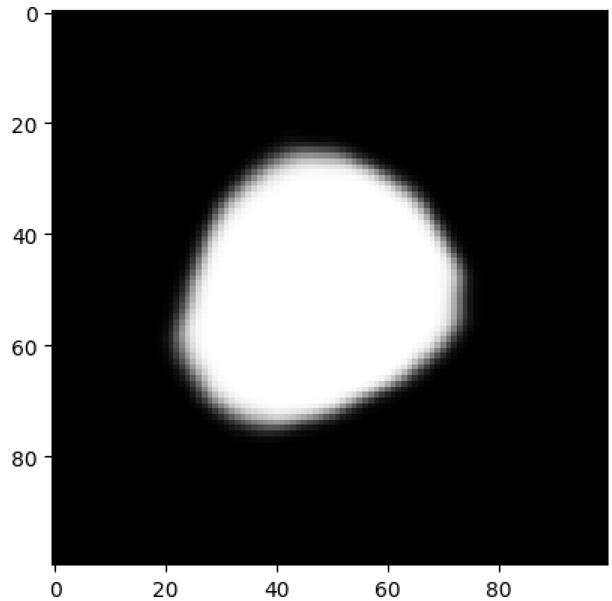

Figure 11: Source Image

The figure displays the original source image for the geodesic shooting.

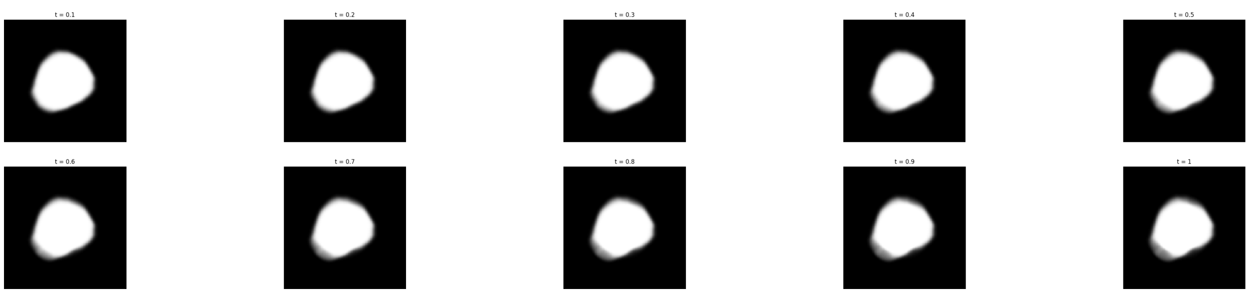

Figure 12: Image Deformation for $t = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

The figure displays the image deformation for $t = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The initial source to $t = .1$ has a noticeable difference. The rest are relatively indistinguishable. However, there is a calculated difference among all the images.

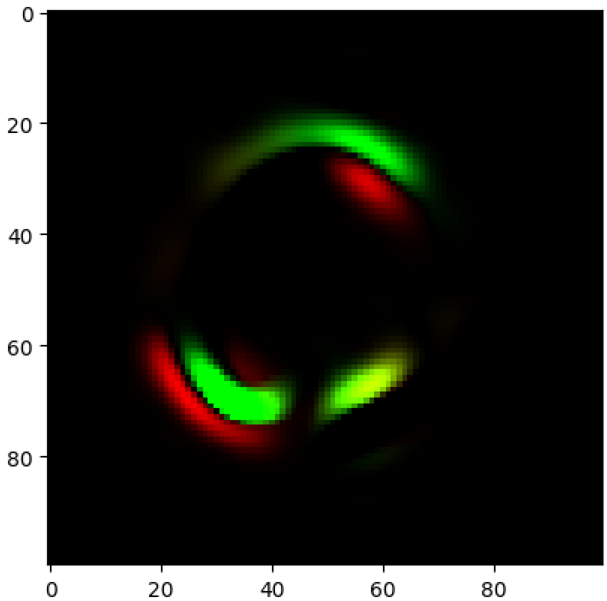

Figure 13: Final Velocity

The figure displays the final velocity as an RGB image.

The same process was applied to an initialized velocity field using a Gaussian distribution with $\sigma = \{2.0, 4.0, 8.0\}$. Figures 14, 15, 18, 19, **??**, **??** display the results for $\sigma = \{2.0, 4.0, 8.0\}$ respectively.

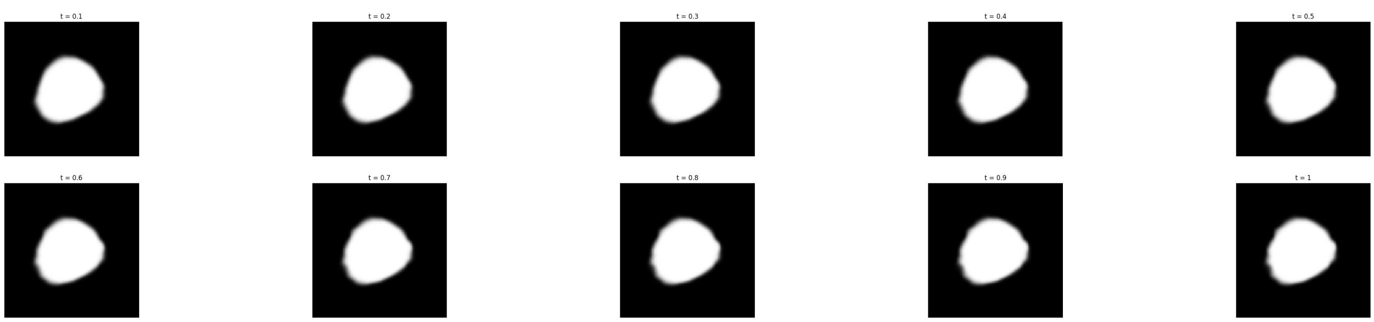

Figure 14: Image deformation for $\sigma = 2.0$

The figure displays the image deformation for all t values given $\sigma = 2.0$.

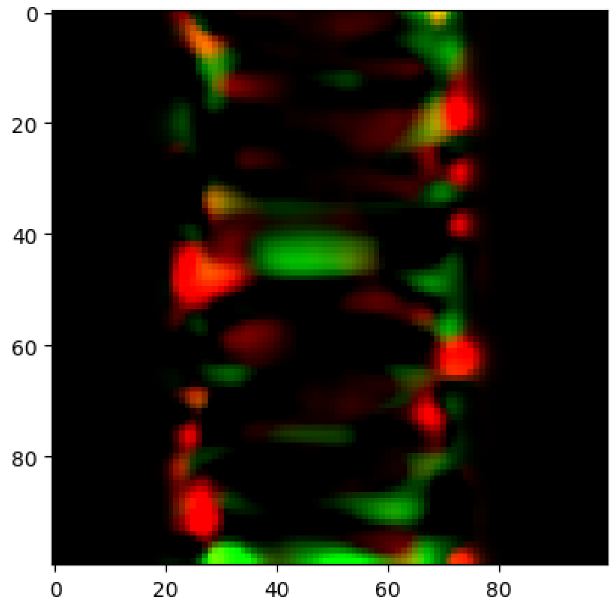

Figure 15: Velocity $\sigma = 2.0$

The figure displays the final velocity for $\sigma = 2.0$.

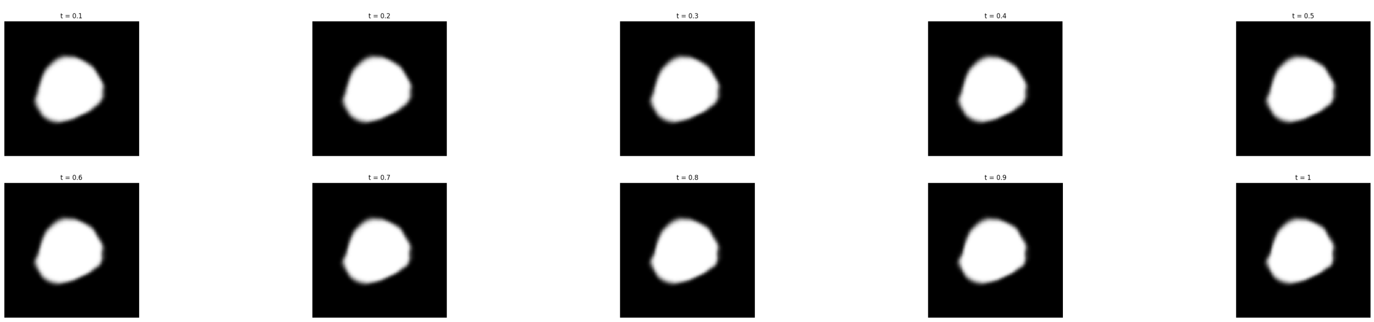

Figure 16: Image deformation for $\sigma = 4.0$

The figure displays the image deformation for all t values given $\sigma = 4.0$.

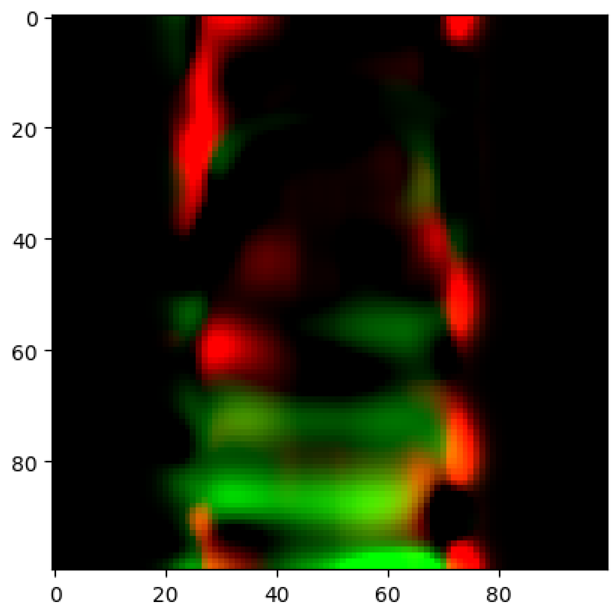

Figure 17: Velocity with $\sigma = 4.0$

The final velocity for $\sigma = 4.0$.

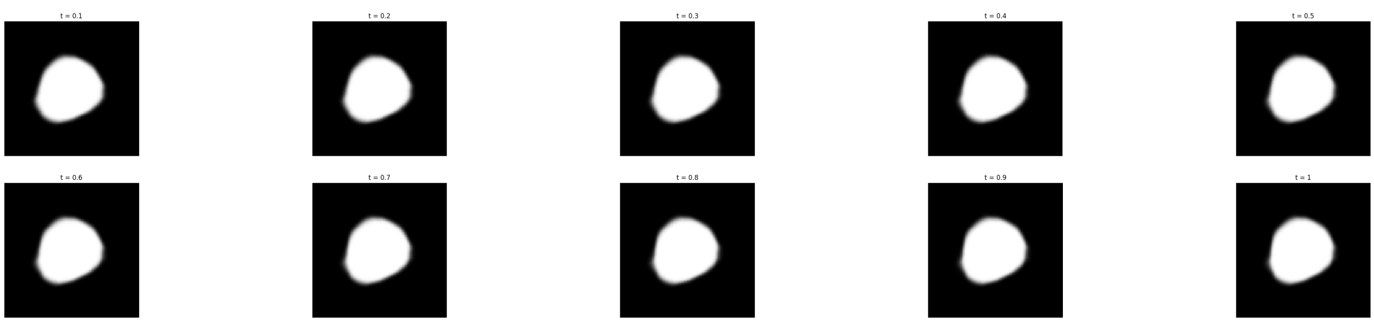

Figure 18: Image deformation for $\sigma = 8.0$

The figure displays the image deformation for all t values given $\sigma = 8.0$.

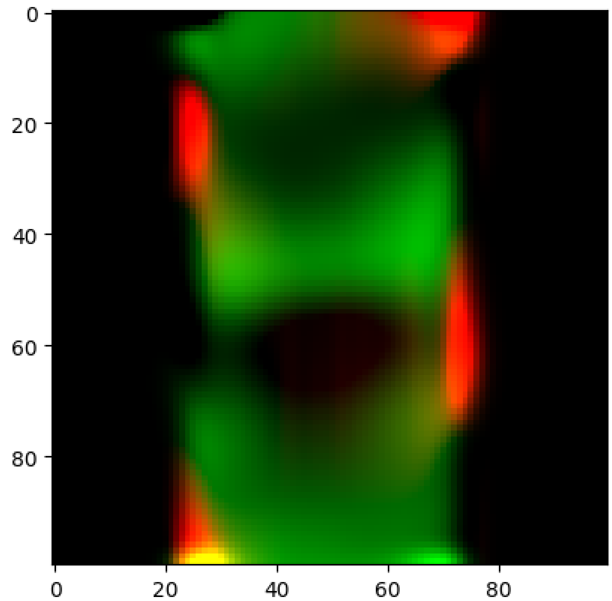

Figure 19: Velocity with $\sigma = 8.0$

The final velocity for $\sigma = 8.0$.

3 Inverse Transformation

The inverse transformation was calculated using the equations $\frac{dv_t}{dt} = K[(Dv_t)^T \cdot v_t + (Dv_t) \cdot v_t + v_t \cdot div(v_t)]$ and $\frac{d\phi_t^{-1}}{dt} = -D\phi_t^{-1} \cdot v_t$. The final velocity is reported in the code and Figure 20 and 21 displays the deformed image using the inverse function and the velocity function.

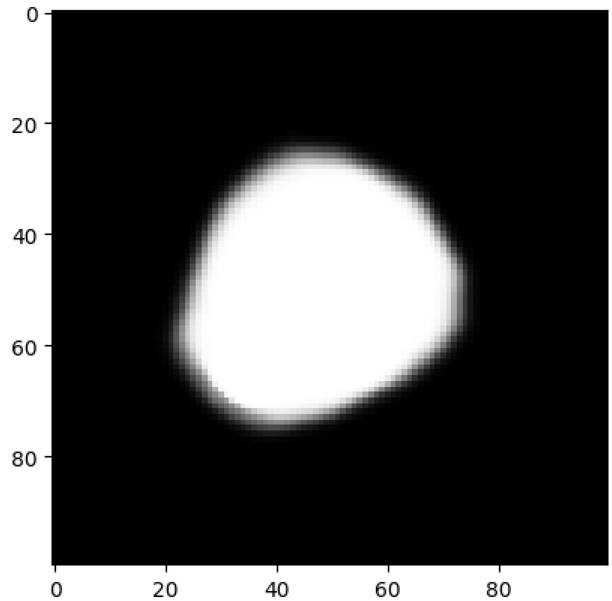

Figure 20: Inverse Image

The inverse image is displayed above.

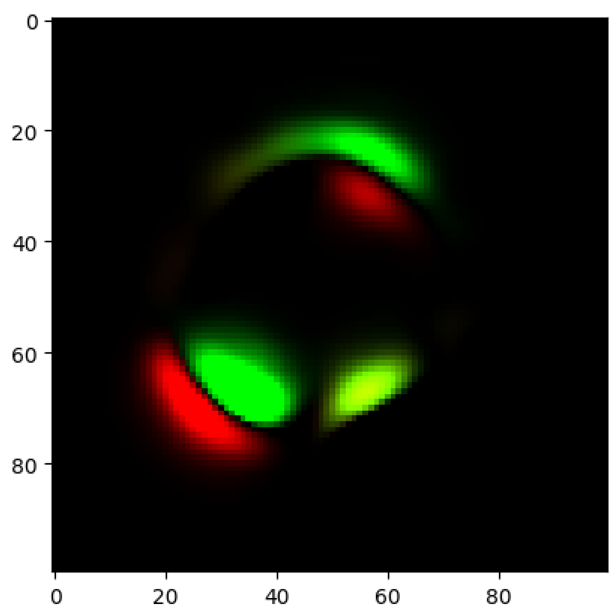

Figure 21: Velocity Inverse Image

The velocity inverse image is displayed above.

The deformed image with the inverse function looks more like the original source image which makes sense since the computation is for the inverse function. In other words, it looks like the reverted form of the final image in Problem 2. The process is almost exactly the same because regardless if it's ϕ or ϕ^{-1}, the bijective mapping still applies. There are a lot of things I learned from these experiments. I definitely have a better understanding on how LDDMM operates in deforming an image and the role of ϕ and the velocity in the process.

A couple things I can do to improve the results include using a smaller h step for a smoother function and perhaps try different smoothing approaches.

4 Sources

https://numpy.org/doc/

https://stackoverflow.com/questions/7370801/how-do-i-measure-elapsed-time-in-python

https://www.geeksforgeeks.org/how-to-display-multiple-images-in-one-figure-correctly-in-matplotlib/