PCA and Shape Mode Analysis

Qihan Guan, **Lucas Wong**

Date: Jul 31, 2025

Introduction

The objective of this project is to describe the flagellar movement patterns by extracting and using their the dominant spatial patterns of bending (shape modes) via principal component analysis (PCA). We aim to reproduce key results including the kymograph of tangent angles (Fig 2B), the two dominant shape modes (Fig 2E), and the limit cycle in shape space (Fig 2F), demonstrating that the beating flagellum operates as a low-dimensional oscillator.

Part (1): Visualization of Flagellum Shape over Time

To begin, we visualize how the **raw** flagellum's shape changes over each time step using the dataset provided in the **"fluidE.mat"** file. This dataset includes:

- **XX and YY**: Matrices containing **x-** and **y-coordinates** for each spatial point along the flagellum at each frame. These matrices fully describe the flagellum's shape.
- **space_scale**: The physical distance between adjacent tracked points along the flagellum (in micrometers, μm).
- **time_scale**: The elapsed time between consecutive frames (in seconds).

We first extract and display essential dataset dimensions using MATLAB:

```
disp('Data:');
disp(['Number of spatial points: ', num2str(m)]);
disp(['Number of time points: ', num2str(n)]);
disp(['Space scale (μm): ', num2str(space_scale)]);
disp(['Time scale (seconds): ', num2str(time_scale)]);
```

The outputs are:

```
Data:
Number of spatial points: 32
Number of time points: 80
Space scale (μm): 0.2
Time scale (seconds): 0.0016502
```

Visualizing the First 5 Time Steps

We visualize the first 5 time step:

```
figure; clf;
for i = 1:5
    plot(XX(:,i), YY(:,i));
    hold on;
end
hold off;
axis equal; % ensures equal scaling for x and y axes
grid on;
xlabel('X (μm)');
ylabel('Y (μm)');
title('First 5 Time Steps of Flagellar Movement');
```

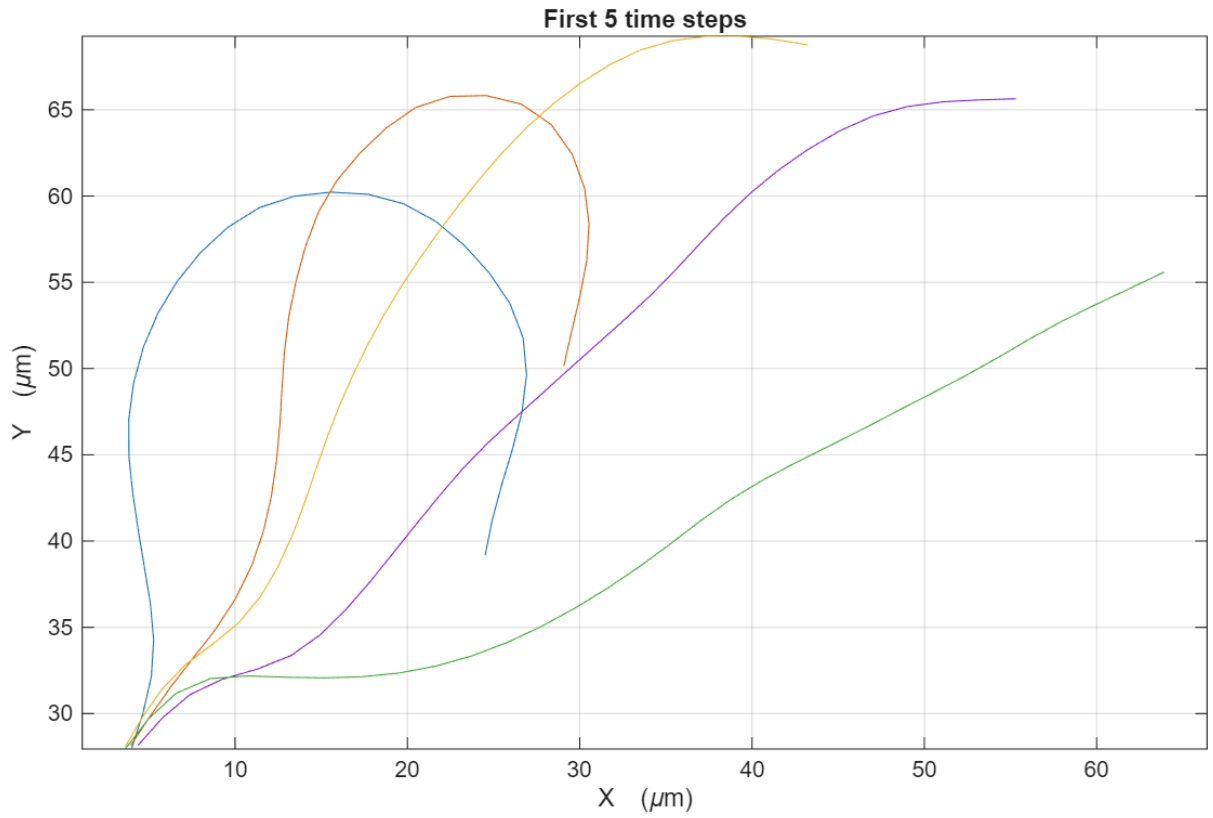

Figure 1A

The figure illustrates sequential snapshots of the flagellar shape changes at 5 successive time intervals, showing how the flagellum bends over approximately 6.6 milliseconds. Calculated by:

```
disp(['Duration of the first 5 time steps: ', num2str(4 * time_scale), ' seconds']);
```

This outputs:

```
Duration of the first 5 time steps: 0.0066007 seconds
```

Visualizing All 80 Time Steps

To fully see the flagellum's motion over the entire observation period, we plot all 80 time step.

```
figure; clf;
for i = 1:n
    plot(XX(:,i), YY(:,i));
    hold on;
end
hold off;
axis equal; % maintains equal axis scaling
grid on;
xlabel('X (μm)');
ylabel('Y (μm)');
title('All Flagellar Shapes Over Time');
```

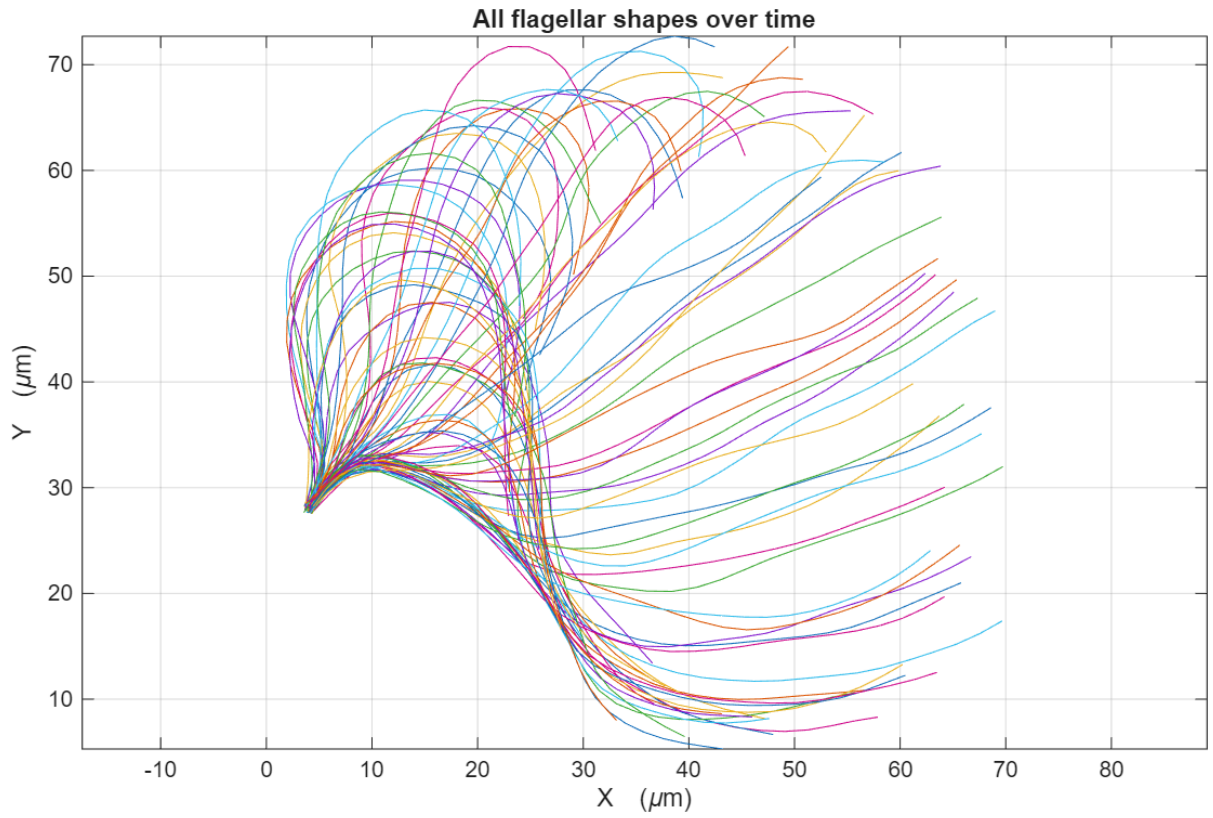

Figure 2 This visualization provides a complete temporal representation of flagellar dynamics:

Explanation of Coordinate Pairing and MATLAB Plotting:

For each time step t_i, the dataset provides two vectors of length m:

$$XX(:,i) = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix}, \quad YY(:,i) = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{bmatrix}$$

The coordinate of the (j)th spatial point is obtained by pairing:

$$(X_j, Y_j) \leftrightarrow (XX(j,i), YY(j,i))$$

In MATLAB, plotting these pairs automatically connects consecutive points, forming the complete shape at each time step:

```
plot(XX(:,i), YY(:,i));
```

Thus, to plot the initial 5 shapes or all 80 shapes, we iterate through the corresponding t_i as shown in the MATLAB code examples above.

Part (2): Tangent-Angles matrix and Kymograph

Next, we convert flagellum shapes into tangent angles at each time step, representing the local direction of the flagellum along its length, and use these angles to create a kymograph.

Matlab Computation of Tangent angles

For each time step t_i we are given m points along the flagellum $\{(X_1, Y_1),\ (X_2, Y_2), \ldots, (X_m, Y_m)\}$

We first calculate the slope of all adjacent points:

$$\frac{\Delta y_j(t_i)}{\Delta x_j(t_i)} = \frac{y_{j+1}(t_i) - y_j(t_i)}{x_{j+1}(t_i) - x_j(t_i)}$$

.

Then convert them into angle using `atan2` which give us the tangent angle $\phi_j(t_i)$.

$$\phi_j(t_i) = \operatorname{atan2}\big(\Delta y_j, \Delta x_j\big) \in (-\pi, \pi]. \ \equiv \ arctan\left(\frac{\Delta y_j(t_i)}{\Delta x_j(t_i)}\right)$$

We repeat this process for every time step t_i for $i = 1, .., 80$, and build the tangent-angle matrix $\Phi^{(m-1)\times n}$. With each column represent Vector of the tangents angles of that time step t_i:

$$\left[\phi_1(t_i),\ \phi_2(t_i),\ \ldots,\ \phi_{j=m-1}(t_i)\right]^\top$$

Full matrix

$$\Phi = \begin{bmatrix} \phi_1(t_1) & \phi_1(t_2) & \cdots & \phi_1(t_n) \\ \phi_2(t_1) & \phi_2(t_2) & \cdots & \phi_2(t_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m-1}(t_1) & \phi_{m-1}(t_2) & \cdots & \phi_{m-1}(t_n) \end{bmatrix},$$

with enties

$$\Phi_{j,i} = \phi_j(t_i).$$

Because $\phi_j \in (-\pi, \pi]$, we use `unwrap` on tangent angles function to remove "jump" by $\pm 2\pi$.

Creating the kymograph

We define physical axes for the kymograph:

- **X-axis:** Spatial (arclength) axis:

$$s_j = (j-1) \cdot \text{space\_scale}$$

- **Y-axis:** Temporal axis:

$$t_i = (i-1) \cdot \text{time\_scale}$$

This tell us where each tangent angle measurement occurred(in micrometers) along the length of the flagellum, and when.

This whole process is done in matlab as shown below:

```
phi = zeros(m-1, n);
for i = 1:80
    dx = diff(XX(:,i));  % [x2-x1, ..., x32-x31] at time i
    dy = diff(YY(:,i));
    phi(:,i) = atan2(dy, dx);  % Store column i
end

phi_unwrapped = zeros(size(phi));
for i = 1:80
    phi_unwrapped(:,i) = unwrap(phi(:,i));
end

delta_s = space_scale; % Distance between two adjacent points
delta_t = time_scale;  % Time between frames

s = (0:size(phi_unwrapped,1)-1)' * delta_s; % Position where phi is defined.
t = (0:n-1)' * delta_t; % When phi is calculated
```

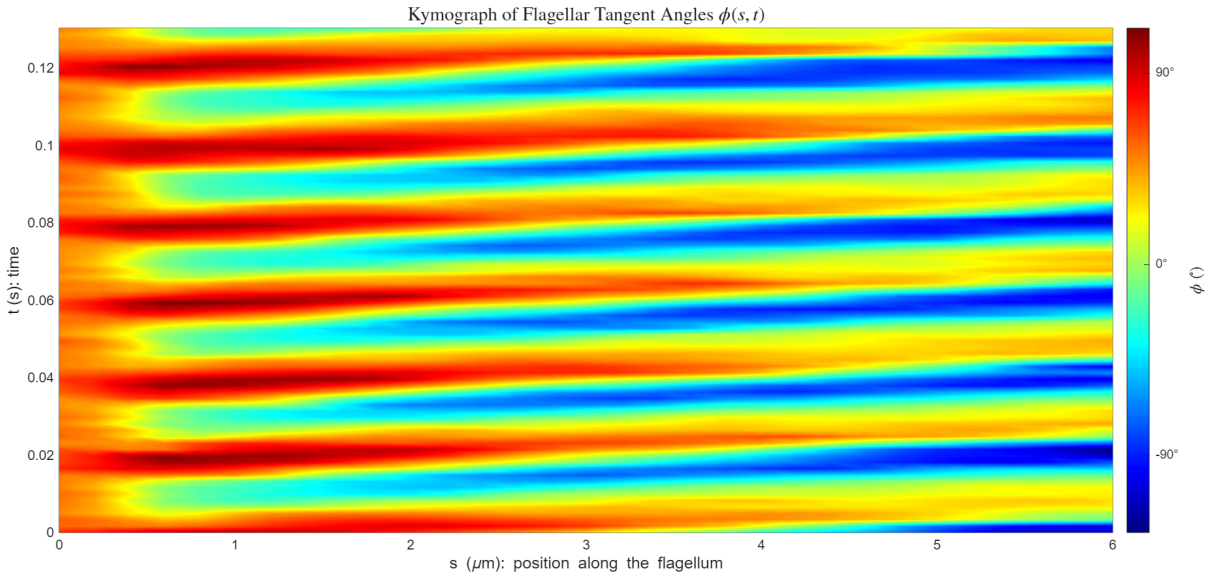

Figure 3

Intepretation of kymograph

The **x-axis** is the position along the flagellum, The leftmost $(s = 0)$ is the base, where the flagellum attaches to the cell. And the rightmost $(s = 6)$ corresponds to the very tip of the flagellum. The **color bar** indicates the bending angle $\phi(s, t)$ in degrees

If we focus at the far left section of graph($s \lesssim 0.4$), we can observe the $\mathrm{orange} \leftrightarrow \mathrm{red}$ repeating pattern as time goes forward, $t = 0,\ 0.02\ , \ldots,\ 0.12, \ldots$
It only oscillating between two positive-bend values and never passing through zero (green) or negative (blue). Meanwhile the mid section ($s \approx 3$) shows full oscillation: from positive, through zero, and to negative and back in a periodic cycle. This indicates that most of the flagellar motion occurs away from the base.

The striped, repeating bands in the plot reflect the periodic nature of the flagellar motion. And we can observe that the flagellum completes one full oscillation (or beat) approximately every **0.02 seconds**. Since each time step is $\Delta t = 0.0016502$ seconds apart, we can calculate the number of time steps per beat as:

$$\text{time steps per beat} = \frac{T}{\Delta t} \approx \frac{0.02}{0.0016502} \approx 12.$$

Thus, one beat spans about **12 consecutive time steps** in the data.

Part (3): SVD for shape Mode Analysis

While the tangent-angle matrix Φ tells you the local bend angle for every position alongs the flagellum at every t_i. the matrix is high dimensional with $m-1=40$ spatial points and $n=80$ time points. Thus we need to compress the Φ matrix by performing SVD, and use it to find the dominant spatial patterns of bending (shape modes) that explain the most variance in the data.

Matlab implementation of SVD

First we compute `phi_mean` ($\bar{\phi}_j$), the average of the tangent angle at each spatial points across all n time steps.

$$\bar{\phi}_j = \frac{1}{n}\sum_{i=1}^{n}\Phi_{j,i}, \quad j = 1,\ldots,m-1.$$

Then we subtract it from every column of the data matrix to get the demeaned matrix `demean_phi` (Φ')

$$\Phi' = \Phi_{j,i} - \bar{\phi}_j \qquad \Phi' \in \mathbb{R}^{(m-1)\times n}.$$

Next we perform **SVD** on Φ'.

$$\Phi' = U\,S\,V^{\top},$$

Where

- $U \in \mathbb{R}^{(m-1)\times(m-1)}$: Columns representing **shape modes** $\mathbf{U}_k$
- $\Sigma \in \mathbb{R}^{(m-1)\times n}$: Diagonal matrix of **singular values** $\sigma_k = s_k \geq 0$
- $V \in \mathbb{R}^{n\times n}$: Coumns representing **temporal modes** $\mathbf{V}_k$

```
phi_mean = mean(phi_unwrapped, 2);
demean_phi = phi_unwrapped - phi_mean;
[U,S,V] = svd(demean_phi, 'econ');
```

Then we proceed to these 3 matrices to plot the following figures:

The first 4 shape modes

```
figure; clf; hold on;
for k = 1:4
    plot(U(:,k))
end
title('First 4 Shape modes');
hold off;
```

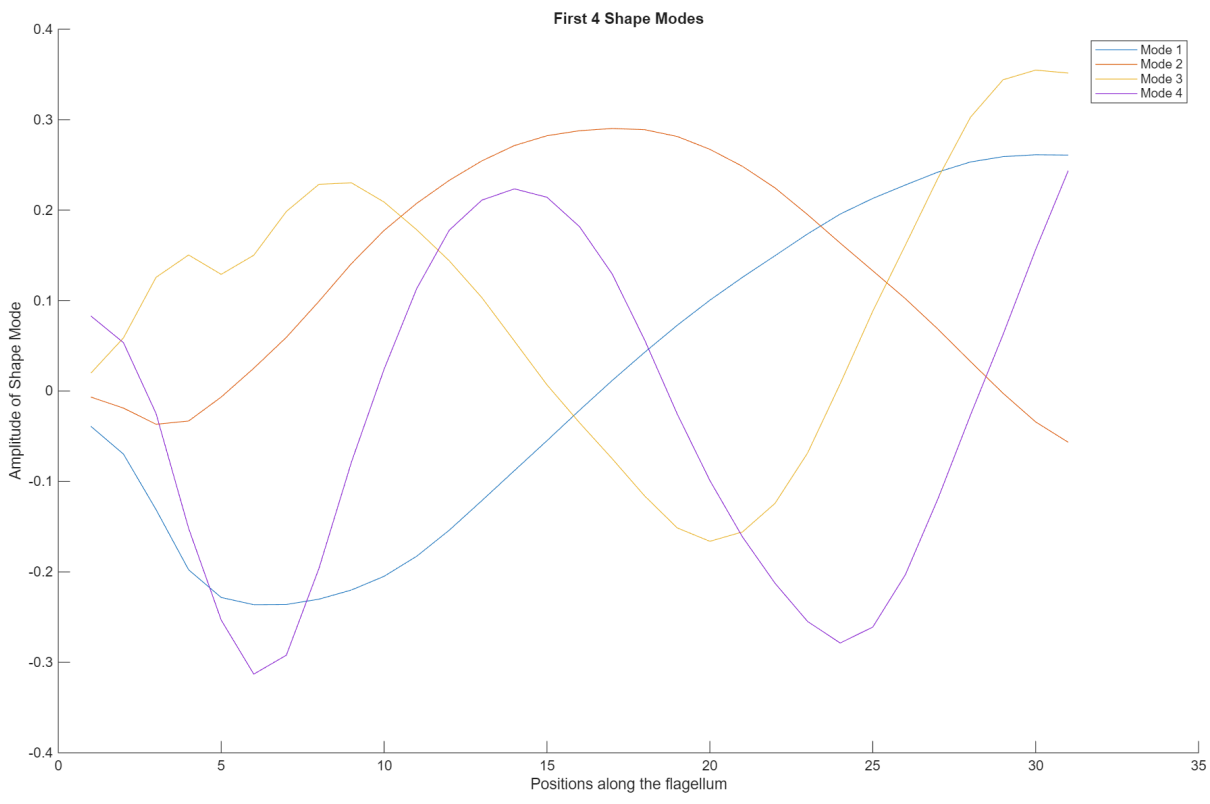

Figure 4

Each lines(shape modes) correspond to a singular from U, with the **X-axis** correspond to the spatial position along the flagellum, from base (j=1) to tip (j=m-1). And **Y-axis**: The bending amplitude (in radians) of mode point j.

The first 4 temporal modes

```
figure; clf; hold on;
for k = 1:4
    plot(V(:,k))
end
title('First 4 Temporal Modes');
hold off;
```

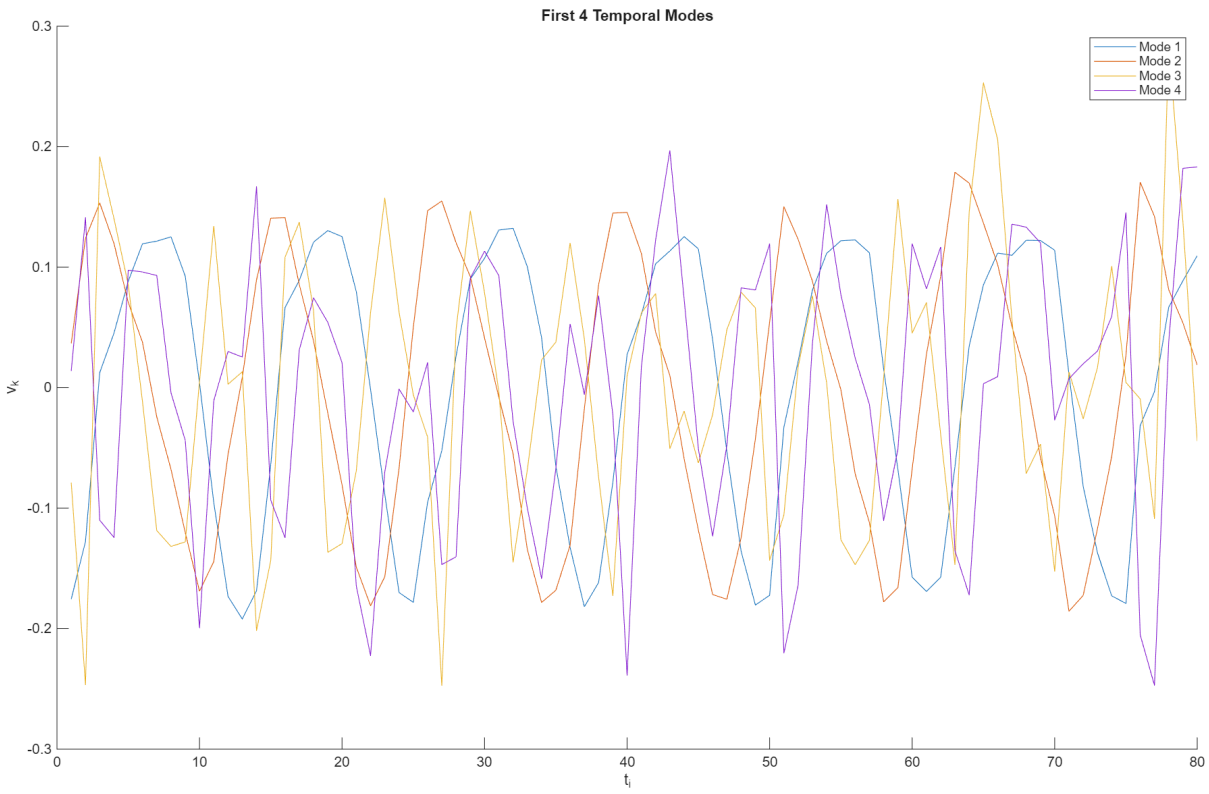

Figure 5

The **X-axis** correspond to time t_i while the **Y-axis** represent mode coefficient V with unit length.

The strength of the modes (Singular values): Variance Retained vs. Number of Modes

We compute the cumulative fraction of total variance caputured by the first k modes:

$$\frac{\sum_{k=1}^{K} s_k^2}{\sum_{k=1}^{r} s_k^2} \qquad \text{for k} = 1,....,\text{m -1}$$

Where $s_k^2 = \sigma_k^2$ is the squared singular values.

```
s = diag(S);
figure;
plot(cumsum(s.^2)./sum(s.^2), 'bo', 'MarkerFaceColor','b','MarkerSize',6)
xlabel("kth Singular Value");
ylabel("% Information Retained");
title("Strength of Singular Values")
```

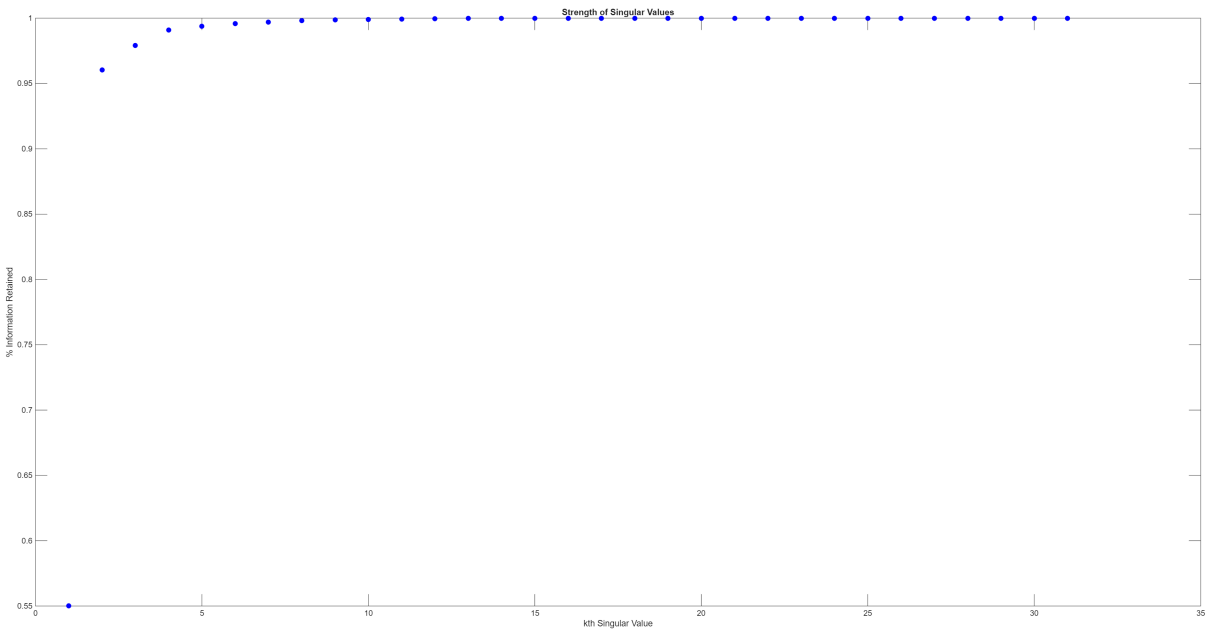

Figure 6

Observe that this figure show that the first two modes U_1, U_2 captures most of the flagellum's shape variance. If we run the following matlab code:

```
retained =cumsum(s.^2)./sum(s.^2);
fprintf('Two-mode variance retained: %.2f%%\n', retained(2)*100);
```

It would tell us the two modes caputures 96.2% of variance.

> Two-mode variance retained: 96.03%

Part (4): Fourier Series Fit for Shape Space

Using the shape and temporal modes that we found in part (3), we now want to find the vectors V_1 and V_2 that represent the first two dimensions of the shape space. V_i can be found by multiplying the i th singular value by the i th temporal mode, or

$$V_i = \sigma_i \cdot \mathbf{v}_i$$

as the V matrix found in part (3) represents the temporal modes. We then want to plot V_1 against V_2, which gives

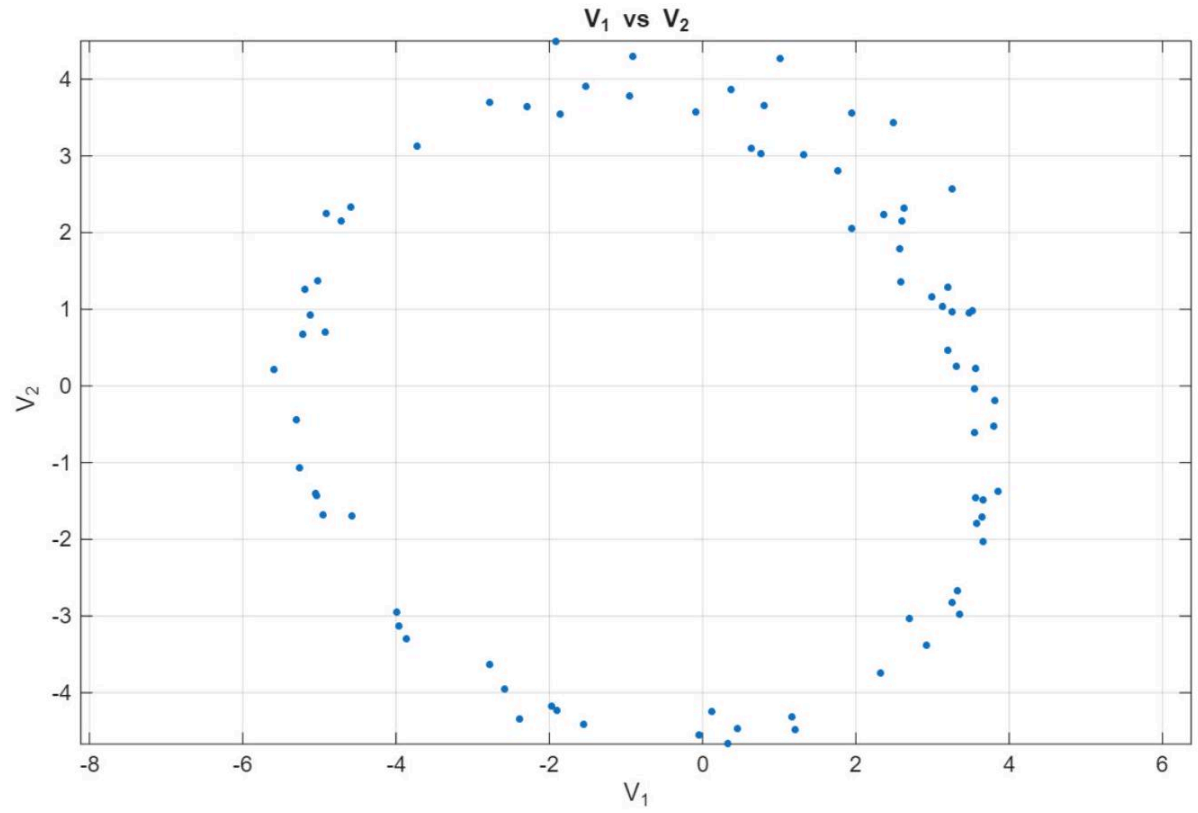

Figure 7B

It can be observed that these points form a closed loop. We want to find the best fitting low dimensional Fourier series $A\cos(\theta) + B\sin(\theta)$ to describe the loop. To do this, we want to find the angle (θ) between the V_1 and V_2. This can be done with the MATLAB code

```
theta = unwrap(atan2(V_2, V_1));
```

Now, we can use least squares to fit $V_i = A_i\cos(\theta) + B_i\sin(\theta)$ to find A and B for both V_1 and V_2. To perform this, we want to solve the least squares equation:

$$A^\top A\mathbf{x} = A^\top B$$

where $B = V_i$, $A = \left[\overrightarrow{\cos(\theta)} \quad \overrightarrow{\sin(\theta)}\right]$, and $\vec{x} = [\, A_i \quad B_i \,]^T$. Doing these calculations in MATLAB gives

$$V_1 \approx 4.2925\cos(\theta) - 0.1475\sin(\theta)$$
$$V_2 \approx -0.1124\cos(\theta) + 4.2011\sin(\theta)$$

Plotting these best fit series against the data gives

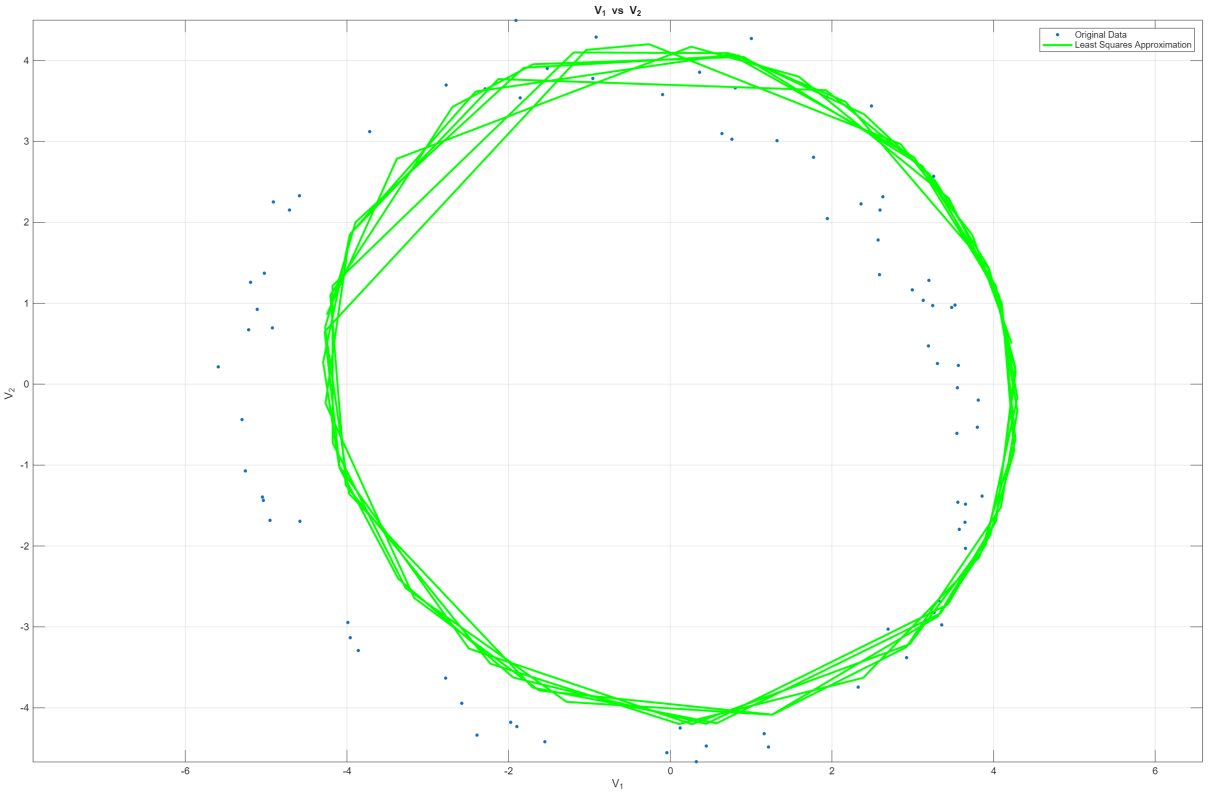

Figure 7A

We will refer to these fitted shape scores as $V_{1,\text{fit}}$ and $V_{2,\text{fit}}$ when used in the reconstruction in Part (5)

Part (5): Reconstruction of the Tangent-Angle Matrix and Centerline

We reconstruct the tangent-angle matrix by linearly combining the first two shape modes U_1, U_2 with the fitted shape scores $V_{1,\text{fit}}, V_{2,\text{fit}}$ obtained in **Part (4)**, plus the spatial mean:

$$\Phi_{j,i}^{\text{recon}} = \bar{\phi}_j + V_{1,\text{fit}}(i)\, U_{1,j} + V_{2,\text{fit}}(i)\, U_{2,j}.$$

MATLAB Code for Tangent-Angle Reconstruction:

```
phi_recon = zeros(size(phi_unwrapped));
for i = 1:size(phi_unwrapped, 2)
    phi_recon(:,i) = phi_mean + V_1_fit(i) * U(:,1) + V_2_fit(i) * U(:,2);
end
```

To reconstruct (x, y) coordinates of the flagellar centerline, For each time step t_i we initialize:

$$x_{1,i} = 0, \quad y_{1,i} = 0$$

And then for $j = 2, \ldots, m$, we do:

$$x_j = x_{j-1} + \Delta s \cdot \cos\left(\Phi_{j-1,i}^{\text{recon}}\right),$$
$$y_j = y_{j-1} + \Delta s \cdot \sin\left(\Phi_{j-1,i}^{\text{recon}}\right),$$

Where $\Delta s = \text{space\_scale}$.

MATLAB Code for Centerline Reconstruction:

```
x_recon = zeros(m, 1);
    y_recon = zeros(m, 1);
    for j = 2:m
        x_recon(j) = x_recon(j-1) + space_scale * cos(phi_recon(j-1, i));
        y_recon(j) = y_recon(j-1) + space_scale * sin(phi_recon(j-1, i));
    end
```

Plotting the reconstructed shapes compared to the original data confirms strong accuracy:

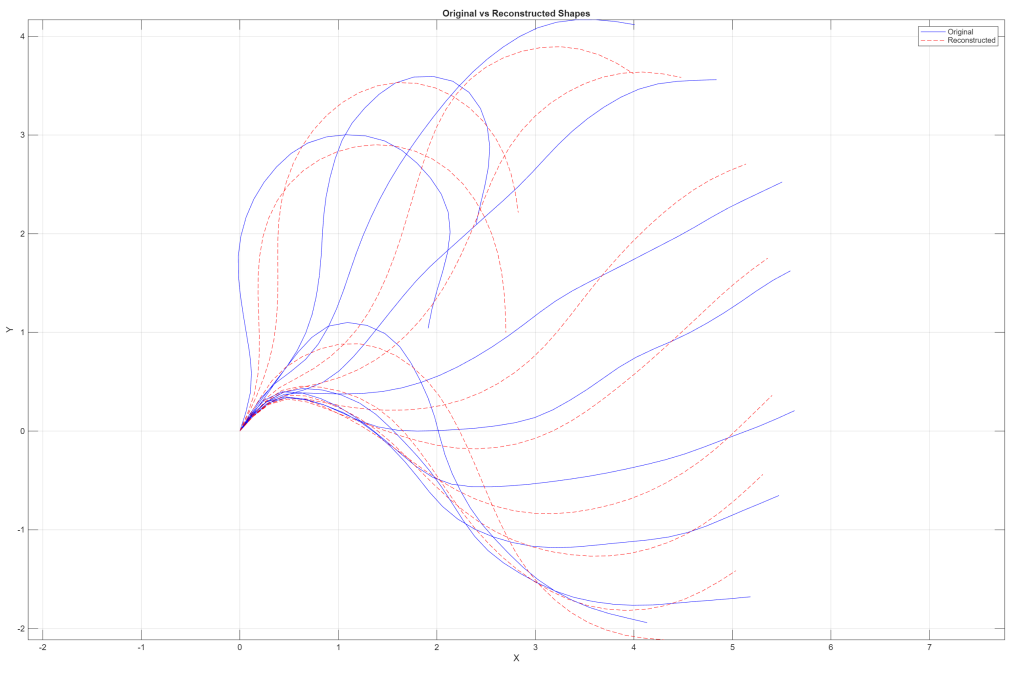

Figure 8A: First few strokes

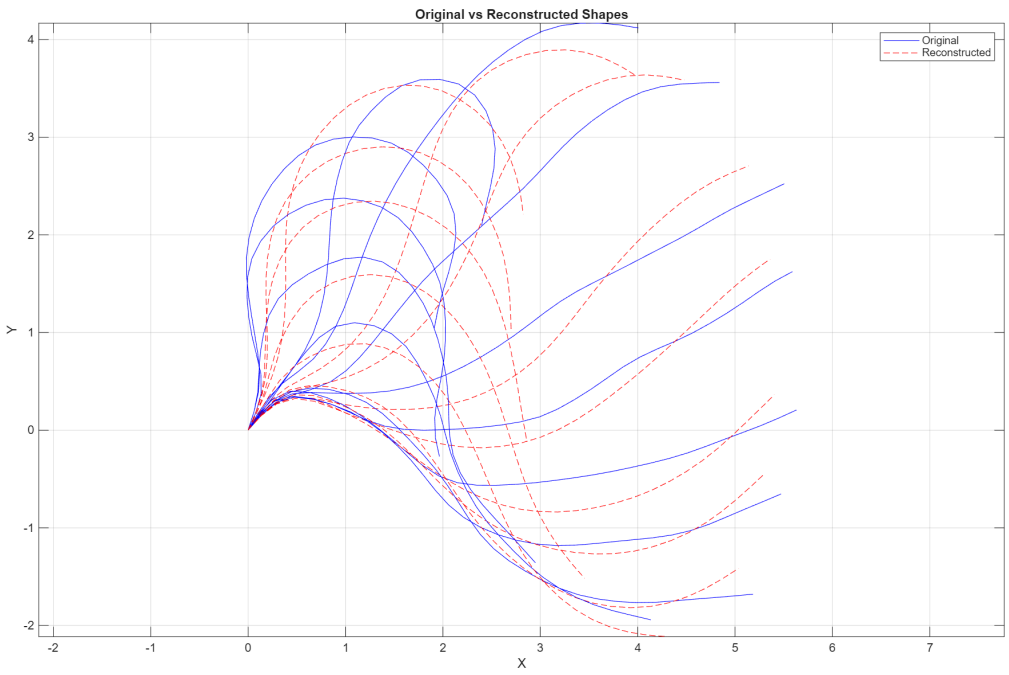

Figure 8B: 1 beat: First 12 time steps

Part (6): Analysis and Conclusion

The reconstruction based on the least squares fit closely matches the original data, demonstrating the effectiveness of using just the first two shape modes U_1, U_2 and their fitted shape scores $V_{1,\text{fit}}, V_{2,\text{fit}}$. These two modes alone capture approximately 96% of the total shape variance.

The minor deviation is expected since the remaining 4% resides on modes 3 and 4. So to improve the reconstruction we could include one or two shape modes. If we say include the third modes for example we could likely raise the captured variance to ~98%.

The decomposition and subsequent reconstruction of data in a lower dimension would be useful in the real world as compressing the data with minimal information loss would make the data easiser to work with and analyze compared to if it were in a higher dimension.

References:

References:

1. **Werner et al. (2014)** Shape Mode Analysis Exposes Movement Patterns in Biology: Flagella and Flatworms as Case Studies . Printable PDF: https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0113083\&type=printable
2. **Fig 1** https://i.ibb.co/GvZXHZLs/figure1.png
3. **Fig 2A** https://i.ibb.co/8LRCHrF2/figure2.png
4. **Fig 2B** https://i.ibb.co/qMVsmn5h/figure2.png
5. **Fig 3B** https://i.ibb.co/Jw7wWQtg/figure3.png
6. **Fig 3A** https://i.ibb.co/8nKwSybf/figure3.png
7. **Fig 4** https://i.ibb.co/t191Cng/figure4.png
8. **Fig 5** https://i.ibb.co/yJbpvvT/figure5.png
9. **Fig 6** https://i.ibb.co/vv12dTTR/figure6.png
10. **Fig 7A** https://i.ibb.co/nsMmYRfK/figure7.png
11. **Fig 7B** https://cdn.mathpix.com/cropped/2025_07_30_73dec793a37a0ff85a0fg-1.jpg?height=906\&width=1335\&top_left_y=906\&top_left_x=366
12. **Fig 8** https://i.ibb.co/chN8Q9pg/figure8.png
13. **Fig 8A** https://i.ibb.co/ZRJTgBv0/figure8.png