PHYSCON 2011, León, Spain, September, 5–September, 8 2011

MULTIAGENT SYSTEMS: FROM THE VAN LOAN SCHEME TO SIMPLE LINEAR ALGORITHMS

Pavel Shcherbakov
Institute of Control Science
Russia
sherba@ipu.ru

Abstract

We consider an algorithm developed in [Elmachtoub and Van Loan, 2010] which drives a set of points on the plane to a certain regular configuration. Several generalizations of the algorithm are proposed, its new properties are analyzed, its relationship with formation control methods is discussed, and new and simpler linear algorithms of this type are devised.

Key words

Multiagent systems, the Van Loan scheme, Power iterations, Linear algorithms.

1 Introduction

In applications, there exist various problems which can be roughly formalized as follows. Given points (on the line, circle, plane, space), each of which possesses certain information about the others (itself, neighbors, certain checkpoint, etc.). The problem is to design an algorithm which exploits only such a limited information and drives the points (sometimes referred to as agents) to a certain desired configuration.

The popularity of this research area in the last two to three decades is largely explained by the attained high level of evolution of computational hardware and software with the possibility of applying the results in such diverse and intrinsically remoted areas as transportation, military affairs, control of social groups, etc.

Respectively, the associated body of literature is vast and highly nonhomogeneous, and the terminology often varies depending on the practical interpretation of the mathematical statement of the problem. The keywords in this area are decentralized control, group/cooperative control, formation control, self-organized systems, multi-agent control. One of the most recent monographs that provide a substantial introduction to this subject area, discussion of the typical problems, algorithms, and applications, is [Shoham and Leyton-Brown, 2009], which as well contains a large bibliography.

Typical of all the problems considered in this area are (i) the incompleteness of the information about the system available to the agents and (ii) the absence of a central control body.

This note is aimed at attracting attention to one of unexpected and elegant algorithms of this sort, which was developed in [Elmachtoub and Van Loan, 2010]; we also discuss its generalizations and refinements and propose alternative, simple linear algorithms.

2 The Van Loan Scheme

The problem discussed below was the subject of deep analysis performed in [Elmachtoub and Van Loan, 2010]; in the sequel, it will be referred to as the Van Loan scheme.

On the plane, consider n numbered points $p_i, i = 1, \dots n$, such that every ith point possesses information about its own position and that of the $(i+1)$st point; the last, nth point is informed about its own position and that of the point having number 1.

In [Elmachtoub and Van Loan, 2010], the following control algorithm of evolution of the point set $\{p_i\}_1^n$ was devised. Let x_i^k, y_i^k denote the coordinates of the ith point at the kth step of the algorithm and introduce the compound vectors $x^k = (x_1^k, \dots, x_n^k)^\mathsf{T}$ and $y^k = (y_1^k, \dots, y_n^k)^\mathsf{T}$. Every step of the algorithm consists of the following two stages:

$$\mathbf{A}: \begin{cases} \tilde{x}_i^{k+1} = \frac{1}{2}(x_i^k + x_{i+1}^k), \quad i = 1, \dots, n-1; \\ \quad \tilde{x}_n^{k+1} = \frac{1}{2}(x_n^k + x_1^k), \\ \\ \tilde{y}_i^{k+1} = \frac{1}{2}(y_i^k + y_{i+1}^k), \quad i = 1, \dots, n-1; \\ \quad \tilde{y}_n^{k+1} = \frac{1}{2}(y_n^k + y_1^k); \end{cases}$$

$$\mathbf{B}: \ x^{k+1} = \tilde{x}^{k+1}/\|\tilde{x}^{k+1}\|; \ y^{k+1} = \tilde{y}^{k+1}/\|\tilde{y}^{k+1}\|.$$

In other words, at stage **A**, the new position of the ith point is taken as its own position at the previous step averaged with that of the successor point numbered $(i+1)$; the last, nth point is cyclically averaged with the first one. At stage **B**, independent normalization is applied to the compound vectors x and y.

The following unexpectedly surprising result was obtained in [Elmachtoub and Van Loan, 2010].

Theorem 1. *([Elmachtoub and Van Loan, 2010]). Let the initial configuration satisfy the centering condition* $\sum x_i^0 = 0$, $\sum y_i^0 = 0$. *Then, as* $k \to \infty$, *the points tend to locate on the ellipse centered at the origin and rotated by* $\pi/4$.

The matrix S *of the limiting ellipse is calculated as follows. Introduce the vectors*

$$\tau = \frac{2\pi}{n}\begin{pmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{pmatrix}; \quad c = \cos(\tau); \quad s = \sin(\tau),$$

compose the scalars $c_x \doteq c^\mathsf{T} x^0$, $s_x \doteq s^\mathsf{T} x^0$, $c_y \doteq c^\mathsf{T} y^0$, $s_y \doteq s^\mathsf{T} y^0$, *and form the* 2×2 *matrix*

$$A = \sqrt{\frac{2}{n}}\begin{pmatrix} \dfrac{c_x}{\sqrt{c_x^2+s_x^2}} & \dfrac{s_x}{\sqrt{c_x^2+s_x^2}} \\ \dfrac{c_y}{\sqrt{c_y^2+s_y^2}} & \dfrac{s_y}{\sqrt{c_y^2+s_y^2}} \end{pmatrix}. \quad (1)$$

The limiting ellipse has the form

$$\mathcal{E} = \{p \in \mathbb{R}^2 : p^\mathsf{T} S^{-1} p = 1\}, \quad S = (AA^\mathsf{T})^{1/2}. \quad (2)$$

Figure 1 depicts certain initial configuration for $n = 10$ points generated randomly uniformly on the unit square $[-1;\ 1]^2$, and Figure 2 shows their location after 40 steps of the algorithm.

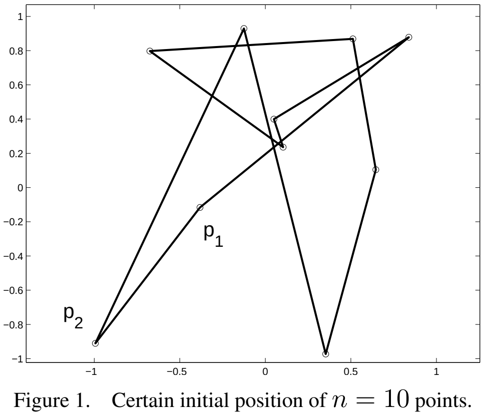

Figure 1. Certain initial position of $n = 10$ points.

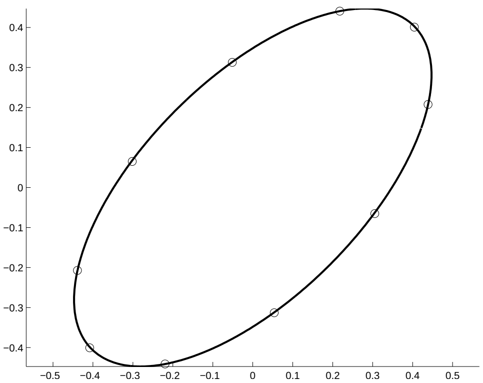

Figure 2. Location of points after $k = 40$ steps.

2.1 Discussion

It is easy to see that stage **A** can be written down in matrix-vector form as soon as the following matrix is introduced:

$$M = \frac{1}{2}\begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

where all unmarked entries are zeros. With this notation, the step of the algorithm writes

$$x^{k+1} = Mx^k/\|Mx^k\|;\ y^{k+1} = My^k/\|My^k\|, \quad (3)$$

which is nothing but the independent application of the power method for the matrix M with initial conditions x^0 and y^0; e.g., see [Amosov, Dubinskii, and Kopchenova, 2008; Tyrtyshnikov, 2007; Golub and Van Loan, 1993]. This consideration is the cornerstone of the detailed analysis of the algorithm performed in [Elmachtoub and Van Loan, 2010]; accordingly, the tools and methods exploited in this analysis can be limited to the basics of linear algebra.

Specifically, it was shown in [Elmachtoub and Van Loan, 2010] that the rate of convergence of the point set to the limiting ellipse depends on the initial location (namely, on the closedness of the n-dimensional vectors x^0, y^0 to a certain 2D plane) and on the number n of points, which defines uniquely the matrix M. More precisely, the rate of convergence depends on the ratio of the 3rd and the 2nd largest eigenvalues of the matrix M (they can be obtained in closed form); the smaller n, the faster the convergence.

By convergence, we mean the following. For every $\varepsilon > 0$ there exists an integer K such that for all $k \geq K$ and $i = \overline{1,n}$, it holds $|(p_i^k)^\mathsf{T} S^{-1} p_i^k - \varepsilon| \leq 1$; i.e., from a certain time instant K onward, every point p_i^k will be located close to the limiting ellipse. However, the whole set of points exhibits a nonstationary behavior; i.e., in the process of iterations, the points “jump over the ellipse" rather than tend to certain fixed positions on it. It is shown in [Elmachtoub and Van Loan, 2010] that at all even (respectively, odd) iterations, the *total location* of the points is the same.

Note that both x– and y–coordinates of the points change according to the same law but independently of each other. moreover, at every step, the coordinates of each 2D point are normalized by the whole compound vector. In the framework of the theory of multiagent systems this normalization is associated with a considerable information available to each agent and therefore does not fit well the ideas of decentralization, distributed information, etc., since for the proper evolution of every point $p_i^k \to p_i^{k+1}$ one needs to have information about the whole population. It would be of great interest to propose any meaningful interpretation of the algorithm in terms of multiagent systems, especially because no hints are given in [Elmachtoub and Van Loan, 2010] regarding this issue, and the conducted analysis is purely theoretical.

In the sections to follow, simple algorithms are proposed, which are ideologically closer to the concepts of multiagent systems. Prior to formulating these algorithms, we now discuss some new properties and generalizations of the Van Loan scheme.

2.2 New properties and generalizations

Uniformity of the limiting location. The first simple property relates to the uniform distribution of points. First, it is shown in [Elmachtoub and Van Loan, 2010] that after several iterations onward (this number can be evaluated) the points are positioned in the order of their labels (numbers). Moreover, a quick routine analysis immediately yields the following regularity of the location of the points on the limiting ellipse. Under a nonsingular linear coordinate transformation with matrix S^{-1}, the ellipse $\mathcal{E}$ transforms into the unit circle, and the points $S^{-1}p_i^k$ will be located equidistantly on this circle.

Round-off errors. This property relates to the sensitivity of the algorithm to the round-off errors in the machine arithmetic.

First of all we note that the centering property $\sum x_i^0 = \sum y_i^0 = 0$ is essential; in what follows, the number $\frac{1}{n}\sum x_i$ will be referred to as the centroid of the vector x. This property guarantees that in the expansion of the initial vectors x^0 and y^0 over the eigenvectors $e_1, \ldots, e_n$ of the matrix M, the coefficient of e_1 (this eigenvector is associated with the largest eigenvalue $\lambda_1 = 1$), is equal to zero. It is known (e.g., see [Amosov, Dubinskii, and Kopchenova, 2008; Tyrtyshnikov, 2007]) that under this condition, the power method *does not* converge which results in the observed behavior of the system of points. Otherwise, the power method converges to the vector e_1, since λ_1 is the simple eigenvalue (having multiplicity 1). It can be immediately shown that e_1 is of the form $e_1 = (a, \ldots, a)^\mathsf{T}$, where a is a number, since eigenvectors are defined to the accuracy of a constant scalar multiplier. Hence, if x^0 and y^0 are not centered then, depending on the norm in (3), the iterations will converge to the limiting vector $(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})^\mathsf{T}$ for the euclidean norm, or to $(\frac{1}{n}, \ldots, \frac{1}{n})^\mathsf{T}$ for the l_1-norm, or to $(1, \ldots, 1)^\mathsf{T}$ for the l_∞-norm, etc. Finally, it is clear to see that the sign of the limiting vector (the sign of a) coincides with that of centroid of the initial vector. In other words, the total set of points p_i converges to one of the points $(\pm\frac{1}{\sqrt{n}}, \pm\frac{1}{\sqrt{n}})^\mathsf{T}$ (for the euclidean norm).

As implemented in a computer, round-off errors of matrix-vector operations lead to the same effect; namely, after a sufficiently large number of iterations, the centroid of initially centered vector is no longer equal to zero, so that the iterations will converge to the eigenvector e_1. Figure 3 depicts the change of the initially zero value of the centroid of x^k for the system of points in the example on Fig. 1 (the centroid of y^k exhibits a similar behavior) and, as a consequence, the convergence of the set $\{p_i\}_1^n$ to a single point. To avoid such an effect and retain the desired ellipsoidal behavior, it is sufficient to perform artificial centering once in a few iterations.

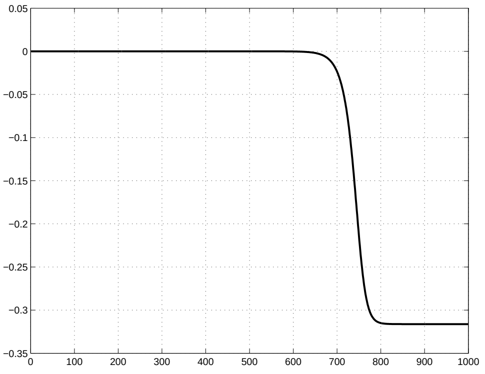

Figure 3. Loss of centering and convergence to e_1.

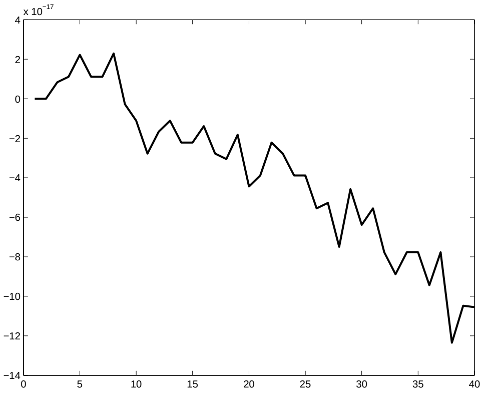

Figure 4. Zoom of the first 40 iterations at the previous figure.

Shift. Let us briefly consider the following modification of the basic scheme:

$$
\begin{aligned}
x^{k+1} &= Mx^k/\|Mx^k\| + \mathbf{1}u, && (4)\\
x^{k+1} &= x^{k+1} - \mathbf{1}\,\frac{1}{n}\sum_{i=1}^{n} x_i^{k+1};\\
y^{k+1} &= My^k/\|My^k\| + \mathbf{1}v, && (5)\\
y^{k+1} &= y^{k+1} - \mathbf{1}\,\frac{1}{n}\sum_{i=1}^{n} y_i^{k+1};
\end{aligned}
$$

where $u, v \in \mathbb{R}$ are fixed and $\mathbf{1} \in \mathbb{R}^n$ is the vector having all components equal to unity. Note that centering is performed at every step. The exogenous inputs $\mathbf{1}u, \mathbf{1}v$ can be thought of as noise, control, reference signal, etc. Obviously, incorporation of the terms $\mathbf{1}u, \mathbf{1}v$ and centering are mutually inverse operations; therefore, the trajectories of the points p_i remain the same. A more meaningful example of this modified algorithm will be considered in Section 3 below.

Generalization to many dimensions. A generalization of the Van loan scheme to the case where n points $p_i^0 = (\underbrace{x_i^0,\, y_i^0,\, \ldots,\, z_i^0}_{d\ \text{components}})^\mathsf{T}$, $i = 1, \ldots, n$, are located in the d-dimensional space rather than on the plane is thought to be most interesting.

In that case, the evolution is represented by the d-fold application of the power method:

$$
\begin{aligned}
x^{k+1} &= Mx^k/\|Mx^k\|;\\
y^{k+1} &= My^k/\|My^k\|; && (6)\\
&\ \vdots\\
z^{k+1} &= Mz^k/\|Mz^k\|,
\end{aligned}
$$

and at the first glance, the limiting position (if any) is far from being obvious. However it is clear that, similarly to the 2D case, each of the d compound vectors $x, y, \ldots, z$ is subject to change according to the same algorithm as above, independently of each other. Hence, the projection of the limiting position on any of the coordinate planes is nothing but the ellipse specified by a matrix of the form (1). Equivalent to the canonical description of ellipse (2) is the description via the A matrix (1), as a linear transformation of the circle:

$$
\mathcal{E} = \left\{ p \in \mathbb{R}^2 : p = A\begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix} \right\},
$$

as φ sweeps $[0,\ 2\pi]$. Hence, using the same notation $c_x \doteq c^\mathsf{T}x^0$, $s_x \doteq s^\mathsf{T}x^0$, $c_y \doteq c^\mathsf{T}y^0$, $s_y \doteq s^\mathsf{T}y^0$, $\ldots$, $c_z \doteq c^\mathsf{T}z^0$, $s_z \doteq s^\mathsf{T}z^0$, as in Theorem 1, in the d-dimensional case the transformation matrix has the form

$$
A_d = \sqrt{\frac{2}{n}}\begin{pmatrix}
\dfrac{c_x}{\sqrt{c_x^2 + s_x^2}} & \dfrac{s_x}{\sqrt{c_x^2 + s_x^2}}\\
\dfrac{c_y}{\sqrt{c_y^2 + s_y^2}} & \dfrac{s_y}{\sqrt{(c_y^2 + s_y^2}}\\
\vdots & \vdots\\
\dfrac{c_z}{\sqrt{c_z^2 + s_z^2}} & \dfrac{s_z}{\sqrt{c_z^2 + s_z^2}}
\end{pmatrix} \in \mathbb{R}^{d\times 2}. \quad (7)
$$

Accordingly, $S = (A_d A_d^\mathsf{T})^{1/2}$ is a $d \times d$–matrix of rank 2 having $d-2$ zero eigenvalues; hence, this matrix defines a 2-dimensional (“flat”) limiting ellipsoid in d-dimensional space of points!

3 Simple linear algorithms

As noted above, the Van Loan scheme assumes excessive information available to the agents; moreover, algorithm (3) is nonlinear and hence, not immediate to analyze.

There is a vast literature directed toward new schemes and analysis of evolution of multi-agent systems in continuous-time, e.g., see [Hara, Kanno, and Tanaka, 2009; Olfati-Saber, Fax, and Murray, 2007]. However, to the best of our knowledge, the discrete-time case is not well elaborated. Following the iterative ideology of the Van Loan scheme, in the rest of the paper we consider several simple linear algorithms in discrete time that require much less information.

Segment. Consider the simplest one-dimensional scheme. Given n points on a line, each of them, x_i, $i = 2, \ldots, n-1$, possesses information (knowledge of the coordinate) about x_{i-1} and x_{i+1}; the point x_1 is informed about certain fixed point a and x_2; finally, x_n possesses information about x_{n-1} and another fixed point $b > a$. The problem is to locate the points equidistantly on the segment $[a,\ b]$ in the order of their indices.

The suggested algorithm is as follows:

$$
\begin{aligned}
&x_i^{k+1} = (x_{i-1}^k + x_{i+1}^k)/2,\ i = 2, \ldots, n-1; \quad (8)\\
&x_1^{k+1} = (a + x_2^k)/2; \quad x_n^{k+1} = (x_{n-1}^k + b)/2,
\end{aligned}
$$

i.e., the new position of a point is equal to the arithmetic mean of its two neighbors (the point a is considered to be a neighbor of x_1, and b is a neighbor for x_n). As above, by neighbors we mean the points whose indices differ by unity. Note that the algorithm does not even require the knowledge of the total amount n of points but is rather based on the information of endpoints and information about only two neighbors.

Let us augment the state vector $x = (x_1, \ldots, x_n)$ by the two fixed endpoints a and b and denote by $\tilde{x}^0 = (a,\ x_1^0, \ldots, x_n^0,\ b)^\mathsf{T} \in \mathbb{R}^{n+2}$ the extended vector of initial position. Introducing the matrix

$$M = \begin{pmatrix} 1 & 0 & & & \\ 0.5 & 0 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.5 & 0 & 0.5 \\ & & & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+2)\times(n+2)} \qquad (9)$$

for the extended vector, the step of algorithm (8) writes

$$\tilde{x}^{k+1} = M\tilde{x}^k,$$

so that the two artificially added points $\tilde{x}_1 = a$ and $\tilde{x}_{n+2} = b$ remain unchanged.

The algorithm is linear in $\tilde{x}$, and its analysis is transparent. Indeed, since $\tilde{x}^k = M^k\tilde{x}^0$, it remains to understand the behavior of the powers of the matrix M. Lengthy but straightforward manipulations lead to

$$M^k \;\rightarrow\; \begin{pmatrix} 1 & 0 \;\ldots\; 0 & 0 \\ \frac{n}{n+1} & \vdots \qquad \vdots & \frac{1}{n+1} \\ \frac{n-1}{n+1} & & \frac{2}{n+1} \\ \vdots & & \vdots \\ \frac{1}{n+1} & & \frac{n}{n+1} \\ 0 & \ldots \qquad 0 & 1 \end{pmatrix} \doteq M_{lim}, \qquad (10)$$

as $k \rightarrow \infty$, so that

$$M_{lim}\begin{pmatrix} a \\ x_1^0 \\ \vdots \\ \vdots \\ x_n^0 \\ b \end{pmatrix} = \begin{pmatrix} a \\ \frac{n}{n+1}a + \frac{1}{n+1}b \\ \frac{n-1}{n+1}a + \frac{2}{n+1}b \\ \vdots \\ \frac{1}{n+1}a + \frac{n}{n+1}b \\ b \end{pmatrix} = \begin{pmatrix} a \\ a + \frac{1}{n+1}(b-a) \\ a + \frac{2}{n+1}(b-a) \\ \vdots \\ a + \frac{n}{n+1}(b-a) \\ b \end{pmatrix}$$

for any initial vector x^0. In other words, the algorithm is globally convergent to a unique limiting position on the segment $[a,\, b]$ such that the points are juxtaposed in the order of increase of their indices, and all the distances $x_1 - a,\ x_2 - x_1, \ldots, x_n - x_{n-1},\ b - x_n$ are equal to $(b-a)/(n+1)$.

The analysis above can as well be performed using the eigenstructure of the matrix M. Let us demonstrate this via the evaluation of the rate of convergence of this algorithm. The matrix M has a unit eigenvalue of multiplicity two that correspond to the two fixed endpoints of the segment; hence, the rate of convergence depends on the third largest eigenvalue. Let us remove the first and the last rows and columns of the matrix M and denote the resulting $n \times n$ matrix by $\tilde{M}$. Next, consider the matrix $A = 2(I - \tilde{M})$ having the form

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

This matrix appears in various applications such as fast Fourier transform, solution of the Sturm–Liouville problem by difference methods, etc. Its eigenvalues are known to be equal to $\lambda_k(A) = 4\sin^2(\pi k/2(n+1))$, $k = 1, \ldots, n$; e.g. see [Voevodin and Kuznetsov, 1984], Section 19. Therefore, the eigenvalues of $\tilde{M}$ (and hence, the rest n of the eigenvalues of M) are equal to $\lambda_k(M) = 1 - 2\sin^2(\pi k/2(n+1))$, $k = 1, \ldots, n$. The largest among them, $\lambda = 1 - 2\sin^2(\pi/2(n+1))$ defines the rate of convergence of the method.

Instead of the two extra points a and b one might consider the initial coordinates of the first and the last points x_1^0 and x_n^0 as fixed endpoints of the target segment. In that case, the algorithm remains the same with the only difference that the matrix M is of dimension $n \times n$, and the points will be located equidistantly on the segment $[x_1^0,\, x_n^0]$.

Segment: Given ratio of distances. Assume that in the scheme above (with the fixed endpoints a and b) the agents are to be positioned in such a way as to attain the given ratio of distances between them: $\lambda_1 : \lambda_2 : \cdots : \lambda_n : \lambda_{n+1}$. Then, instead of the arithmetic mean as in algorithm (8), the next-step position of the point is to be taken as the weighted sum of the coordinates of its neighbors, namely

$$x_i^{k+1} = \frac{\lambda_{i+1}}{\lambda_i + \lambda_{i+1}} x_{i-1}^k + \frac{\lambda_i}{\lambda_i + \lambda_{i+1}} x_{i+1}^k,$$

for $i = 2, \ldots, n-1$ and

$$\begin{aligned} x_1^{k+1} &= \frac{\lambda_2}{\lambda_1 + \lambda_2} a + \frac{\lambda_1}{\lambda_1 + \lambda_2} x_2^k; \\ x_n^{k+1} &= \frac{\lambda_{n+1}}{\lambda_n + \lambda_{n+1}} x_{n-1}^k + \frac{\lambda_n}{\lambda_n + \lambda_{n+1}} b. \end{aligned}$$

As above, introduce the extended vector $\tilde{x} = (a,\, x_1, \ldots, x_n,\, b)^\mathsf{T} \in \mathbb{R}^{n+2}$ and the matrix

$$M = \begin{pmatrix} 1 & 0 & & & \\ \frac{\lambda_2}{\lambda_1+\lambda_2} & 0 & \frac{\lambda_1}{\lambda_1+\lambda_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\lambda_{n+1}}{\lambda_n+\lambda_{n+1}} & 0 & \frac{\lambda_n}{\lambda_n+\lambda_{n+1}} \\ & & & 0 & 1 \end{pmatrix}.$$

Then, the algorithm writes in the same form as $\tilde{x}^{k+1} = M\tilde{x}^k$, so that $\tilde{x}^k = M^k\tilde{x}^0$, and the convergence of iterations to the desired limiting position is guaranteed by the convergence of the powers of the matrix M:

$$M^k \to \begin{pmatrix} 1 & 0 \dots 0 & 0 \\ \frac{\lambda_1+\dots+\lambda_n}{\lambda_1+\dots+\lambda_{n+1}} & \vdots \quad\quad \vdots & \frac{\lambda_1}{\lambda_1+\dots+\lambda_{n+1}} \\ \frac{\lambda_1+\dots+\lambda_{n-1}}{\lambda_1+\dots+\lambda_{n+1}} & & \frac{\lambda_1+\lambda_2}{\lambda_1+\dots+\lambda_{n+1}} \\ \vdots & & \vdots \\ \frac{\lambda_1}{\lambda_1+\dots+\lambda_{n+1}} & & \frac{\lambda_1+\dots+\lambda_n}{\lambda_1+\dots+\lambda_{n+1}} \\ 0 & \dots \quad 0 & 1 \end{pmatrix} \doteq M_{lim},$$

which can be shown similarly to the case of equal distances.

Circle. Algorithm (8) can be easily modified to the following setup. Assume that n numbered points p_i are located on the circle centered at zero so that their positions are defined by the angles $\theta_i \in (0,\, 2\pi)$. Each point is aware of the angles of its (index-related) neighbors and the position of the center. The problem is the same, that is, to drive the points to the equidistant configuration on the circle.

One of the possibilities is to "cut" the circle at an arbitrary point $\theta_c \in [0,\, 2\pi]$ and apply the algorithm for the segment. Indeed, denote $\tilde{\theta} = (\theta_c,\, \theta_1, \dots, \theta_n,\, \theta_c + 2\pi)^{\mathsf{T}} \in \mathbb{R}^{n+2}$ and introduce the $(n+2) \times (n+2)$ matrix

$$M = \begin{pmatrix} 1 & 0 & & & & & \\ \mathbf{2/3} & 0 & \mathbf{1/3} & & & & \\ & 0.5 & 0 & 0.5 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0.5 & 0 & 0.5 & \\ & & & & \mathbf{1/3} & 0 & \mathbf{2/3} \\ & & & & & & 1 \end{pmatrix}.$$

Then the step of the algorithm $\tilde{\theta}^{k+1} = M\tilde{\theta}^k$ corresponds to the averaging of the *polar coordinates* of the neighboring points, i.e., their angles. Here, the two "copies" of the artificially incorporated cutting point, θ_c and $\theta_c + 2\pi$ are analogous to the fixed endpoints in the scheme for the segment, and the difference with this latter scheme with matrix (9) is the appearance of the weights $1/3$ and $2/3$. These weights guarantee that after "binding" the segment back to the circle, the angle between the first and the last points, θ_1 and θ_n (i.e., those around the "invisible sewing point") be equal to the other angles; otherwise, for equal weights 0.5 this angle will be twice as big as the others. The convergence of the algorithm to the equidistant position on the circle is analogous to the above.

The requirements to the initial location can be relaxed by assuming it to be arbitrary and performing the normalization $p_i \to p_i/\|p_i\|$ (letting each point know its own coordinates).

In the circular scheme, the cut can be performed via one of the actual points from the set $\theta_1, \dots, \theta_n$, hence, distinguishing it as a "leading" one. It is natural to choose the first point as a leader.

In that case, denote $\tilde{\theta} = (\theta_1, \dots, \theta_n,\, \theta_1 + 2\pi)^{\mathsf{T}} \in \mathbb{R}^{n+1}$, introduce the matrix

$$M = \begin{pmatrix} 1 & 0 & & & \\ 0.5 & 0 & 0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & 0.5 & 0 & 0.5 \\ & & & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad (11)$$

and write down the step of the algorithm:

$$\tilde{\theta}^{k+1} = M\tilde{\theta}^k. \quad (12)$$

Here, the point θ_1 is fixed, and the analysis of convergence is completely analogous to the previous ones.

Circle: Motion. Similarly to (4)–(5), let us introduce the exogenous noise in the dynamics of system (12)–(11):

$$\tilde{\theta}^{k+1} = M\tilde{\theta}^k + \mathbf{1}\omega, \quad (13)$$

where $\omega \in \mathbb{R}$ is a constant, and $\mathbf{1}$ is the vector with all components being unities. It is seen that as the result of this algorithm, the points tend to juxtapose at equiangular positions on the circle and rotate with the constant velocity ω. Indeed, since $M\mathbf{1}\omega = \mathbf{1}\omega$, we have

$$\tilde{\theta}^k = M^k\tilde{\theta}^0 + \sum_{i=0}^{k-1} M^i\mathbf{1}\omega = M^k\tilde{\theta}^0 + k\,\mathbf{1}\omega,$$

hence,

$$\tilde{\theta}^{k+1} - \tilde{\theta}^k = (M^{k+1} - M^k)\tilde{\theta}^0 + \mathbf{1}\omega$$

and for sufficiently large k we have $\tilde{\theta}^{k+1} - \tilde{\theta}^k \approx \mathbf{1}\omega$, and the vectors $\tilde{\theta}^k$ and $\tilde{\theta}^{k+1}$ define the uniform location.

For a specific initial position $\theta_1^0 < \theta_2^0 < \dots < \theta_n^0$, which prevents the points of mutual "leap-frogging", algorithm (13) can be interpreted from the multi-agent point of view: The points can be thought of as automobiles located in the circular tunnel in the order of increase of their numbers, and the goal is to attain the equidistant position while moving at the velocity ω. Each agent in this scheme is informed of the desired velocity and the coordinates of the two neighbors, but not the overall amount of agents. A modification of the algorithm for the case where only the leader is informed of the desired velocity can also be devised.

4 Conclusion and further research

In this note, we considered a simple and elegant algorithm that governs the location of points on the plane and provided several modifications of this scheme. Clearly, the abilities of the Van Loan scheme are not limited to those discussed above; for instance, change of sign of the south-east entry of the matrix M leads to interesting animated behavior.

We also note that the Van Loan scheme can be formulated in continuous time; in that case, difference equations (**A**) take the form of the corresponding differential equations.

There are several directions for future research. First of all, of most interest is the design and analysis of linear algorithms of the form (13), which use a limited amount of information yet having transparent practical interpretation. Next, the algorithms in Section 3 are one-dimensional, including those devised for the circle. It is of apparent interest to develop multidimensional analogs of these algorithms such as those driving three-dimensional points to a desired spatial configuration, e.g., a sphere.

The author is indebted to B.Polyak for attracting his attention to formation control issues, specifically, to paper [Elmachtoub and Van Loan, 2010], and for fruitful discussions.

References

A.N. Elmachtoub and C.F. Van Loan, From Random Polygon to Ellipse: An Eigenanalysis, *SIAM Review*, 2010, **52**, 1, pp. 151–170.

Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2009.

A.A. Amosov, Yu.A, Dubinskii, and N.V. Kopchenova, *Computational Methods for Engineers* (in Russian), Moscow: Vysshaya Shkola, 2008.

E.E. Tyrtyshnikov, *Methods of Numerical Analysis* (in Russian), Moscow: Publishing Center “Akademiya”, 2007.

G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

Ya.I. Petrikevich, Linear Algorithms for Control of Geometric Location of the Plants in Mult-Agent Systems, *Control of Large-Scale Systems, “Networked Models in Control”* (in Russian), Moscow: ICS, 2010, pp. 665-680.

R. Olfati-Saber, J.A. Fax, and R.M. Murray, Consensus and Cooperation in Networked Multi-Agent Systems, *Proc. IEEE*, 2007, **95**, 1, pp. 215–233.

S. Hara, M. Kanno, and H. Tanaka, Cooperative Gain Output Feedback Stabilization for Multi-Agent Dynamic Systems, *Proc. IEEE CDC*, Shanghai, December 2009, pp. 877–882.

V.V. Voevodin and yu.A. Kuznetsov, *Matrix Computations*, (in Russian), Moscow: Nauka, 1984.