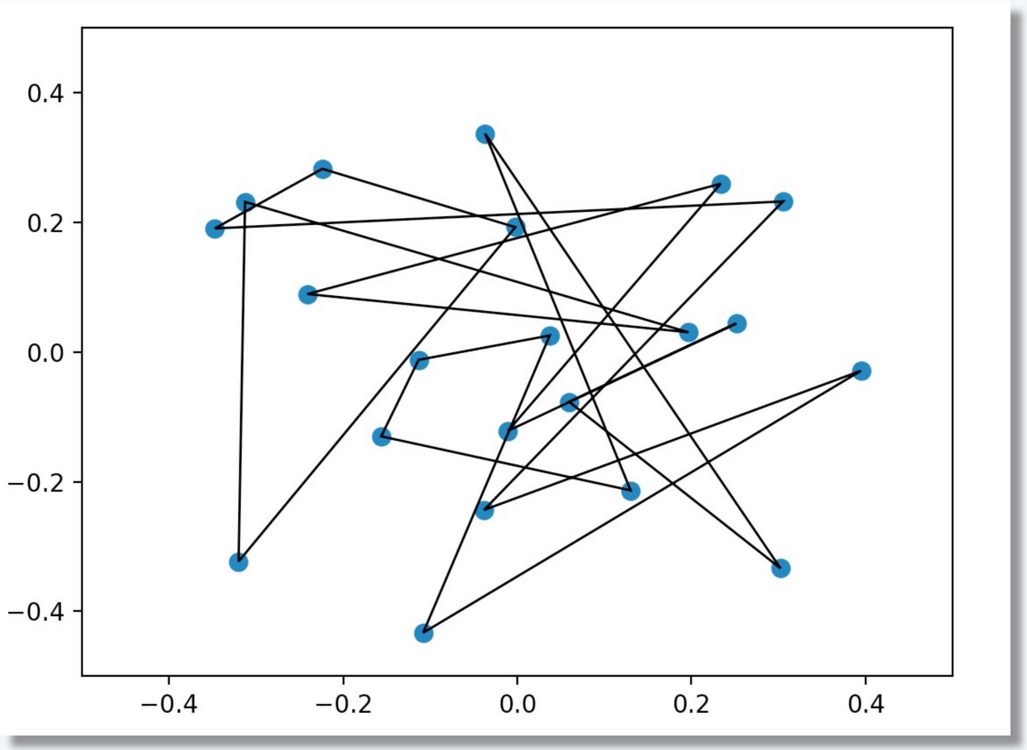
DM561 – Linear Algebra with Applications

Week 48, 2021

<u>From Random Polygon to Ellipse</u>



Daniel Merkle

daniel@imada.sdu.dk

John von Neumann Lecture

Established in 1959, the prize honors John von Neumann, a founder of modern computing. The lecture is awarded annually for distinguished contributions to applied mathematics and for the effective communication of these ideas to the community.

Prize Description

The John von Neumann Lecture is awarded annually to one individual for outstanding and distinguished contributions to the field of applied mathematics and for the effective communication of these ideas to the community. It is one of SIAM's most distinguished prizes as well as an important lecture at the SIAM Annual Meeting.

The 2018 winner: Charles Francis Van Loan

The topic of this lecture is based on his speech when receiving the award (Sept 2018).

DM561 – Linear Algebra with Applications

<u>Based on:</u>

A.N. Elmachtoub, C.F. Van Loan (2010), From random polygon to ellipse: an eigenanalysis, SIAM Rev. 52, 151–170.

John von Neumann prize lecture, Sept 2018, for C.F. van Loan

Intro Programming with Matlab (2008)

$\downarrow$

Intro Matrix Computations (2009)

$\downarrow$

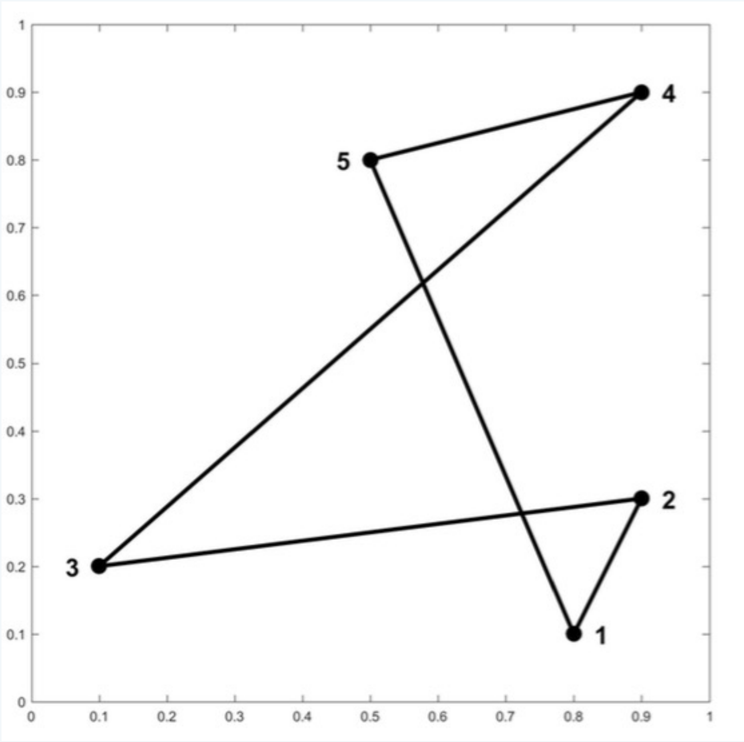
SIAM Review (2010)

$\downarrow$

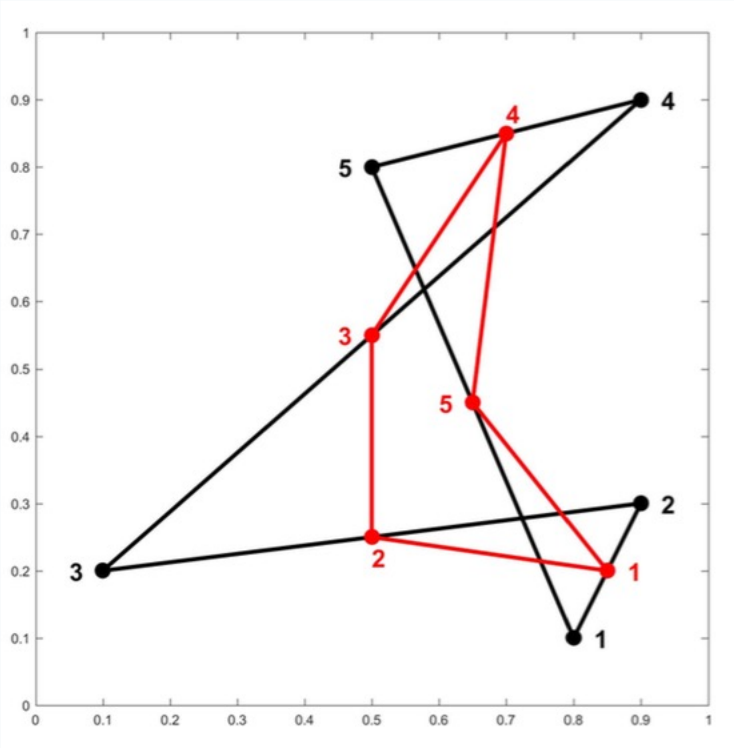
SIAM News (2018)

Display a sequence of polygons where each polygon is obtained from its predecessor by connecting the midpoints of its sides.

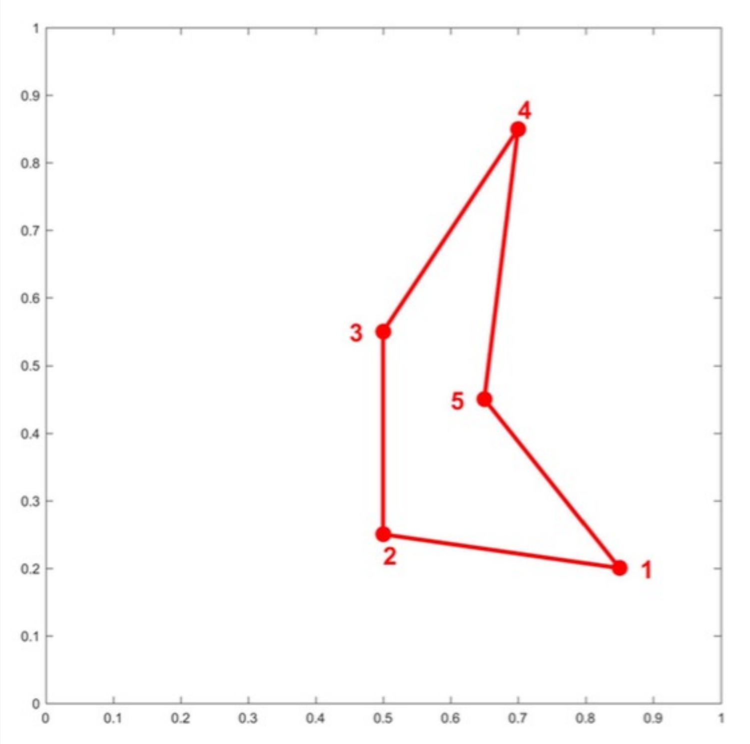
Let the original polygon be random.



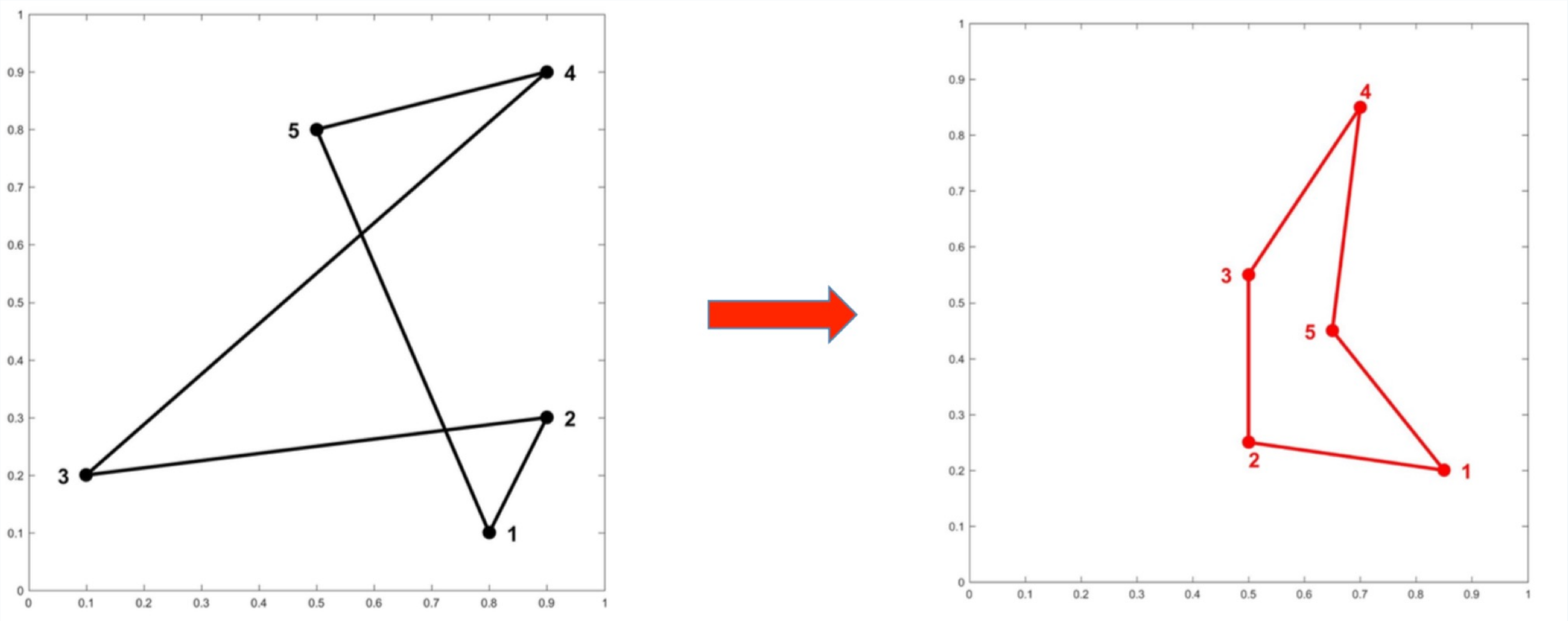
A random polygon (5 points)



Connecting the midpoints



New polygon



One step. This obviously can be repeated.

One Step in Vector Terminology

Assume the following five points define a close polygon: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)$

$$\begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \\ \widehat{x}_3 \\ \widehat{x}_4 \\ \widehat{x}_5 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 + x_2 \\ x_2 + x_3 \\ x_3 + x_4 \\ x_4 + x_5 \\ x_5 + x_1 \end{bmatrix} \qquad \begin{bmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \widehat{y}_3 \\ \widehat{y}_4 \\ \widehat{y}_5 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} y_1 + y_2 \\ y_2 + y_3 \\ y_3 + y_4 \\ y_4 + y_5 \\ y_5 + y_1 \end{bmatrix}$$

One Step in Matrix Terminology

$$\widehat{x} = \begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \\ \widehat{x}_3 \\ \widehat{x}_4 \\ \widehat{x}_5 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} x_1 + x_2 \\ x_2 + x_3 \\ x_3 + x_4 \\ x_4 + x_5 \\ x_5 + x_1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \equiv M_5 x$$

$$\widehat{y} = \begin{bmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \widehat{y}_3 \\ \widehat{y}_4 \\ \widehat{y}_5 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} y_1 + y_2 \\ y_2 + y_3 \\ y_3 + y_4 \\ y_4 + y_5 \\ y_5 + y_1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} \equiv M_5 y.$$

In general

$$\widehat{x} = M_n x$$

$$\widehat{y} = M_n y$$

$$M_n = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & 1 & \vdots \\ 0 & 0 & \cdots & & 1 & 1 \\ 1 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

A First Try

length = 1, i.e., $x^{(0)} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$

$$||x^{(0)}||_2 = \sqrt{\sum_{1 \leq i \leq n} x_i^2} = 1$$

Algorithm 1

Input: Unit 2-norm n-vectors $x^{(0)}$ *and* $y^{(0)}$.

Display $\mathcal{P}_0 = \mathcal{P}(x^{(0)}, y^{(0)})$.

for $k = 1, 2, \ldots$

% Compute $\mathcal{P}_k = \mathcal{P}(x^{(k)}, y^{(k)})$ from $\mathcal{P}_{k-1} = \mathcal{P}(x^{(k-1)}, y^{(k-1)})$

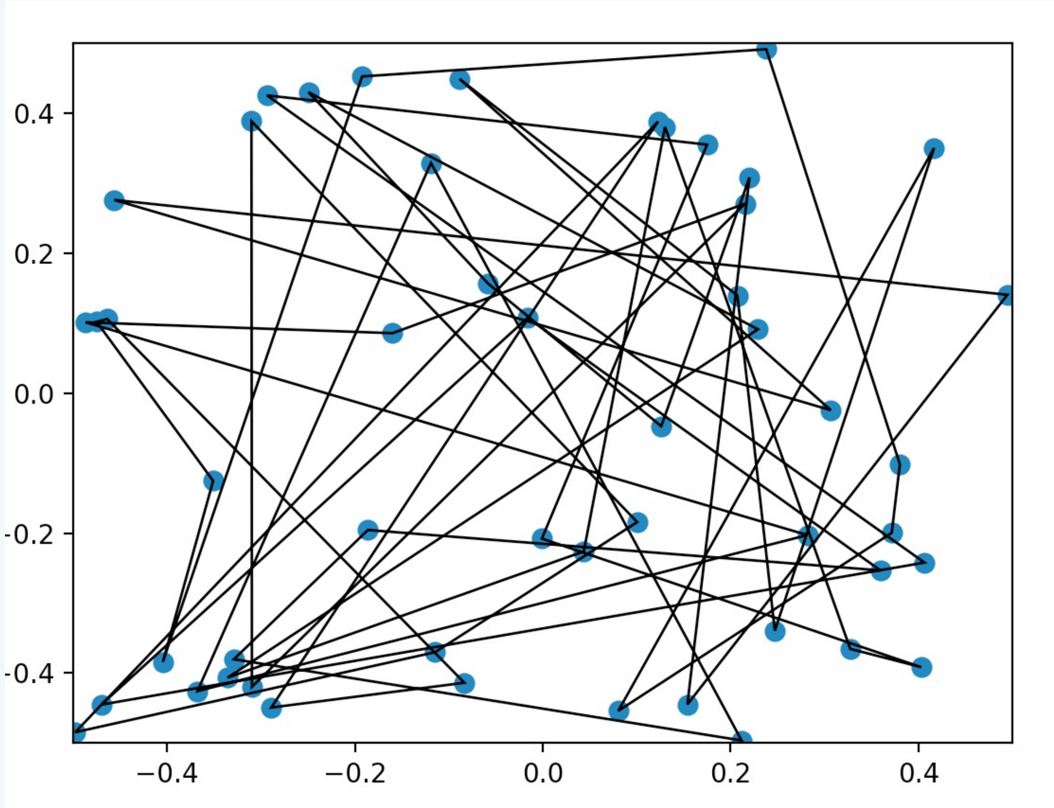
$$x^{(k)} = M_n x^{(k-1)}$$

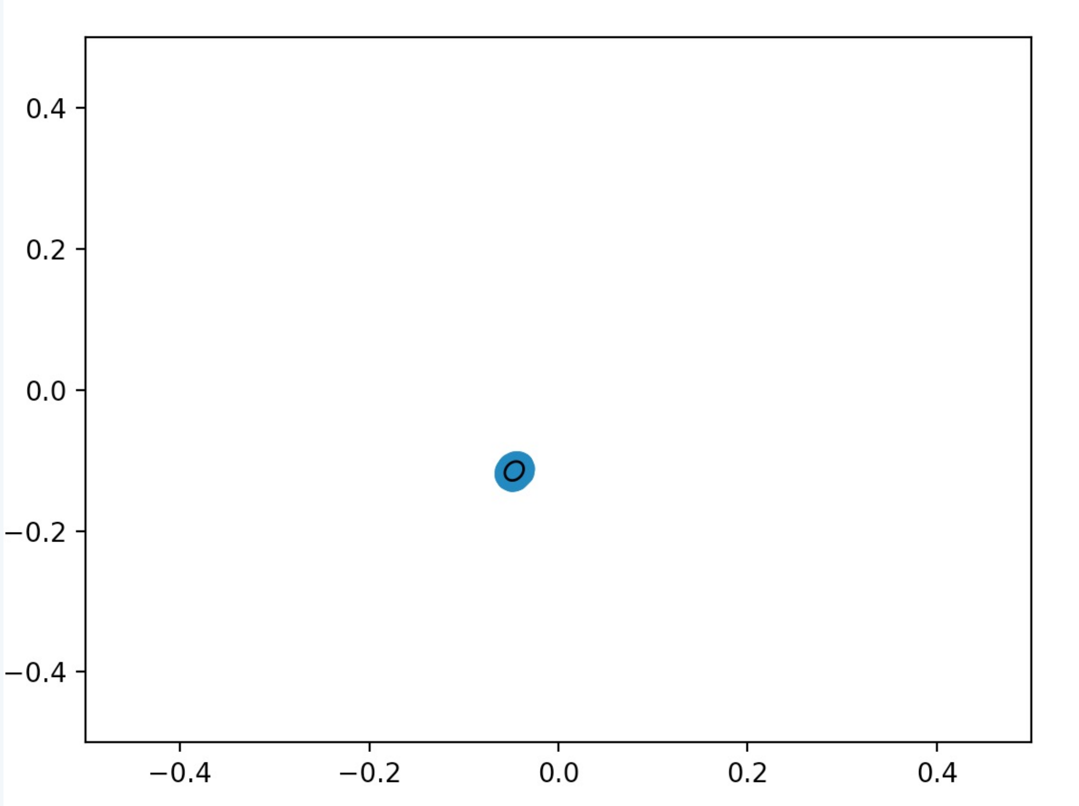
$$y^{(k)} = M_n y^{(k-1)}$$

Display $\mathcal{P}_k$.

end

Not too interesting …





The points seem to converge to a point. Not too surprisingly, it’s the centroid of the input points.

Centroid remains unchanged

Centroid:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{e^T x}{n} \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{e^T x}{n} \qquad e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

As $e^T M_n = e^T$ it holds:

$$\frac{e^T x^{(k)}}{n} = \frac{e^T M_n x^{(k-1)}}{n} = \frac{e^T x^{(k-1)}}{n}$$

$$\frac{e^T y^{(k)}}{n} = \frac{e^T M_n y^{(k-1)}}{n} = \frac{e^T y^{(k-1)}}{n}.$$

A Second Try

$$x^{(0)} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\sum_{1 \le i \le n} x_i = 0$$

Algorithm 2

Input: Unit 2-norm n-vectors $x^{(0)}$ *and* $y^{(0)}$ *whose components sum to zero.*

Display $\mathcal{P}_0 = \mathcal{P}(x^{(0)}, y^{(0)})$.

for $k = 1, 2, \ldots$

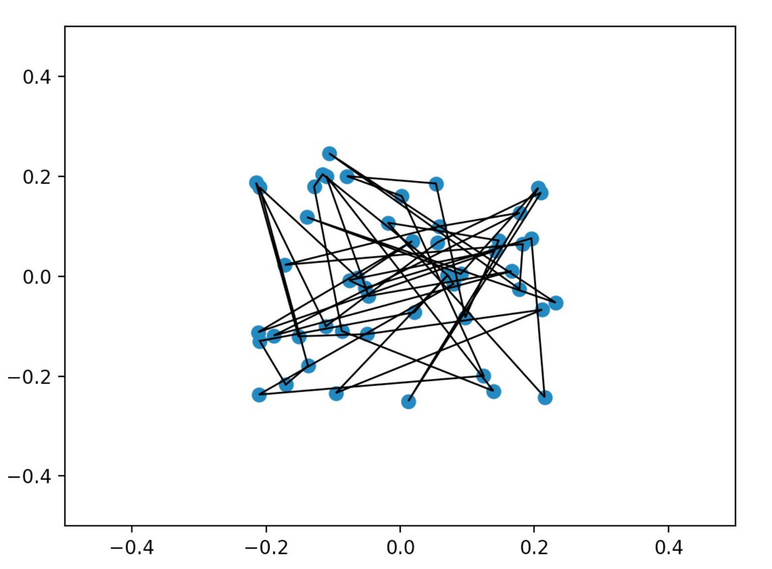
% Compute $\mathcal{P}_k = \mathcal{P}(x^{(k)}, y^{(k)})$ from $\mathcal{P}_{k-1} = \mathcal{P}(x^{(k-1)}, y^{(k-1)})$

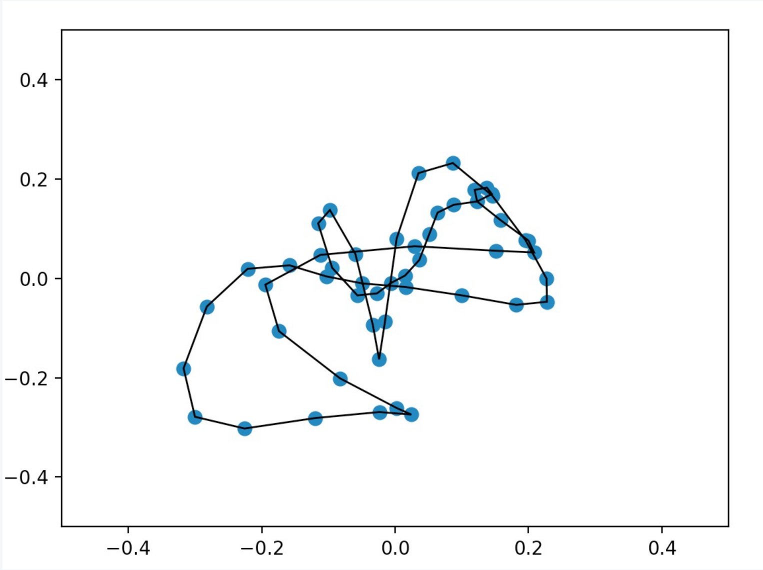
$f = M_n x^{(k-1)}, \quad x^{(k)} = f / \| f \|_2$

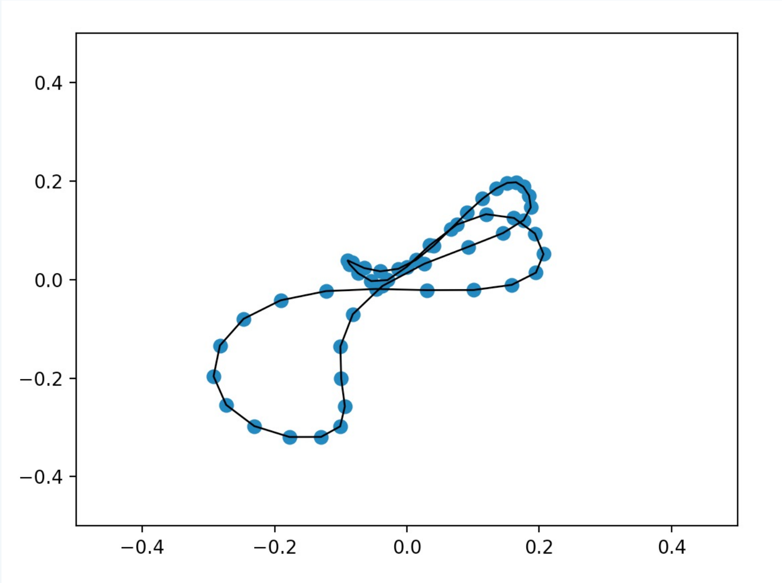
$g = M_n y^{(k-1)}, \quad y^{(k)} = g / \| g \|_2$

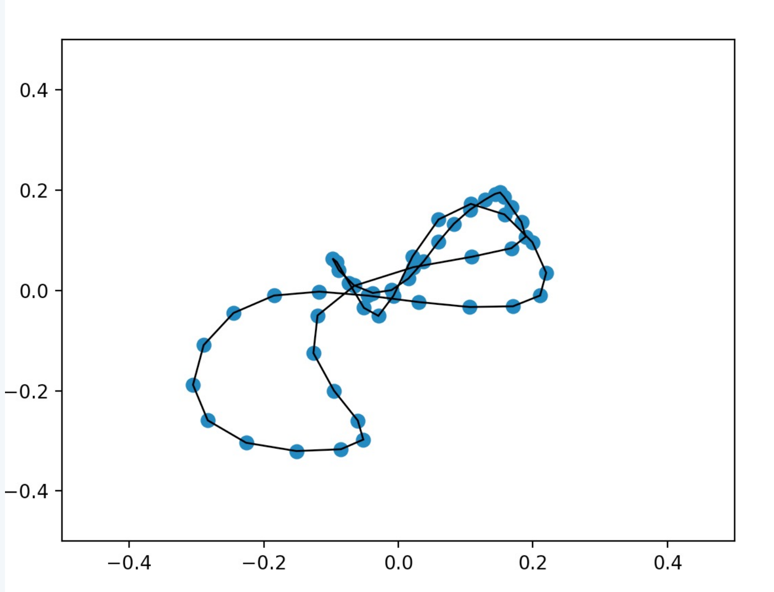
Display $\mathcal{P}_k$.

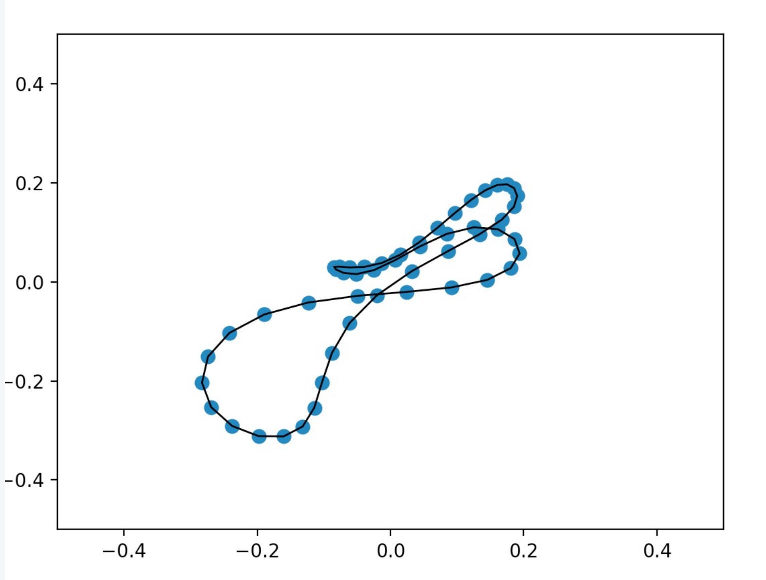
end





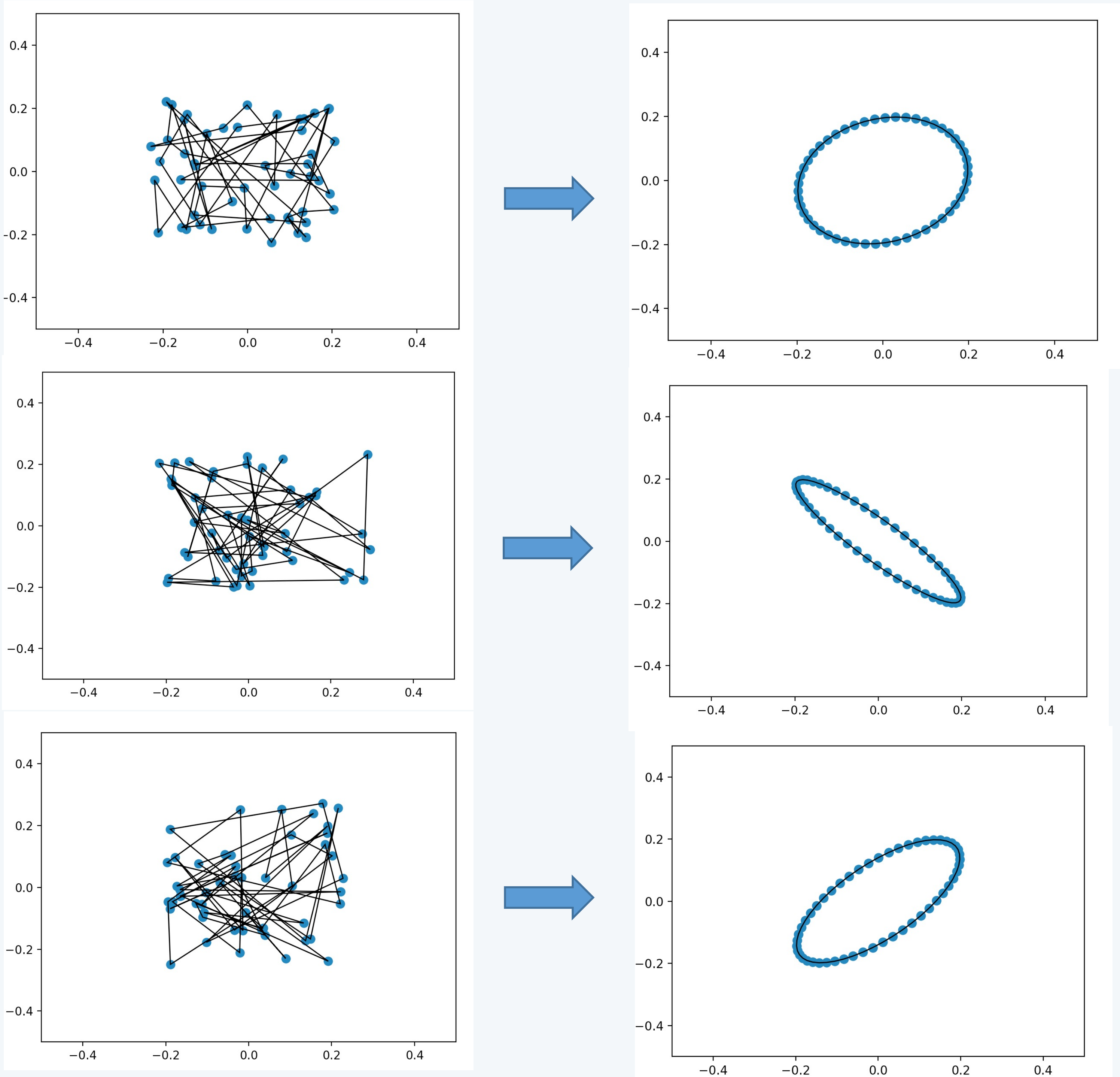


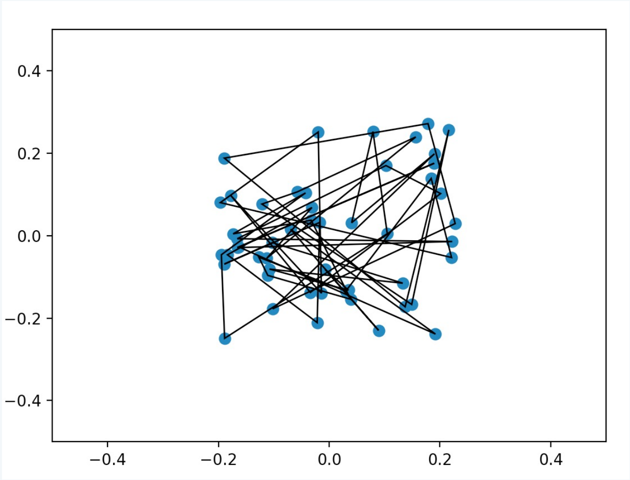


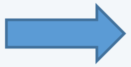


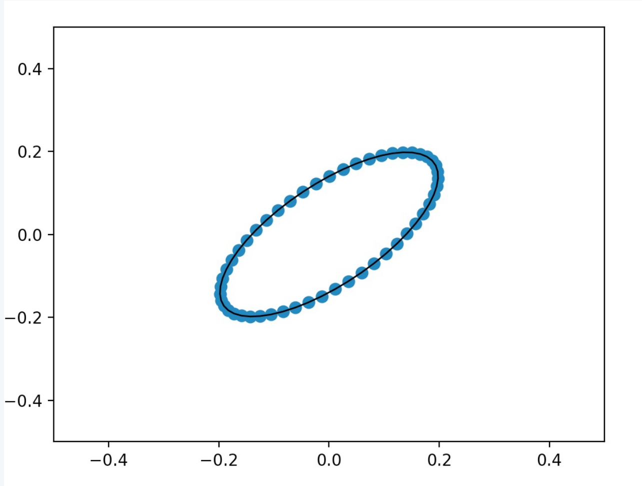
What’s happening???

Three test runs









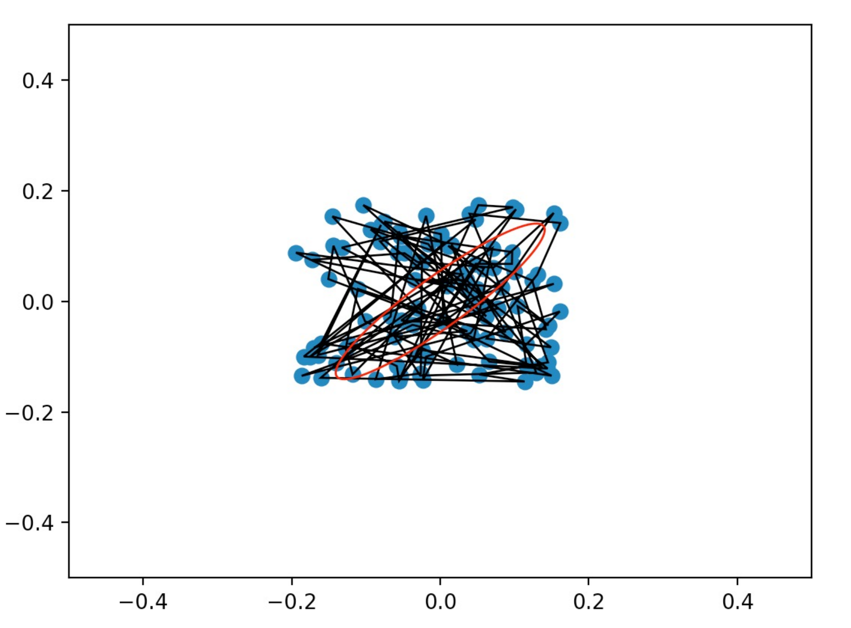
The points seem to converge to an ellipse with a 45-degree tilt

1. What is the limiting ellipse and why the 45-degree tilt?
2. How long does it take to "converge"?
3. Does it always converge?
4. What is the inverse of the repeated polygon averaging process, and does it always exits?

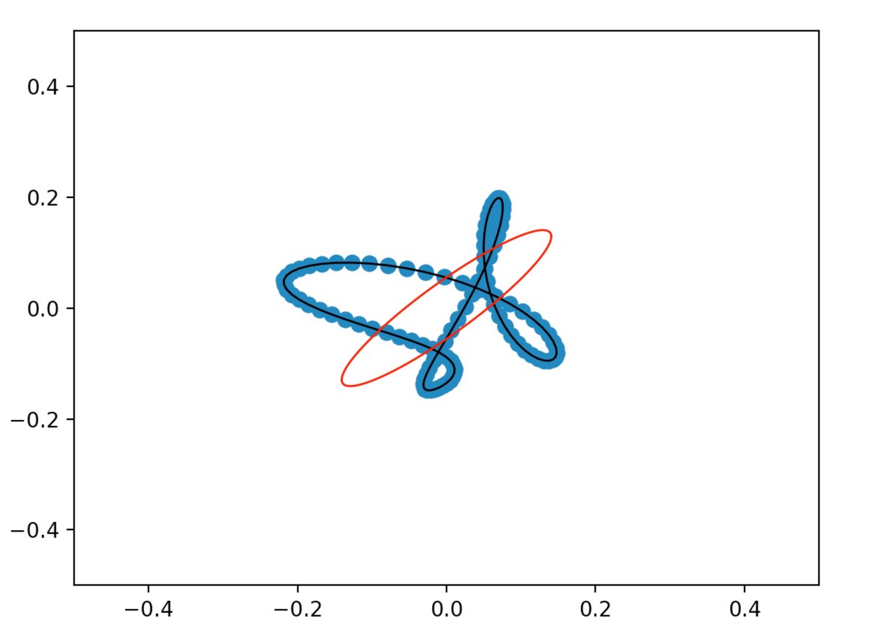
The ellipse can be computed in advance

<u>Example: Polygon with 101 points</u>

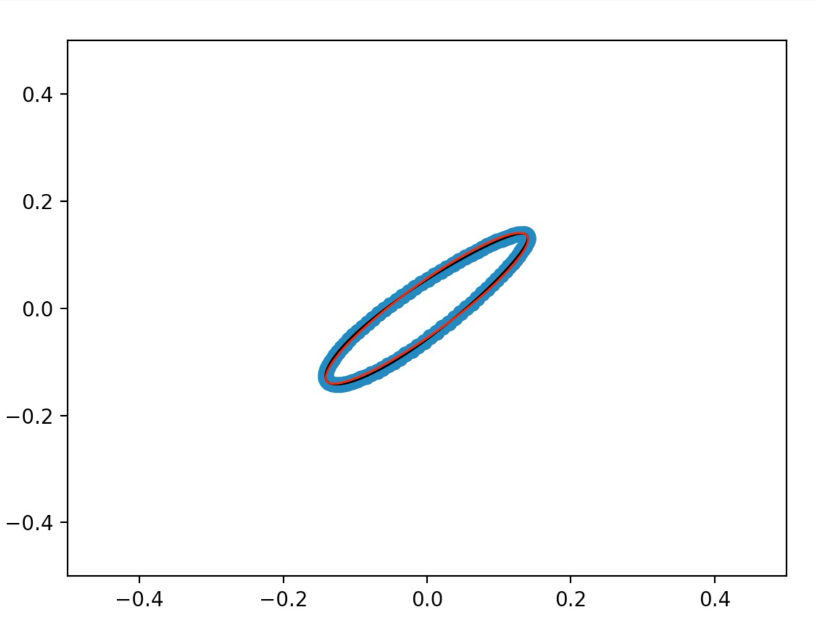
Step 1



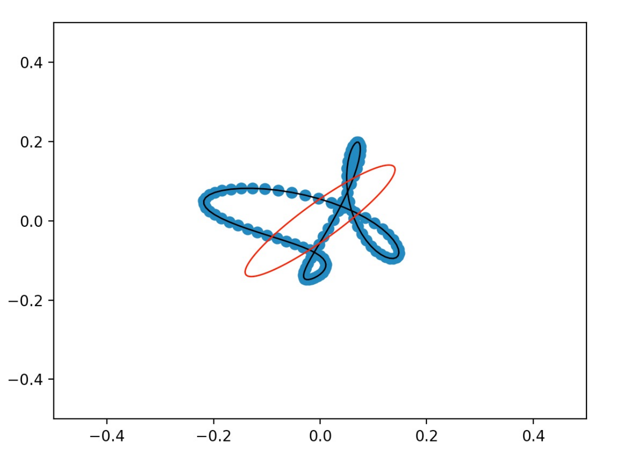
Step 100



Step 500



Pre-Computation of the ellipse



<u>Background:</u>

The analysis is not trivial, and requires

- an understanding of the eigen-system of the matrix M
- decomposition techniques (SVD- and Schur- decomposition)
- Interested?
  - Follow the wonderful von Neumann Price Lecture from **Charles F. Van Loan**
  - Read the (not so easy parts of the) paper
    “**From Random Polygon to Ellipse: An Eigenanalysis”, Adam N. Elmachtoub and Charles F. van Loan**

Pre-Computation of the ellipse

For a polygon with n points, define vectors **c** and **s** as follows (they form a orthonormal basis)

$$\tau = \begin{bmatrix} 0 \\ 2\pi/n \\ 4\pi/n \\ \vdots \\ 2(n-1)\pi/n \end{bmatrix} \qquad c = \sqrt{2/n}\begin{bmatrix} \cos(\tau_1) \\ \cos(\tau_2) \\ \vdots \\ \cos(\tau_n) \end{bmatrix} \qquad s = \sqrt{2/n}\begin{bmatrix} \sin(\tau_1) \\ \sin(\tau_2) \\ \vdots \\ \sin(\tau_n) \end{bmatrix}$$

Pre-Computation of the "ellipse"

The polygon converges (at even even steps) to the closed polygon defined by vector $u^{(0)}$ and $v^{(0)}$

$$u^{(0)} = \cos(\theta_u)c \ + \ \sin(\theta_u)s$$

$$v^{(0)} = \cos(\theta_v)c \ + \ \sin(\theta_v)s$$

where

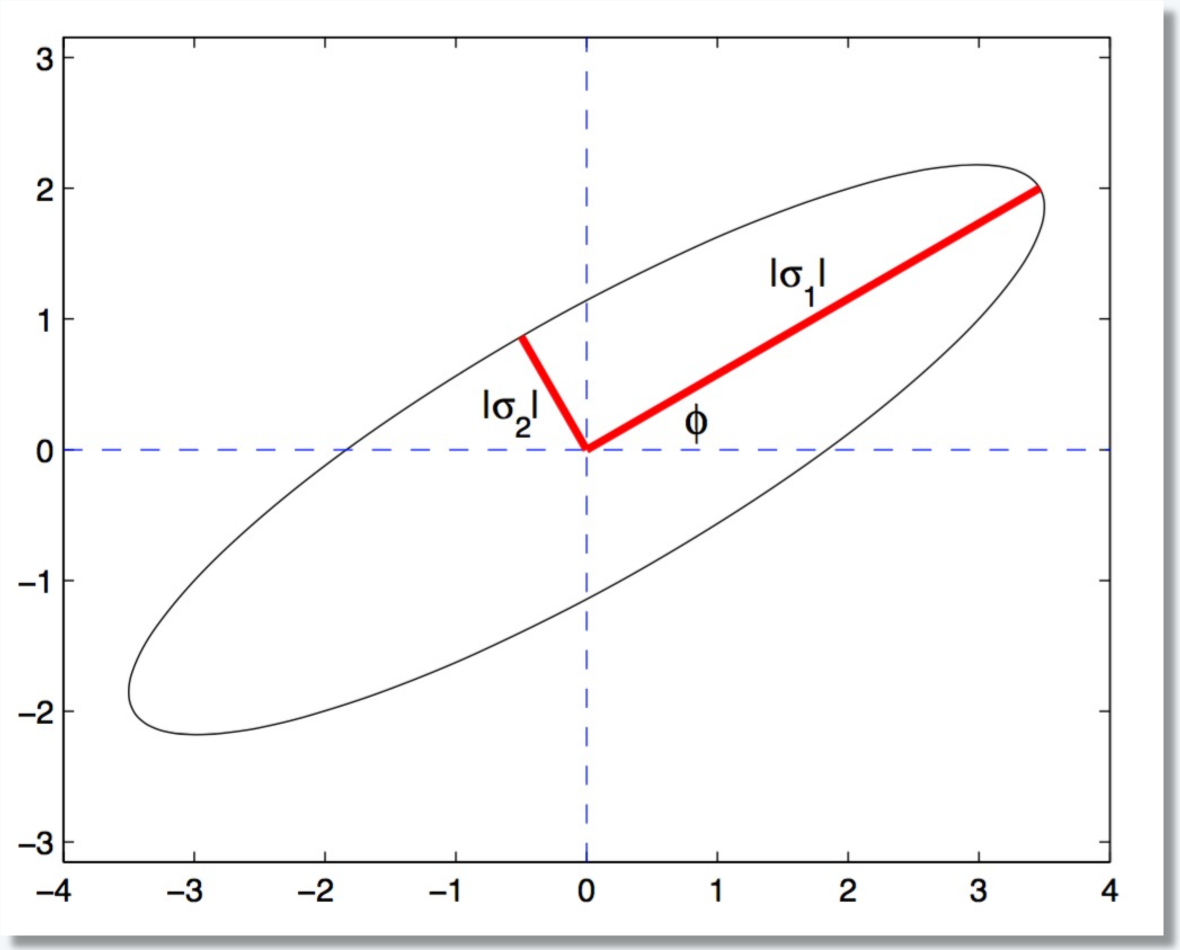
$$\cos(\theta_u) \ = \ \frac{c^T x^{(0)}}{\sqrt{(c^T x^{(0)})^2 + (s^T x^{(0)})^2}} \qquad \sin(\theta_u) \ = \ \frac{s^T x^{(0)}}{\sqrt{(c^T x^{(0)})^2 + (s^T x^{(0)})^2}}$$

$$\cos(\theta_v) = \frac{c^T y^{(0)}}{\sqrt{(c^T y^{(0)})^2 + (s^T y^{(0)})^2}} \qquad \sin(\theta_v) \ = \ \frac{s^T y^{(0)}}{\sqrt{(c^T y^{(0)})^2 + (s^T y^{(0)})^2}}.$$

Proof: paper! (Note, TYPO in paper)

Pre-Computation of the ellipse

Also in the paper: An understanding of the shape of the ellipse in terms of $\sigma_1 \text{ and } \sigma_2$:



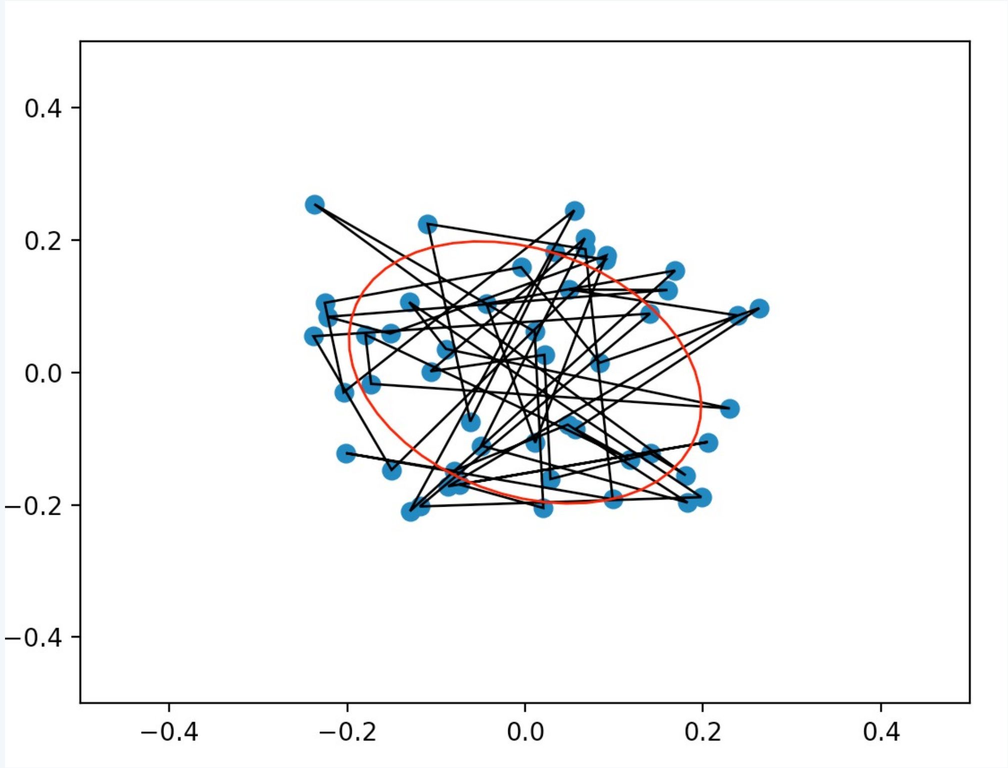
Backwards in Time?

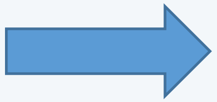
Generating Polygons $P_{-1}, P_{-2}, \ldots$

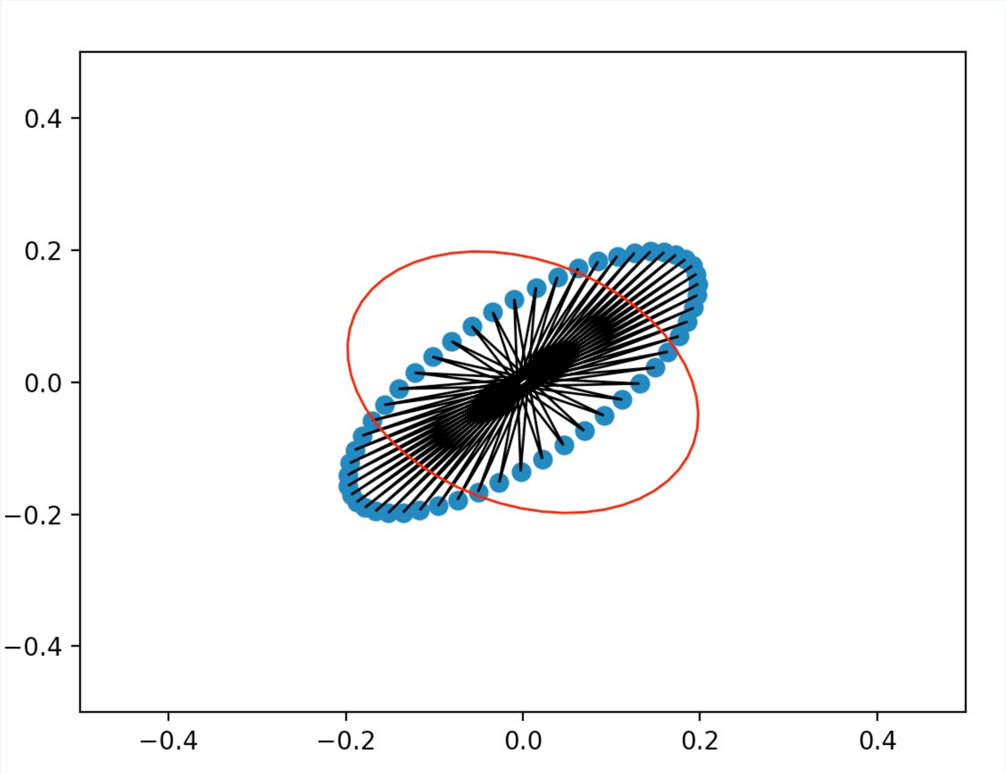
```
x = rand(n,1); x = x - mean(x); x = x/norm(x)
y = rand(n,1); y = y - mean(y); y = y/norm(y)
for k = 1,2,...
    x = inv(M)*x; x = x/norm(x)
    y = inv(M)*y; y = y/norm(y)
end
```

$$M = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \end{bmatrix}$$

Example: n=51 points. WTH?







Does M^{-1} always exist? No! **Only for odd n.**

[For intuition on the non-existence of the inverse for even n, chose 4 random points (assume time step t) and try to infer 4 points (time t-1) which lead to the 4 points at time t. After being successful, try to find another 4 points which can lead to the same 4 points. After being successful, what does this tell us about the inverse of M?]

An important note of caution

Algorithm 2

Input: Unit 2-norm n-vectors $x^{(0)}$ *and* $y^{(0)}$ *whose components sum to zero.*

Display $\mathcal{P}_0 = \mathcal{P}(x^{(0)}, y^{(0)})$.

for $k = 1, 2, \ldots$

% Compute $\mathcal{P}_k = \mathcal{P}(x^{(k)}, y^{(k)})$ from $\mathcal{P}_{k-1} = \mathcal{P}(x^{(k-1)}, y^{(k-1)})$

$$f = M_n x^{(k-1)}, \quad x^{(k)} = f/\| f \|_2$$

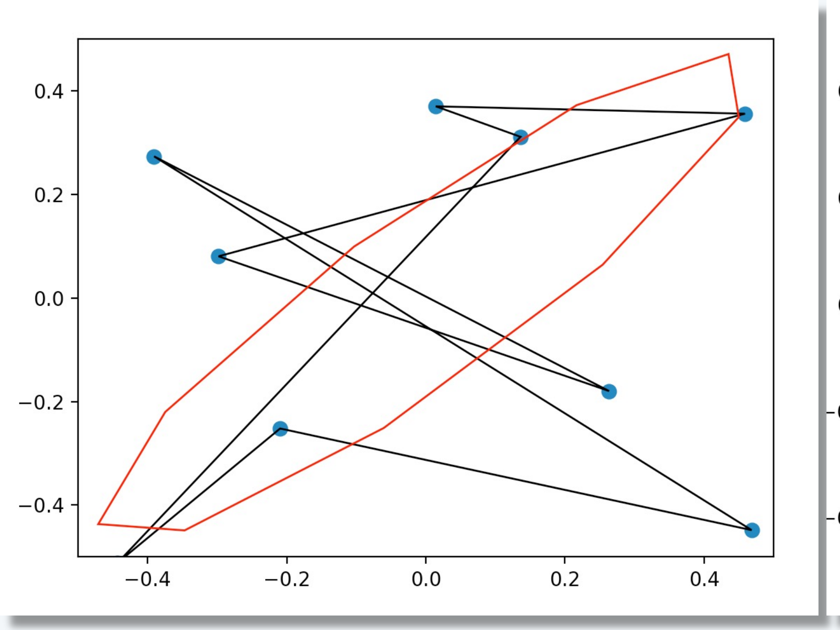
$$g = M_n y^{(k-1)}, \quad y^{(k)} = g/\| g \|_2$$

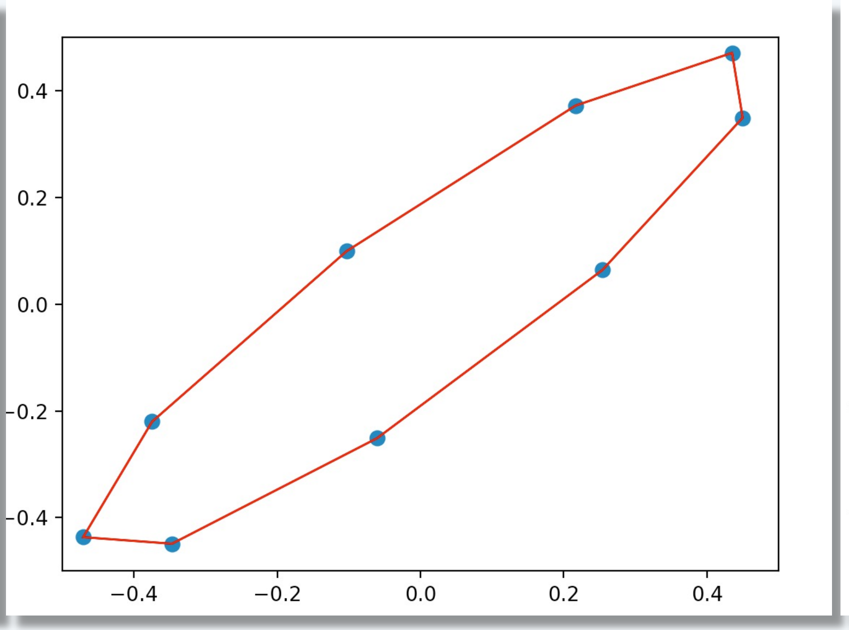
Display $\mathcal{P}_k$.

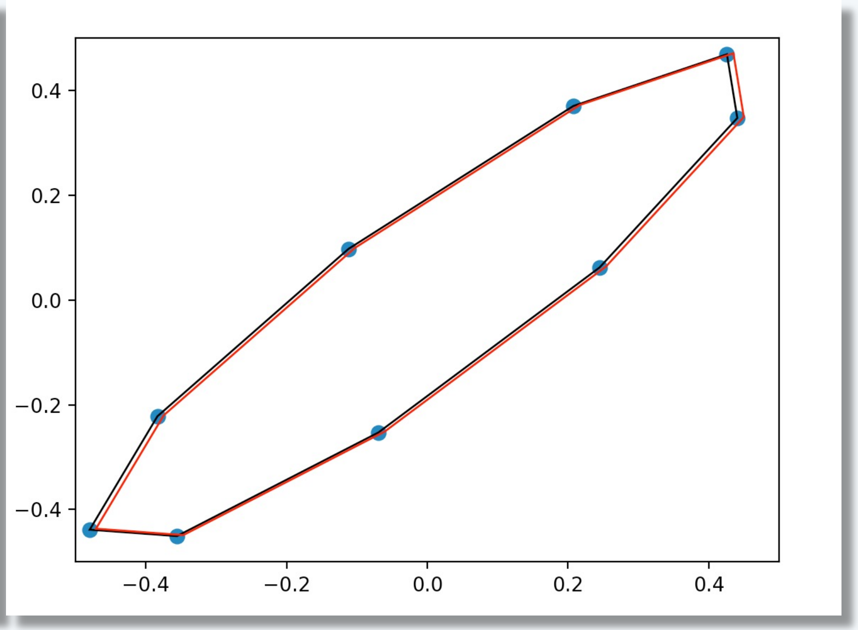
end

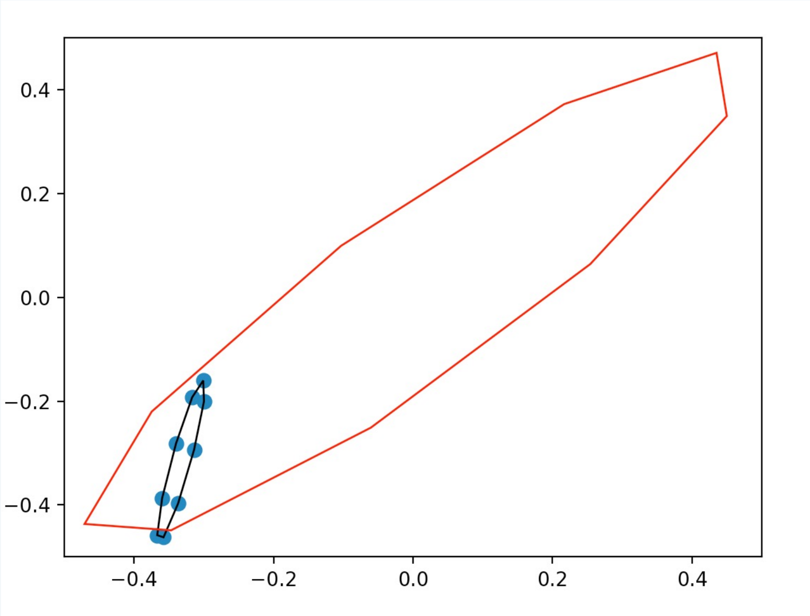
We learned that the centroids do not change (by mathematical proof!)
When submitting a solution to the lab exercise, please (try to) check: is this true in your simulation?
If not, how to "fix" it? (-> enforce a mean(x)=0 and mean(y) after each step? Is this a fix? No! But at least better.)

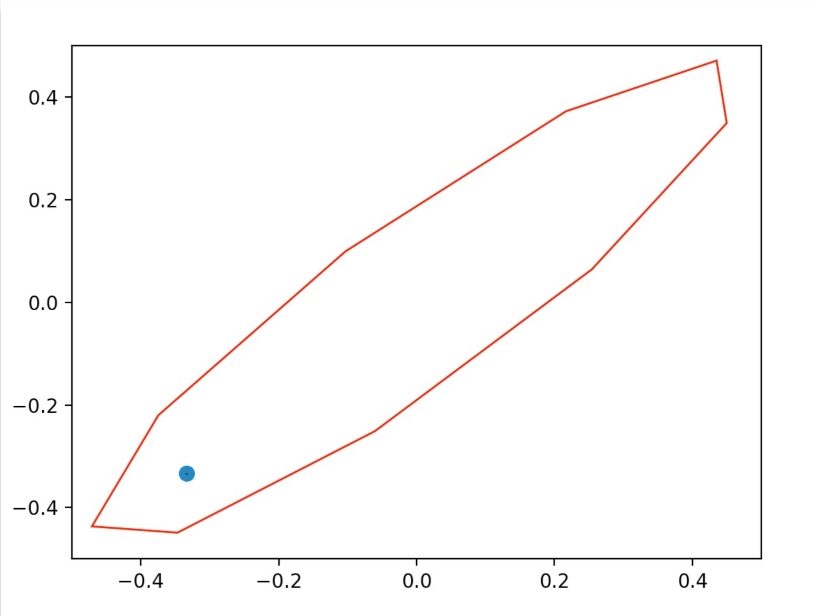
Numerical issues can be very complicated i.) to detect and ii.) to resolve!!











Iteration with 9 points at step 0, 400, 500, 600, and 700, without enforcing the mean to be 0. The simulation was done using M^{10} for increased simulation speed.

Numerical Issues!