SIAM Review
Vol. 52, No. 1, pp. 151–170

From Random Polygon to Ellipse: An Eigenanalysis*

Adam N. Elmachtoub†
Charles F. Van Loan‡

Abstract. Suppose x and y are unit 2-norm n-vectors whose components sum to zero. Let $\mathcal{P}(x,y)$ be the polygon obtained by connecting $(x_1,y_1),\ldots,(x_n,y_n),(x_1,y_1)$ in order. We say that $\widehat{\mathcal{P}}(\widehat{x},\widehat{y})$ is the normalized average of $\mathcal{P}(x,y)$ if it is obtained by connecting the midpoints of its edges and then normalizing the resulting vertex vectors $\widehat{x}$ and $\widehat{y}$ so that they have unit 2-norm. If this process is repeated starting with $\mathcal{P}_0=\mathcal{P}(x^{(0)},y^{(0)})$, then in the limit the vertices of the polygon iterates $\mathcal{P}(x^{(k)},y^{(k)})$ converge to an ellipse $\mathcal{E}$ that is centered at the origin and whose semiaxes are tilted forty-five degrees from the coordinate axes. An eigenanalysis together with the singular value decomposition is used to explain this phenomenon. The problem and its solution is a metaphor for matrix-based research in computational science and engineering.

Key words. power method, eigenvalue analysis, ellipse, polygon

AMS subject classifications. 15A18, 65F15

DOI. 10.1137/090746707

1. Introduction.

Suppose we have a pair of vectors $x,y\in\mathbb{R}^n$ that specify the vertices of a polygon $\mathcal{P}$. If we connect the edge midpoints, then we obtain another polygon $\widehat{\mathcal{P}}$. Since each midpoint is a vertex average, we can think of $\widehat{\mathcal{P}}$ as an "averaged" version of $\mathcal{P}$. See Figure 1.1, which displays the averaging of a random pentagon.

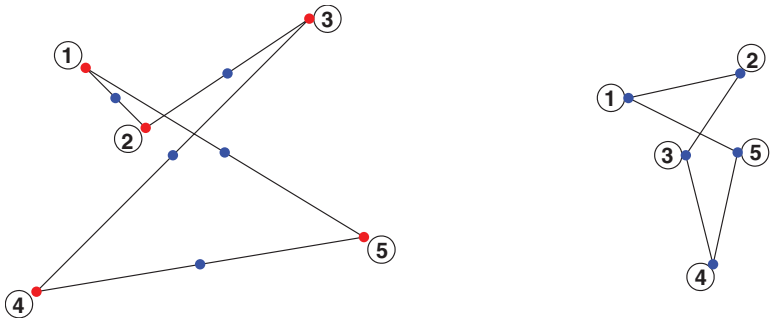

Fig. 1.1 *A pentagon $\mathcal{P}$ (left) and its average $\widehat{\mathcal{P}}$ (right) obtained by connecting midpoints.*

*Received by the editors January 15, 2009; accepted for publication (in revised form) November 10, 2009; published electronically February 5, 2010.
http://www.siam.org/journals/sirev/52-1/74670.html

†School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853-7510 (ane6@cornell.edu).
‡Department of Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853-7510 (cv@cs.cornell.edu).

If we repeat this process, do the polygons converge to something structured or do they remain "tangled up"? Our goal is to frame this question more precisely and provide an answer.

1.1. A Matrix-Vector Description. The connect-the-midpoints operation invites formulation as a vector operation. After all, we have a vector of identical operations to perform:

for $i = 1{:}n$
    Compute the midpoint of $\mathcal{P}$'s ith edge.
end

In scalar terms, an individual midpoint computation involves the averaging of two x-values and the averaging of two y-values, e.g.,

$$\widehat{x}_3 \;=\; \frac{x_3 + x_4}{2}, \qquad\qquad \widehat{y}_3 \;=\; \frac{y_3 + y_4}{2}.$$

In vector terms, the mission of the above loop is to average the x-vector with its upshift and the y-vector with its upshift:

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

The term "upshift" is appropriate because the vector components move up one notch with the top component wrapping around to the bottom. Because these transformations are linear, they can be described as matrix-vector products:

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

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

The subscript "5" in M_5 indicates the dimension of the matrix. In general, the transition from $\mathcal{P}$ to its average $\widehat{\mathcal{P}}$ requires the multiplications

$$\widehat{x} \;=\; M_n x, \qquad\qquad \widehat{y} \;=\; M_n y,$$

where M_n is the n-by-n matrix defined by

$$M_n = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & 1 & \vdots \\ 0 & 0 & \cdots & & 1 & 1 \\ 1 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}. \tag{1.1}$$

We refer to M_n as the *averaging matrix* and the vectors x, y, $\widehat{x}$, and $\widehat{y}$ as *vertex vectors*. The notation $\mathcal{P}(x, y)$ is used to denote the polygon defined by the vertex vectors x and y. Note that after k averaging steps we have the polygon $\mathcal{P}(M_n^k x, M_n^k y)$.

The linear algebraic formulation of the polygon averaging process is appealing for two reasons:

> *Algorithmic Reason.* It allows us to describe the repeated averaging process more succinctly.
>
> *Analysis Reason.* It reveals that the central challenge is to understand the behavior of the vectors $M_n^k x$ and $M_n^k y$.

Being able to "spot" matrix-vector operations is a critical talent in computational science and engineering.

1.2. Iteration: A First Try. There is no better way to develop an intuition about repeated polygon averaging than to display graphically the progression from one polygon to the next.

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

By experimenting with this procedure we discover that the polygon sequence converges to a point! See Figure 1.2, which displays $\mathcal{P}_0$, $\mathcal{P}_5$, $\mathcal{P}_{20}$, and $\mathcal{P}_{100}$ for a typical $n = 15$ example. A rigorous explanation for the limiting behavior of Algorithm 1 will follow. We first explain why the limit point is the *centroid* of $\mathcal{P}_0$. The centroid $(\bar{x}, \bar{y})$ of an n-sided polygon $\mathcal{P}(x, y)$ is defined by the vector centroids

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{e^T x}{n}, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{e^T x}{n},$$

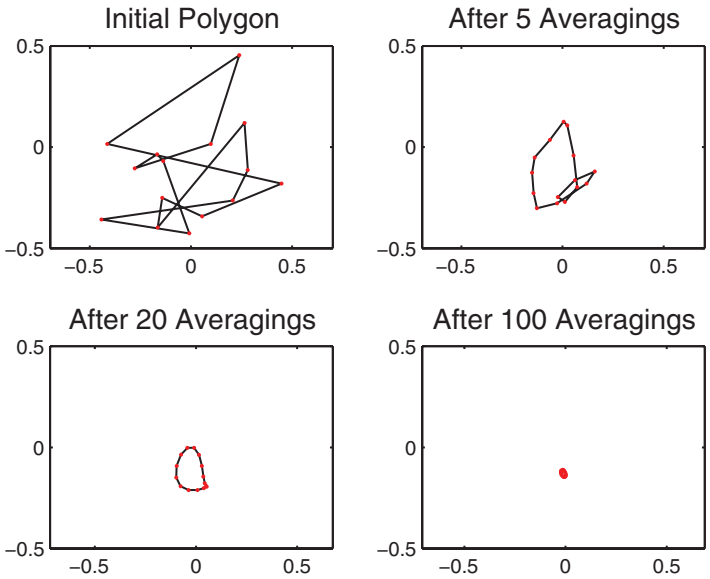

Fig. 1.2 *Repeated averaging.*

where $e \in \mathbb{R}^n$ is the vector of ones, i.e.,

$$e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

(Its dimension will always be clear from context.) Notice that $e^T M_n = e^T$ and thus

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

and

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

This shows that each polygon in the sequence generated by Algorithm 1 has the same centroid. It follows that if the sequence of polygons $\{\mathcal{P}_k\}$ converges to a point, then that point must be the centroid of $\mathcal{P}_0$.

It is interesting to examine experimentally the rate of convergence. For a given $\mathcal{P}_0$ and $\delta > 0$, let k_δ be the smallest value of k such that

$$\sqrt{\sum_{i=1}^{n} |x_i^{(k)} - \bar{x}|^2 + |y_i^{(k)} - \bar{y}|^2} \leq \delta.$$

This just means that the vertices of $\mathcal{P}_{k_*}$ are within δ of its centroid. Experimentation reveals that the average value of k_δ increases with the square of n. See Table 1.1. The averages in the table are based upon 100 trials. The initial vertex vectors were generated using the MATLAB `rand` function.

Table 1.1 *Average number of iterations until the radius of $\mathcal{P}_k$ is about 10^{-3}.*

n	Average k_δ	
10	127	
20	485	$(\delta = 0.001)$
40	1830	
80	6857	

1.3. Iteration: A Second Try. Figure 1.2 suggests that the polygons "untangle" before they disappear from view in Algorithm 1. In order to examine this more carefully, we introduce a pair of normalizations that keep the $\mathcal{P}_k$ from collapsing down to a point:

1. We assume that $\mathcal{P}(x^{(0)}, y^{(0)})$ has centroid $(0, 0)$.
2. We scale the vertex vectors after each update so that they have unit 2-norm.

These adjustments have the effect of keeping the polygon sequence $\{\mathcal{P}_k\}$ reasonably sized and centered around the origin.

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

From the remarks made in section 1.2, we know that each $\mathcal{P}_k$ has centroid $(0, 0)$. The 2-norm scaling has no effect in this regard.

It is important to recognize that Algorithm 2 amounts to a double application of the power method, one of the most basic iterations in the field of matrix computations [2, p. 330]. Applied to an $n \times n$ matrix A and a unit 2-norm starting vector $w^{(0)}$, the power method involves repeated matrix-vector multiplication and scaling:

$$w^{(k)} = A w^{(k-1)} \, / \, \| A w^{(k-1)} \|_2, \qquad k = 1, 2, \ldots .$$

It is important to note that the scaling need not be done at every iteration, and can theoretically be done at termination. The power method is ordinarily used to compute an eigenvector for A associated with A's largest eigenvalue in absolute value. To see why it works, assume for clarity that A has a full set of n independent eigenvectors $z_1, \ldots, z_n$ and eigenvalues $\lambda_1, \ldots, \lambda_n$ with $A z_i = \lambda_i z_i$. If

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$$

and

$$w^{(0)} = \gamma_1 z_1 + \gamma_2 z_2 + \cdots + \gamma_n z_n \, ,$$

then $w^{(k)}$ is a unit vector in the direction of

$$A^k w^{(0)} = \lambda_1^k \left(\gamma_1 z_1 + \sum_{i=2}^{n} \gamma_i \left(\frac{\lambda_i}{\lambda_1} \right)^k z_i \right). \tag{1.2}$$

Notice that this vector is increasingly rich in the direction of z_1 provided $\gamma_1 \neq 0$. The eigenvector z_1 is referred to as a *dominant eigenvector* because it is associated with the *dominant eigenvalue* λ_1.

The dominant eigenvalue of the matrix M_n is 1 corresponding to the dominant eigenvector e. To see this observe that $Me = e$, so 1 is an eigenvalue with eigenvector e. No eigenvalue of M_n can be larger than its 1-norm, and since

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$$

it follows that $\|M_n\|_1 = 1$. Thus, 1 is a dominant eigenvalue. It is also unique, as we show below.

These observations shed light on why the polygons in Algorithm 1 converge to a point. Both sequences of vertex vectors are increasingly rich in the direction of e, which is tantamount to saying that the components of $x^{(k)}$ and $y^{(k)}$ are increasingly uniform. Since the iteration preserves centroids, the limiting values are the centroids of the initial vertex vectors $x^{(0)}$ and $y^{(0)}$, respectively.

1.4. From Experimentation to Conjecture. If we experiment with Algorithm 2 and display the $\mathcal{P}_k$, then we discover the following surprising phenomenon.

> CONJECTURE 1. *No matter how random the initial $\mathcal{P}_0$, the edges of the $\mathcal{P}_k$ eventually "uncross," and in the limit the vertices appear to arrange themselves around an ellipse that is tilted 45 degrees from the coordinate axes.*

See Figure 1.3, which traces a typical $n = 20$ example. Our goal is to explain the apparent transition from "chaos" to "order." Table 1.2 reports how many iterations are required (on average) before the polygon "untangles." Because of the power method connection, the explanation revolves around the eigensystem properties of the averaging matrix M_n. The vertex vector iterates $x^{(k)}$ and $y^{(k)}$ in Algorithm 2 have centroid zero and are therefore orthogonal to M_n's dominant eigenvector e. In

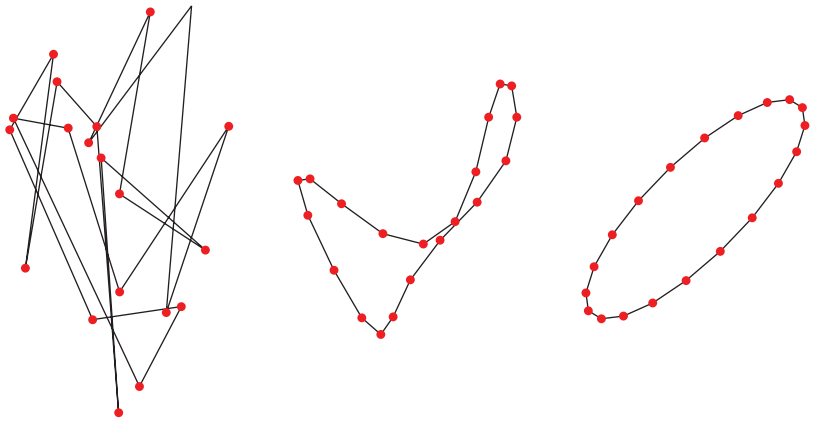

Fig. 1.3 *The progression from $\mathcal{P}_0$ to $\mathcal{P}_{18}$ to $\mathcal{P}_{200}$ for an $n = 20$ example.*

Table 1.2 *k_u is the smallest k so that the edges of $\mathcal{P}(x^{(k)}, y^{(k)})$ do not cross. For each n, the averages were estimated from 100 random examples.*

n	Average k_u
10	8.7
20	38.1
40	163.0
80	647.5

the notation of (1.2), the γ_1 term for both $x^{(k)}$ and $y^{(k)}$ is missing. Thus the polygons in Algorithm 2 will not converge to a single point. Instead, these vectors converge to a very special two-dimensional invariant subspace which we identify in section 2. Experimentation reveals that within this subspace the sequence of vertex vectors is cyclic. We explain this in section 3 and go on to show in section 4 that the vertices $(x_i^{(k)}, y_i^{(k)})$ converge to an ellipse $\mathcal{E}$ having a 45-degree tilt. The semiaxes of $\mathcal{E}$ are specified in terms of a 2-by-2 singular value decomposition (SVD) related to the initial vertex vectors. Concluding remarks are offered in section 5.

2. The Subspace $\mathcal{D}_2$. The behavior of the vertex vectors $x^{(k)}$ and $y^{(k)}$ in Algorithm 2 requires a thorough understanding of the eigensystem of the averaging matrix M_n. Fortunately, the eigenvalues and eigenvectors of this matrix can be specified in closed form.

2.1. The Upshift Matrix. Define the n-by-n *upshift matrix* S_n by

$$S_n \;=\; \left[\; e_n \;\middle|\; e_1 \;\middle|\; e_2 \;\middle|\; \cdots \;\middle|\; e_{n-1} \;\right], \tag{2.1}$$

where e_k is the kth column of the n-by-n identity matrix I_n, e.g.,

$$S_5 \;=\; \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

From (1.1) we see that the averaging matrix M_n is given by

$$M_n \;=\; \frac{1}{2}\left(I_n + S_n\right). \tag{2.2}$$

The eigensystem of S_n is completely known and involves the nth roots of unity:

$$\omega_j = \cos\left(\frac{2\pi j}{n}\right) \;+\; i \cdot \sin\left(\frac{2\pi j}{n}\right), \qquad j = 0{:}n-1.$$

Using the fact that $\omega_j^n = 1$, it is easy to verify for $j = 0{:}n-1$ that

$$v_j \;=\; \sqrt{1/n}\begin{bmatrix} 1 \\ \omega_j \\ \omega_j^2 \\ \vdots \\ \omega_j^{n-1} \end{bmatrix}$$

has unit 2-norm and satisfies $S_n v_j = \omega_j v_j$. Moreover, $v_0, \ldots, v_{n-1}$ are mutually orthogonal. To see this we observe that

$$v_i^H S_n v_j = v_i^H (S_n v_j) = \omega_j v_i^H v_j.$$

On the other hand, since $S_n^H = S_n^T = S_n^{-1}$ we also have

$$v_i^H S_n v_j = (v_i^H S_n) v_j = (S_n^{-1} v_i)^H v_j = (v_i/\omega_i)^H v_j = \omega_i v_i^H v_j.$$

It follows that $v_i^H v_j = 0$ if $i \neq j$. The "H" superscript indicates the Hermitian transpose.

2.2. The Eigenvalues and Eigenvectors of M_n. The averaging matrix $M_n = (I_n + S_n)/2$ has the same eigenvectors as S_n. Its eigenvalues $\lambda_1, \ldots, \lambda_n$ are given by

$$\begin{aligned}
\lambda_1 &= (1+\omega_0)/2 = 1,\\
\lambda_2 &= (1+\omega_1)/2 = (1+\cos(2\pi/n) + i\cdot\sin(2\pi/n))/2,\\
\lambda_3 &= (1+\omega_{n-1})/2 = \bar{\lambda}_2,\\
\lambda_4 &= (1+\omega_2)/2 = (1+\cos(4\pi/n) + i\cdot\sin(4\pi/n))/2,\\
\lambda_5 &= (1+\omega_{n-2})/2 = \bar{\lambda}_4,\\
&\vdots \qquad\qquad \vdots\\
\lambda_n &= (1+\omega_m)/2,
\end{aligned} \tag{2.3}$$

where $m = \text{floor}(n/2)$. We have chosen to order the eigenvalues this way because it groups together complex conjugate eigenvalue pairs. As illustrated in Figure 2.1, the λ_i are on a circle in the complex plane that has center $(0.5, 0.0)$ and diameter one. Moreover,

$$|\lambda_1| > |\lambda_2| = |\lambda_3| > \cdots > |\lambda_{n-1}| \geq |\lambda_n|. \tag{2.4}$$

Let $z_1, \ldots, z_n$ be a reordering of the eigenvectors $v_0, \ldots, v_{n-1}$ so that

$$M_n z_k = \lambda_k z_k, \qquad k = 1{:}n. \tag{2.5}$$

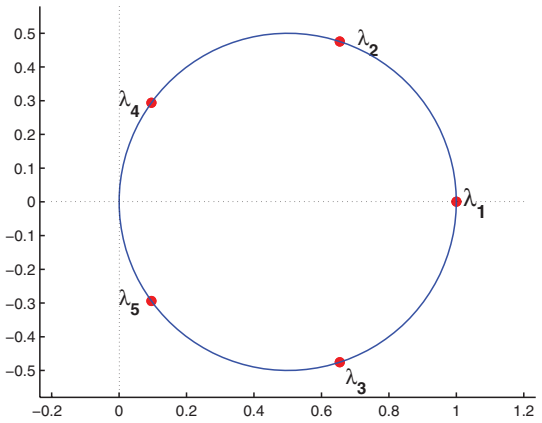

Fig. 2.1 *The eigenvalues of M_5.*

The behavior of the vertex vectors $x^{(k)}$ and $y^{(k)}$ depends upon the ratio $|\lambda_2/\lambda_4|^k$ and the projection of $x^{(0)}$ and $y^{(0)}$ onto the span of z_2 and z_3. We proceed to make this precise by establishing a number of key facts.

2.3. The Damping Factor. A unit vector $w \in \mathbf{C}^n$ whose components sum to zero is said to have centroid zero. It has an eigenvector expansion of the form

$$w \;=\; \gamma_2 z_2 \;+\; \cdots \;+\; \gamma_n z_n,$$

where $|\gamma_2|^2 \;+\; \cdots \;+\; |\gamma_n|^2 \;=\; 1$. This is because $\{z_1, \ldots, z_n\}$ is an orthonormal basis, z_1 is a multiple of e, and a centroid zero vector w satisfies $e^T w = 0$. It follows that

$$M_n^k w \;=\; |\lambda_2|^k \left(\gamma_2 \left(\frac{\lambda_2}{|\lambda_2|} \right)^k z_2 \;+\; \gamma_3 \left(\frac{\lambda_3}{|\lambda_2|} \right)^k z_3 \;+\; \sum_{j=4}^{n} \gamma_j \left(\frac{\lambda_j}{|\lambda_2|} \right)^k z_j \right). \tag{2.6}$$

From the assumption (2.4) we see that the vectors $M_n^k w$ are increasingly rich in z_2 and z_3. The rate at which the components in directions $z_4, \ldots, z_n$ are damped clearly depends upon the quotient

$$\rho_n \;\equiv\; \left| \frac{\lambda_4}{\lambda_2} \right| \;=\; \max \left\{ \left| \frac{\lambda_4}{\lambda_2} \right|, \ldots, \left| \frac{\lambda_n}{\lambda_2} \right| \right\}. \tag{2.7}$$

Since

$$\left| \frac{\lambda_4}{\lambda_2} \right|^2 \;=\; \frac{|1 + \cos(4\pi/n) + i \sin(4\pi/n)|^2}{|1 + \cos(2\pi/n) + i \sin(2\pi/n)|^2} \;=\; \frac{1 + \cos(4\pi/n)}{1 + \cos(2\pi/n)} \;=\; \frac{\cos^2(2\pi/n)}{\cos^2(\pi/n)},$$

it follows that

$$\rho_n \;=\; \left| \frac{\lambda_4}{\lambda_2} \right| \;=\; \frac{\cos^2(\pi/n) - \sin^2(\pi/n)}{\cos(\pi/n)} \;=\; \cos(\pi/n) \left(1 - \tan^2(\pi/n) \right).$$

Selected values of ρ_n are given in Table 2.1. By manipulating the approximations

$$\cos(\pi/n) \;=\; 1 - \frac{1}{2} \left(\frac{\pi}{n} \right)^2 \;+\; O\left(\frac{1}{n^4} \right)$$

and

$$\cos(2\pi/n) \;=\; 1 - \frac{1}{2} \left(\frac{2\pi}{n} \right)^2 \;+\; O\left(\frac{1}{n^4} \right)$$

it can be shown that

$$\rho_n \;=\; 1 \;-\; \frac{3}{2} \left(\frac{\pi}{n} \right)^2 \;+\; O\left(\frac{1}{n^4} \right).$$

Thus, the damping proceeds much more slowly for large n, a point already conveyed by Table 2.1.

Table 2.1 *The damping factor ρ_n.*

n	ρ_n
5	0.3820
10	0.8507
20	0.9629
30	0.9835
40	0.9907
50	0.9941
100	0.9985

2.4. Real Versus Complex Arithmetic. It is important to appreciate the interplay between real and complex arithmetic in (2.6). The left-hand-side vector $M_n^k w$ is real but the linear combinations on the right-hand side involve complex eigenvectors. However, the eigenvectors and eigenvalues of a real matrix such as M_n come in conjugate pairs, so the imaginary parts cancel out in the summation. In particular, the "undamped" vector

$$\tilde{w}^{(k)} \;=\; \gamma_2 \left(\frac{\lambda_2}{|\lambda_2|}\right)^k z_2 \;+\; \gamma_3 \left(\frac{\lambda_3}{|\lambda_2|}\right)^k z_3$$

in (2.6) is real because $\gamma_3 = \bar{\gamma}_2$, $\lambda_3 = \bar{\lambda}_2$, and $z_3 = \bar{z}_2$. Moreover, it belongs to the real subspace $\mathcal{D}_2$ defined by

$$\mathcal{D}_2 \;=\; \mathrm{span}\{\mathrm{Re}(z_2), \mathrm{Im}(z_2)\} \;=\; \mathrm{span}\{\mathrm{Re}(z_3), \mathrm{Im}(z_3)\}. \tag{2.8}$$

This subspace is an invariant subspace of M_n. To see this, we compare the real and imaginary parts of the eigenvector equation $M_n z_2 = \lambda_2 z_2$:

$$M_n \left(\mathrm{Re}(z_2) + i \cdot \mathrm{Im}(z_2)\right) \;=\; \left(\cos(2\pi/n) + i \cdot \sin(2\pi/n)\right) \left(\mathrm{Re}(z_2) + i \cdot \mathrm{Im}(z_2)\right).$$

It is clear that $M_n \cdot \mathrm{Re}(z_2) \in \mathcal{D}_2$ and $M_n \cdot \mathrm{Im}(z_2) \in \mathcal{D}_2$.

2.5. Convergence to $\mathcal{D}_2$. The next result characterizes the convergence of the power method when it is applied with the matrix M_n and a starting vector whose components sum to zero.

THEOREM 2.1. *Suppose*

$$w^{(0)} \;=\; \sum_{j=2}^{n} \gamma_j z_j$$

is a real unit 2*-norm vector and*

$$\mu = \sqrt{|\gamma_2|^2 + |\gamma_3|^2} \;>\; 0.$$

If P_n is the orthogonal projection onto span$\{z_2, z_3\}^{\perp}$ and $w^{(k)}$ is a unit 2*-norm vector in the direction of $M_n^k w^{(0)}$, then*

$$\| P_n w^{(k)} \|_2 \;\le\; \rho_n^k \, \frac{\sqrt{1-\mu^2}}{\mu}.$$

Proof. Since the eigenvector set $\{z_2, \ldots, z_n\}$ is orthonormal, it follows from (2.6) that

$$w^{(k)} \;=\; \left(\sum_{j=2}^{n} \gamma_j \lambda_j^k z_j\right) \Bigg/ \sqrt{\sum_{j=2}^{n} |\gamma_j|^2 |\lambda_j|^{2k}}.$$

The projection $P_n w^{(k)}$ has the same form without the z_2 and z_3 terms in the numerator. Thus,

$$\| P_n w^{(k)} \|_2^2 \;=\; \frac{\displaystyle\sum_{j=4}^{n} |\gamma_j|^2 |\lambda_j|^{2k}}{\displaystyle\sum_{j=2}^{n} |\gamma_j|^2 |\lambda_j|^{2k}} \le \frac{|\lambda_4|^{2k} \displaystyle\sum_{j=4}^{n} |\gamma_j|^2}{|\lambda_2|^{2k}(|\gamma_2|^2 + |\gamma_3|^2)} = \rho_n^{2k} \, \frac{1-\mu^2}{\mu^2},$$

where we used (2.7), $|\lambda_2| = |\lambda_3|$, and $|\gamma_2|^2 + \cdots + |\gamma_n|^2 \;=\; 1$. ∎

The quantity $\| P_n w^{(k)} \|_2$ can be regarded as the distance from $w^{(k)}$ to $\mathcal{D}_2$. It is easy to visualize this quantity for the case $n = 3$. The subspace $\mathcal{D}_2$ is a plane, $w^{(k)}$ points out of the plane, and $\| P_n w^{(k)} \|_2$ is the length of the dropped perpendicular.

We mention that in exact arithmetic, the vertex vectors in Algorithm 2 have centroid zero. However, roundoff error gradually changes this. Thus, if n is large, then it makes sense to "recenter" the vertex vectors every so often, e.g.,

$$x^{(k)} \leftarrow x^{(k)} - (e^T x^{(k)}/n)e \, .$$

Another issue associated with Algorithm 2 is the possibility that the initial vertex vectors are orthogonal to $\mathcal{D}_2$. In this case the vertex vectors converge to the invariant subspace spanned by $\mathrm{Re}(z_4)$ and $\mathrm{Im}(z_4)$. The cosine/sine evaluations in these vectors are spaced twice as far apart as the cosine/sine evaluations in $\mathrm{Re}(z_2)$ and $\mathrm{Im}(z_2)$. It is observed that the vertices in $\mathcal{P}_k$ undergo a pairwise coalescence as k gets large, although they still converge to a 45-degree ellipse. If the initial vertex vectors are only orthogonal to one of $\mathrm{Re}(z_2)$ and $\mathrm{Im}(z_2)$, then the polygon converges to a 45-degree line. If only one of the initial vertex vectors is orthogonal to $\mathcal{D}_2$, then the polygon converges to a curve that contains two ellipse-like regions. Finally, if one vertex vector is initially orthogonal to $\mathrm{Re}(z_2)$ and the other is the orthogonal to $\mathrm{Im}(z_2)$, then the vertices converge to a circle. We leave the analysis of these interesting degenerate situations to the reader and recommend using an ellipse fitter [2] for computational tests. For the remainder of the paper we will assume that neither vertex vector is orthogonal to the subspace $\mathcal{D}_2$.

2.6. A Real Orthonormal Basis for $\mathcal{D}_2$. The real and imaginary parts of the eigenvector z_2 (and its conjugate z_3) are highly structured. If

$$\tau = \begin{bmatrix} 0 \\ 2\pi/n \\ 4\pi/n \\ \vdots \\ 2(n-1)\pi/n \end{bmatrix}, \tag{2.9}$$

then in vector notation,[1] $z_2 = (\cos(\tau) + i \cdot \sin(\tau))/\sqrt{n}$. Using elementary trigonometric identities, it is easy to show that

$$\cos(\tau)^T \cos(\tau) = \sum_{j=1}^{n} \cos(\tau_j)^2 = \sum_{j=1}^{n} (1 + \cos(2\tau_j))/2 = n/2 \, ,$$

$$\sin(\tau)^T \sin(\tau) = \sum_{j=1}^{n} \sin(\tau_j)^2 = \sum_{j=1}^{n} (1 - \cos(2\tau_j))/2 = n/2 \, ,$$

and

$$\sin(\tau)^T \cos(\tau) = \sum_{j=1}^{n} \sin(\tau_j)\cos(\tau_j) = \sum_{j=1}^{n} \sin(2\tau_j)/2 = 0 \, .$$

It follows from (2.8) that the vectors

$$c = \sqrt{2/n} \begin{bmatrix} \cos(\tau_1) \\ \cos(\tau_2) \\ \vdots \\ \cos(\tau_n) \end{bmatrix}, \qquad s = \sqrt{2/n} \begin{bmatrix} \sin(\tau_1) \\ \sin(\tau_2) \\ \vdots \\ \sin(\tau_n) \end{bmatrix} \tag{2.10}$$

form a real orthonormal basis for $\mathcal{D}_2$.

[1] If f is a function defined on the components of $\tau \in \mathbb{R}^n$, then the ith component of $f(\tau)$ is $f(\tau_i)$.

3. Tracking the Vertex Vectors. Because they are orthogonal to z_1 (a multiple of the vector of all ones) and we assume that they are not orthogonal to $\mathcal{D}_2$, the initial vertex vectors $x^{(0)}$ and $y^{(0)}$ in Algorithm 2 can be expressed as a linear combination of the orthonormal vectors $\{c,\ s,\ z_4, \ldots,\ z_n\}$:

$$\begin{aligned} x^{(0)} &= \alpha_1 c \ + \ \alpha_2 s \ + \ \text{vector in span}\{z_4, \ldots, z_n\}, \\ y^{(0)} &= \beta_1 c \ + \ \beta_2 s \ + \ \text{vector in span}\{z_4, \ldots, z_n\}. \end{aligned} \tag{3.1}$$

It follows from Theorem 2.1 that for large k

$$\begin{aligned} x^{(k)} &= u^{(k)} \ + \ O(\rho_n^k), \\ y^{(k)} &= v^{(k)} \ + \ O(\rho_n^k), \end{aligned} \tag{3.2}$$

where $u^{(k)}$ and $v^{(k)}$ are the unit vectors

$$\begin{aligned} u^{(k)} &= \frac{\alpha_1 M_n^k c + \alpha_2 M_n^k s}{\| \alpha_1 M_n^k c + \alpha_2 M_n^k s \|_2}, \\ v^{(k)} &= \frac{\beta_1 M_n^k c + \beta_2 M_n^k s}{\| \beta_1 M_n^k c + \beta_2 M_n^k s \|_2}. \end{aligned} \tag{3.3}$$

Note that these vectors are in the subspace $\mathcal{D}_2$. Our plan is to study the polygon sequence $\{\mathcal{P}(u^{(k)}, v^{(k)})\}$ since its limiting behavior coincides with limiting behavior of the polygon sequence $\{\mathcal{P}(x^{(k)}, y^{(k)})\}$.

3.1. An Experiment and Another Conjecture. For clarity we reproduce Algorithm 2 for the case when the unit 2-norm starting vectors are in $\mathcal{D}_2$.

ALGORITHM 3.

Input: Real numbers θ_u and θ_v.

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

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

Display $\mathcal{P}_0 = \mathcal{P}(u^{(0)}, v^{(0)})$.

for $k = 1, 2, \ldots$

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

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

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

Display $\mathcal{P}_k$.

end

Experimenting with this iteration leads to a second conjecture.

CONJECTURE 2. *For any input values θ_u and θ_v in Algorithm 3, the even-indexed polygons $\mathcal{P}_{2k}$ are all the same and the odd-indexed polygons $\mathcal{P}_{2k+1}$ are all the same.*

Figure 3.1 displays "$\mathcal{P}_{even}$" and "$\mathcal{P}_{odd}$" for a typical $n = 5$ example. What does this

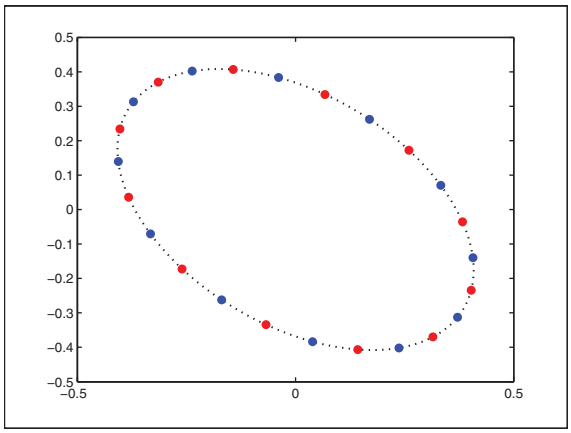

Fig. 3.1 *The vertices of $\mathcal{P}_{even}$ (red) and $\mathcal{P}_{odd}$ (blue) ($n = 12$).*

imply about the vertex vectors? Displaying the sequence $\{u^{(k)}\}$ (or $\{v^{(k)}\}$) reveals the following progression:

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} \to \begin{bmatrix} a' \\ b' \\ c' \\ d' \\ e' \end{bmatrix} \to \begin{bmatrix} e \\ a \\ b \\ c \\ d \end{bmatrix} \to \begin{bmatrix} e' \\ a' \\ b' \\ c' \\ d' \end{bmatrix} \to \begin{bmatrix} d \\ e \\ a \\ b \\ c \end{bmatrix} \to \begin{bmatrix} d' \\ e' \\ a' \\ b' \\ c' \end{bmatrix} \to \begin{bmatrix} c \\ d \\ e \\ a \\ b \end{bmatrix} \to \text{etc.}$$

In other words, the vertex vectors associated with $\mathcal{P}_k$ are upshifted versions of the vertex vectors associated with $\mathcal{P}_{k-2}$. More precisely,

$$\begin{aligned} \mathcal{P}_{2k} &= \mathcal{P}(S_n^k u^{(0)}, S_n^k v^{(0)}) = \mathcal{P}(u^{(0)}, v^{(0)}), \\ \mathcal{P}_{2k+1} &= \mathcal{P}(S_n^k u^{(1)}, S_n^k v^{(1)}) = \mathcal{P}(u^{(1)}, v^{(1)}). \end{aligned} \tag{3.4}$$

Proving this result is an exercise in sine/cosine manipulation, as we now show. Readers with an aversion to trigonometric identities are advised to skip to section 4.1!

3.2. Repeated Averaging of the Vectors *c* and *s*. We need to characterize the vectors $M_n^k c$ and $M_n^k s$ since it is clear from Algorithm 3 that both $u^{(k)}$ and $v^{(k)}$ are in their span. Towards that end we define the n-vectors $\mathbf{c}(\tau + \Delta)$ and $\mathbf{s}(\tau + \Delta)$ by

$$\mathbf{c}(\tau + \Delta) \;=\; \sqrt{2/n} \begin{bmatrix} \cos(\tau_1 + \Delta) \\ \vdots \\ \cos(\tau_n + \Delta) \end{bmatrix} \tag{3.5}$$

and

$$\mathbf{s}(\tau + \Delta) \;=\; \sqrt{2/n} \begin{bmatrix} \sin(\tau_1 + \Delta) \\ \vdots \\ \sin(\tau_n + \Delta) \end{bmatrix}, \tag{3.6}$$

where $\Delta \in \mathbb{R}$ and $\tau \in \mathbb{R}^n$ is given by (2.9). The notation $\tau + \Delta$ designates the vector obtained by adding the scalar Δ to each component of the vector τ.

Note that if $\Delta = 0$, then $\mathbf{c}(\tau + \Delta) = c$ and $\mathbf{s}(\tau + \Delta) = s$, where c and s are defined by (2.10). Moreover, $\mathbf{c}(\tau + \Delta)$ and $\mathbf{s}(\tau + \Delta)$ each have unit 2-norm and are orthogonal to each other. This follows from

$$\mathbf{c}(\tau + \Delta) = \cos(\Delta)c - \sin(\Delta)s,$$
$$\mathbf{s}(\tau + \Delta) = \sin(\Delta)c + \cos(\Delta)s,$$

and section 2.6, where we showed that c and s have unit 2-norm and are orthogonal to each other.

Define $\tau_{n+1} = \tau_1 = 0$. If $\mu_i = \tau_i + \Delta + \pi/n$ for $i = 1{:}n$, then using the identities

$$\cos(\alpha + \beta) + \cos(\alpha - \beta) \;=\; 2\cos(\alpha)\cos(\beta), \tag{3.7}$$

$$\sin(\alpha + \beta) + \sin(\alpha - \beta) \;=\; 2\sin(\alpha)\cos(\beta) \tag{3.8}$$

we have

$$\begin{aligned}
\left(M_n \mathbf{c}(\tau + \Delta)\right)_i &= \frac{1}{2}\sqrt{\frac{2}{n}}\ \left(\cos(\tau_i + \Delta) + \cos(\tau_{i+1} + \Delta)\right) \\
&= \frac{1}{2}\sqrt{\frac{2}{n}}\ \left(\cos(\mu_i - \pi/n) + \cos(\mu_i + \pi/n)\right) \\
&= \sqrt{\frac{2}{n}}\ \cos\left(\frac{\pi}{n}\right)\cos(\mu_i)
\end{aligned}$$

and

$$\begin{aligned}
\left(M_n \mathbf{s}(\tau + \Delta)\right)_i &= \frac{1}{2}\sqrt{\frac{2}{n}}\ \left(\sin(\tau_i + \Delta) + \sin(\tau_{i+1} + \Delta)\right) \\
&= \frac{1}{2}\sqrt{\frac{2}{n}}\ \left(\sin(\mu_i - \pi/n) + \sin(\mu_i + \pi/n)\right) \\
&= \sqrt{\frac{2}{n}}\ \cos\left(\frac{\pi}{n}\right)\sin(\mu_i).
\end{aligned}$$

Thus,

$$M_n\, \mathbf{c}(\tau + \Delta) \;=\; \cos\left(\frac{\pi}{n}\right) \mathbf{c}\left(\tau + \Delta + \frac{\pi}{n}\right),$$
$$M_n\, \mathbf{s}(\tau + \Delta) \;=\; \cos\left(\frac{\pi}{n}\right) \mathbf{s}\left(\tau + \Delta + \frac{\pi}{n}\right),$$

and so by induction we have

$$M_n^k\, c \;=\; \cos\left(\frac{\pi}{n}\right)^k \mathbf{c}\left(\tau + \frac{k\pi}{n}\right),$$
$$M_n^k\, s \;=\; \cos\left(\frac{\pi}{n}\right)^k \mathbf{s}\left(\tau + \frac{k\pi}{n}\right).$$

From Algorithm 3 we see that $u^{(k)}$ and $v^{(k)}$ are unit 2-norm vectors in the direction of

$$M_n^k u^{(0)} = \cos\left(\frac{\pi}{n}\right)^k \left(\cos(\theta_u)\mathbf{c}\left(\tau + \frac{k\pi}{n}\right) + \sin(\theta_u)\mathbf{s}\left(\tau + \frac{k\pi}{n}\right)\right),$$
$$M_n^k v^{(0)} = \cos\left(\frac{\pi}{n}\right)^k \left(\cos(\theta_v)\mathbf{c}\left(\tau + \frac{k\pi}{n}\right) + \sin(\theta_v)\mathbf{s}\left(\tau + \frac{k\pi}{n}\right)\right).$$

Since $\{\mathbf{c}, \mathbf{s}\}$ is an orthonormal set, we conclude that

$$u^{(k)} = \cos(\theta_u)\mathbf{c}\left(\tau + \frac{k\pi}{n}\right) + \sin(\theta_u)\mathbf{s}\left(\tau + \frac{k\pi}{n}\right), \tag{3.9}$$

$$v^{(k)} = \cos(\theta_v)\mathbf{c}\left(\tau + \frac{k\pi}{n}\right) + \sin(\theta_v)\mathbf{s}\left(\tau + \frac{k\pi}{n}\right). \tag{3.10}$$

From the definition of τ, $\mathbf{c}$, and $\mathbf{s}$, it follows that

$$u^{(k+2)} = S_n u^{(k)}, \qquad v^{(k+2)} = S_n v^{(k)}. \tag{3.11}$$

Together with (3.4), this confirms Conjecture 2.

4. The Limiting Ellipse. We now show that for all i and k, the points $(u_i^{(k)}, v_i^{(k)})$ in Algorithm 3 are on the same ellipse $\mathcal{E}$ and that $\mathcal{E}$ has a 45-degree tilt. We refer to this limiting ellipse as the $\mathcal{D}_2$ *ellipse* since it depends upon the vertex vector projections into that invariant subspace.

4.1. Tilted Ellipses. As t ranges over all real values, the set of points $(u(t), v(t))$ given by

$$\begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix} \tag{4.1}$$

define an ellipse with center $(0, 0)$, tilt ϕ, and semiaxes $|\sigma_1|$ and $|\sigma_2|$. It is obtained by rotating the ellipse

$$\left(\frac{u}{\sigma_1}\right)^2 + \left(\frac{v}{\sigma_2}\right)^2 = 1$$

counterclockwise ϕ radians. See Figure 4.1.

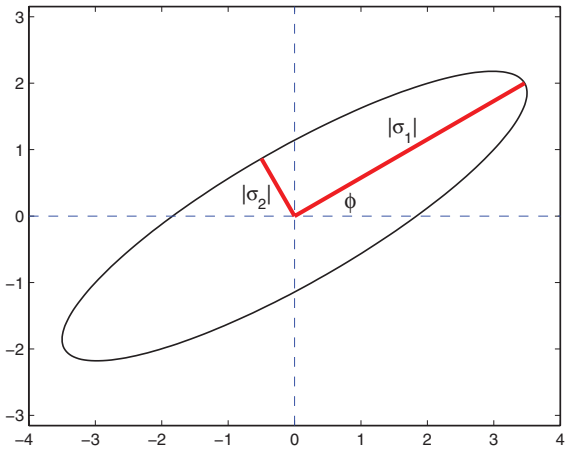

Fig. 4.1 *A tilted ellipse.*

The set of points $(u(t), v(t))$ defined by

$$\begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix} \equiv A \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix} \tag{4.2}$$

is also an ellipse with center $(0, 0)$. Its semiaxes and tilt are specified by the SVD of A. The SVD of a real n-by-n matrix A guarantees that we can find orthogonal U and V such that $A = U\text{diag}(\sigma_1, \ldots, \sigma_n)V^T$. Usually the transformation matrices U and V are chosen so that the σ_i are nonnegative and ordered. However, in our ellipse application we have no need for that normalization. Specialized to the 2-by-2 case, the SVD states that we can find ϕ and ψ so that

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}^T.$$

It follows that the ellipse (4.2) has semiaxes $|\sigma_1|$ and $|\sigma_2|$ and tilt ϕ. Note that it is the U matrix that specifies the tilt.

4.2. The $\mathcal{D}_2$ Ellipse. If $t_i = \tau_i + k\pi/n$ for $i = 1{:}n$ and the vertex vectors $u^{(k)}$ and $v^{(k)}$ are generated by Algorithm 3, then from (3.5), (3.6), (3.9), and (3.10) we have

$$\begin{bmatrix} u_i^{(k)} \\ v_i^{(k)} \end{bmatrix} = \sqrt{\frac{2}{n}} \begin{bmatrix} \cos(\theta_u) & \sin(\theta_u) \\ \cos(\theta_v) & \sin(\theta_v) \end{bmatrix} \begin{bmatrix} \cos(t_i) \\ \sin(t_i) \end{bmatrix}.$$

This shows that the vertices of the polygon $\mathcal{P}(u^{(k)}, v^{(k)})$ are on the ellipse $\mathcal{E}$ defined by the set of all $(u(t), v(t))$ where

$$\begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \sqrt{\frac{2}{n}} \begin{bmatrix} \cos(\theta_u) & \sin(\theta_u) \\ \cos(\theta_v) & \sin(\theta_v) \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}, \qquad -\infty < t < \infty. \tag{4.3}$$

To specify the tilt and semiaxes of $\mathcal{E}$, we need the SVD of the 2-by-2 transformation matrix.

THEOREM 4.1. *If θ_u and θ_v are real numbers and*

$$A = \begin{bmatrix} \cos(\theta_u) & \sin(\theta_u) \\ \cos(\theta_v) & \sin(\theta_v) \end{bmatrix},$$

then $U^T AV = \Sigma$, where

$$U = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix}, \quad V = \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix},$$

and

$$\Sigma = \begin{bmatrix} \sqrt{2}\,\cos(b) & 0 \\ 0 & \sqrt{2}\,\sin(b) \end{bmatrix}$$

with

$$a = \frac{\theta_v + \theta_u}{2}, \qquad b = \frac{\theta_v - \theta_u}{2}.$$

Proof. Using the trigonometric identities (3.7) and (3.8) we have

$$U^T A = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos(\theta_v) + \cos(\theta_u) & \sin(\theta_v) + \sin(\theta_u) \\ \cos(\theta_v) - \cos(\theta_u) & \sin(\theta_v) - \sin(\theta_u) \end{bmatrix}$$
$$= \sqrt{2} \begin{bmatrix} \cos(b)\cos(a) & \cos(b)\sin(a) \\ -\sin(b)\sin(a) & \sin(b)\cos(a) \end{bmatrix}.$$

But this is precisely ΣV^T. ∎

Using this result, it follows that the $\mathcal{D}_2$ ellipse (4.3) has a 45-degree tilt and semiaxes $|\sigma_1|$ and $|\sigma_2|$, where

$$\sigma_1 = \frac{2}{\sqrt{n}} \cos\left(\frac{\theta_v - \theta_u}{2}\right), \qquad \sigma_2 = \frac{2}{\sqrt{n}} \sin\left(\frac{\theta_v - \theta_u}{2}\right). \tag{4.4}$$

This confirms Conjecture 1.

4.3. Specification via $x^{(0)}$ and $y^{(0)}$. We now specify the limiting ellipse associated with Algorithm 2 in terms of the initial vertex vectors $x^{(0)}$ and $y^{(0)}$. From (3.1) and the orthonormality of $\{c, s, z_4, \ldots, z_n\}$ we know that

$$x^{(0)} = (c^T x^{(0)})c \ + \ (s^T x^{(0)})s \ + \ \text{vector in span}\{z_4, \ldots, z_n\},$$
$$y^{(0)} = (c^T y^{(0)})c \ + \ (c^T y^{(0)})s \ + \ \text{vector in span}\{z_4, \ldots, z_n\}.$$

Referring to (3.2) and (3.3), because all but the c and s terms are damped out, the limiting ellipse for Algorithm 2 is the same as the limiting ellipse for Algorithm 3 with starting unit vectors

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

where

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

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

These four numbers together with (4.4) completely specify the 45-degree tilted ellipse associated with Algorithm 2.

4.4. Alternative Normalizations. It is natural to ask what happens to the normalized polygon averaging process if we use an alternative to the 2-norm for normalization. A change in norm will only affect the size of the vertex vectors, not their direction. Since vertex vector convergence to $\mathcal{D}_2$ remains in force, we can explore the limiting effect of different normalizations by considering the following generalization of Algorithm 3.

ALGORITHM 4.
Input: Vector norms $\|\cdot\|_x$ *and* $\|\cdot\|_y$ *and real* n*-vectors* $\tilde{u}^{(0)}$ *and* $\tilde{v}^{(0)}$ *belonging to* $\mathcal{D}_2$ *that satisfy* $\|\tilde{u}^{(0)}\|_x = \|\tilde{v}^{(0)}\|_y = 1$.

for $k = 1, 2, \ldots$

% Compute $\tilde{\mathcal{P}}_k = \mathcal{P}(\tilde{u}^{(k)}, \tilde{v}^{(k)})$ from $\tilde{\mathcal{P}}_{k-1} = \mathcal{P}(\tilde{u}^{(k-1)}, \tilde{v}^{(k-1)})$

$f = M_n \tilde{u}^{(k-1)}$, $\tilde{u}^{(k)} = f/\|f\|_x$

$g = M_n \tilde{v}^{(k-1)}$, $\tilde{v}^{(k)} = g/\|g\|_y$

end

If the initial vertex vectors in Algorithm 4 are multiples of the initial vertex vectors in Algorithm 3, then it follows that for all k

$$\begin{aligned} \tilde{u}^{(k)} &= u^{(k)} \,/\, \|u^{(k)}\|_x, \\ \tilde{v}^{(k)} &= v^{(k)} \,/\, \|v^{(k)}\|_x. \end{aligned} \tag{4.5}$$

Therefore, our task is to examine the polygon sequence

$$\tilde{\mathcal{P}}_k = \mathcal{P}(\tilde{u}^{(k)}, \tilde{v}^{(k)}) = \mathcal{P}(\alpha_k u^{(k)}, \beta_k v^{(k)}),$$

where

$$\begin{aligned} \alpha_k &= 1/\|u^{(k)}\|_x, \\ \beta_k &= 1/\|v^{(k)}\|_y. \end{aligned} \tag{4.6}$$

Since the vertices of $\mathcal{P}(u^{(k)}, v^{(k)})$ are on an ellipse $\mathcal{E}$ of the form

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix},$$

it follows that the vertices of $\tilde{\mathcal{P}}_k$ are on the ellipse $\tilde{\mathcal{E}}_k$ defined by

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} \alpha_i & 0 \\ 0 & \beta_i \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}.$$

Note that if

$$Z_k = \begin{bmatrix} \alpha_k \cos(\phi)\sigma_1 & -\alpha_k \sin(\phi)\sigma_2 \\ \beta_k \sin(\phi)\sigma_1 & \beta_k \cos(\phi)\sigma_2 \end{bmatrix}, \tag{4.7}$$

then

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = Z \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}.$$

From our SVD/ellipse remarks in section 4.1, we know that if

$$Z_k \;=\; \begin{bmatrix} \cos(\phi^{(k)}) & -\sin(\phi^{(k)}) \\ \sin(\phi^{(k)}) & \cos(\phi^{(k)}) \end{bmatrix} \begin{bmatrix} \sigma_1^{(k)} & 0 \\ 0 & \sigma_2^{(k)} \end{bmatrix} \begin{bmatrix} \cos(\psi^{(k)}) & -\sin(\psi^{(k)}) \\ \sin(\psi^{(k)}) & \cos(\psi^{(k)}) \end{bmatrix}^T$$

is an SVD, then $\tilde{\mathcal{E}}_k$ has tilt $\phi^{(k)}$ and semiaxes $|\sigma_1^{(k)}|$ and $|\sigma_2^{(k)}|$.

We know from (3.11) that

$$\begin{aligned} u^{(2k)} &= S_n^k u^{(0)}, \quad u^{(2k+1)} = S_n^k u^{(1)}, \\ v^{(2k)} &= S_n^k v^{(0)}, \quad v^{(2k+1)} = S_n^k v^{(1)}. \end{aligned} \tag{4.8}$$

Since $S_n^n = I$, we have $u^{(2n)} = u^{(0)}$, $v^{(2n)} = v^{(0)}$, $u^{(2n+1)} = u^{(1)}$, and $v^{(2n+1)} = v^{(1)}$. It follows from (4.6) that $\tilde{\mathcal{E}}_{2n+k} = \tilde{\mathcal{E}}_k$. Thus, the vertices of $\tilde{\mathcal{P}}_k$ in Algorithm 4 are on the ellipse $\tilde{\mathcal{E}}_{k \bmod 2n}$.

The cycling is "more rapid" when the x-norm and y-norms in Algorithm 4 are p-norms:

$$\| w \|_p \;=\; (|w_1|^p + \cdots + |w_p|^p)^{1/p}.$$

This class of norms has the property that it is preserved under permutation, in particular, $\| S_n w \|_p = \| w \|_p$. It follows from (4.5)–(4.6) that if

$$\begin{aligned} \alpha_0 &= 1/\| u^{(0)} \|_x, \quad \beta_0 = 1/\| v^{(0)} \|_y, \\ \alpha_1 &= 1/\| u^{(1)} \|_x, \quad \beta_1 = 1/\| v^{(1)} \|_y, \end{aligned}$$

then

$$\begin{aligned} \mathcal{P}(\tilde{u}^{(2k)}, \tilde{v}^{(2k)}) &= \mathcal{P}(\, S_n^k u^{(0)} / \| S_n^k u^{(0)} \|_x \,,\, S_n^k v^{(0)} / \| S_n^k v^{(0)} \|_y \,) \\ &= \mathcal{P}(\, u^{(2k)} / \| S_n^k u^{(0)} \|_x \,,\, v^{(2k)} / \| S_n^k v^{(0)} \|_y \,) \\ &= \mathcal{P}(\alpha_0\, u^{(2k)}, \beta_0\, v^{(2k)}), \end{aligned}$$

and likewise $\mathcal{P}(\tilde{u}^{(2k+1)}, \tilde{v}^{(2k+1)}) \;=\; \mathcal{P}(\alpha_1\, u^{(2k+1)}, \beta_1\, v^{(2k+1)})$. This shows that if the x- and y-norms in Algorithm 4 are p-norms, then $\tilde{\mathcal{P}}_k$ can be obtained from $\mathcal{P}_k$ simply by scaling $u^{(k)}$ and $v^{(k)}$ by α_0 and β_0 if k is even and by α_1 and β_1 if k is odd. There are thus a pair of ellipses underlying Algorithm 4 in this case.

We mention that if the x- and y-norms in Algorithm 4 are p-norms *and* n is odd, then all the polygons have their vertices on the same ellipse. This is because when n is odd, the vector $\mathbf{c}(\tau + \pi/n)$ is a permutation of $\mathbf{c}(\tau)$ and the vector $\mathbf{s}(\tau + \pi/n)$ is a permutation of $\mathbf{s}(\tau)$. It follows from (3.9)–(3.10) that $u^{(1)}$ and $v^{(1)}$ are permutations of $u^{(0)}$ and $v^{(0)}$, respectively, and therefore have the same p-norm.

In addition to playing with norms in Algorithm 2 we can also play with the parameter α. For example, we observe that if α is randomly selected from the interval $[0, 1]$ each iteration, then the polygons still converge to an ellipse with 45-degree tilt.

5. Summary. The polygon averaging problem and its analysis is a metaphor for matrix-based computational science and engineering. Consider the path that we followed from phenomenon to explanation. We *experimented* with a simple iteration and observed that it transforms something that is chaotic and rough into something

that is organized and smooth. As a step towards explaining the limiting behavior of the polygon sequence, we described the averaging process in *matrix-vector notation*. This led to an *eigenanalysis*, the identification of a crucial *invariant subspace*, and a vertex-vector *convergence analysis*. We then used the *singular value decomposition* to connect our algebraic manipulations to a simple underlying geometry. These are among the most familiar waypoints in computational science and engineering. The next step in this tradition would be to see how our results "scale" to higher dimensions. We challenge the reader to characterize what happens if the repeated averaging process is applied to a closed polygonal line in $\mathbb{R}^k$.

Acknowledgments. The authors would like to thank Professor Bobby Kleinberg for discussions leading to the matrix-vector product formulation of the averaging process and the ensuing analysis. They would also like to thank the referees for their various comments and suggestions for improving the paper.

REFERENCES

[1] A.W. Fitzgibbon, M. Pilu, and R.B. Fisher, *Direct least-squares fitting of ellipses*, IEEE Trans. Pattern Anal. Mach. Intell., 21 (1999), pp. 476–480.

[2] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.