Front. Math. China 2016, 11(6): 1379–1418
DOI 10.1007/s11464-016-0573-4

RESEARCH ARTICLE

Efficient initials for computing maximal eigenpair

Mu-Fa CHEN

School of Mathematical Sciences, Beijing Normal University, Laboratory of Mathematics and Complex Systems (Beijing Normal University), Ministry of Education, Beijing 100875, China

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2016

Abstract This paper introduces some efficient initials for a well-known algorithm (an inverse iteration) for computing the maximal eigenpair of a class of real matrices. The initials not only avoid the collapse of the algorithm but are also unexpectedly efficient. The initials presented here are based on our analytic estimates of the maximal eigenvalue and a mimic of its eigenvector for many years of accumulation in the study of stochastic stability speed. In parallel, the same problem for computing the next to the maximal eigenpair is also studied.

Keywords Perron-Frobenius theorem, power iteration, Rayleigh quotient iteration, efficient initial, tridiagonal matrix, Q-matrix
MSC 15A18, 65F15, 93E15, 60J27

1 Introduction. Two algorithms and a typical example

Consider a nonnegative irreducible matrix $A = (a_{ij})$ on $E := \{0, 1, \ldots, N\}$, $N < \infty$. By the well-known Perron-Frobenius theorem, the matrix has uniquely a positive eigenvalue $\rho(A)$ having positive left-eigenvector and positive right-eigenvector. Moreover, both the left-eigenspace and the right-eigenspace of $\rho(A)$ have dimension one. This eigenvalue is maximal in the sense that for every other eigenvalue λ_k, we have $\rho(A) \geqslant |\lambda_k|$. The last equality sign appears only if A has a period $p > 1$. For instance, for

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix},$$

we have $p = 2$ and the eigenvalues of A are $\pm\sqrt{2}$ and 0. However, we may

Received October 22, 2015; accepted June 23, 2016
E-mail: mfchen@bnu.edu.cn

assume that $\rho(A) > |\lambda_k|$ for every other eigenvalue λ_k. Actually, if $\lambda = \rho e^{i\theta}$ with $\theta \neq k\pi/2$ for every odd $k \in \mathbb{Z}$, then for every $\varepsilon > 0$, we have $\rho + \varepsilon > |\rho e^{i\theta} + \varepsilon|$. This means that the required assertion holds for the shifted pair $\rho + \varepsilon$ and $\lambda + \varepsilon$. In other words, an analog of the Perron-Frobenius theorem is meaningful for the matrices having nonnegative off-diagonal elements only, their diagonal elements can be arbitrary but real. By a shift if necessary, such a matrix can be transformed into a nonnegative one: the maximal eigenvector is kept but their maximal eigenvalues are shifted from one to the other. In this paper, we are interested in computing $\rho(A)$ and its corresponding eigenvector. This is a very important problem, we will come back to its motivation in the next section.

There are mainly two popular algorithms for this problem. Unless otherwise stated, the eigenvector below means the right-eigenvector. Then, the maximal eigenpair (the maximal eigenvalue and its eigenvector) is denoted by $(\rho(A), g)$.

Power iteration *Given an initial vector $v_0 \in \mathbb{R}^{N+1}$ having a nonzero component in the direction of g with $\|v_0\| = 1$, define*

$$v_k = \frac{Av_{k-1}}{\|Av_{k-1}\|}, \quad z_k = \|Av_k\|, \quad k \geqslant 1, \tag{1}$$

where $\|\cdot\|$ is an arbitrary but fixed vector norm. Then v_k converges to the eigenvector g of $\rho(A)$ and z_k converges to $\rho(A)$ as $k \to \infty$.

Even it is not necessary, in the next algorithm, we fix the vector norm to be the Euclidean one (or equivalently, the ℓ^2-norm). Actually, a refined choice is using the inner product and the norm in the space $L^2(\mu)$ for a suitable measure μ to be specified case by case, as illustrated by the improved algorithm given at the end of Sections 3 and 4. See also Section 6.

Rayleigh quotient iteration (a variation of inverse iteration) *Choose a pair (z_0, v_0) as an approximation of $(\rho(A), g)$ with $v_0^* v_0 = 1$, where v^* is the transpose of v. In particular, one may set $z_0 = v_0^* A v_0$ for a given v_0. At the kth $(k \geqslant 1)$ step, solve the linear equation in w_k:*

$$(A - z_{k-1} I) w_k = v_{k-1}, \tag{2}$$

where I is the identity matrix on E, and define

$$v_k = \frac{w_k}{\sqrt{w_k^* w_k}}, \quad z_k = v_k^* A v_k.$$

If the pair (z_0, v_0) is close enough to $(\rho(A), g)$, then (z_k, v_k) converges to $(\rho(A), g)$ as $k \to \infty$.

In what follows, unless otherwise stated, we fix z_0 to be the particular choice just defined. We now use a typical example (which will be studied time by time in the paper) to illustrate the effectiveness and their difference of the above two algorithms.

Example 1 Let $E = \{0, 1, \ldots, 7\}$ and

$$Q = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -5 & 2^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^2 & -13 & 3^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3^2 & -25 & 4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4^2 & -41 & 5^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5^2 & -61 & 6^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6^2 & -85 & 7^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7^2 & -113 \end{pmatrix}.$$

Then we have $\rho(Q) \approx -0.525268$ with eigenvector

$$\approx (55.878,\ 26.5271,\ 15.7059,\ 9.97983,\ 6.43129,\ 4.0251,\ 2.2954,\ 1.0)^*.$$

Starting from v_0 which is the normalized vector of

$$(1,\ 0.587624,\ 0.426178,\ 0.329975,\ 0.260701,\ 0.204394,\ 0.153593,\ 0.101142)^*,$$

the power iteration (applied to the nonnegative $A := 113\,I + Q$) arrives at $-0.525268 \approx \rho(Q)$ after 990 iterations. Here, we adopt the ℓ^1-norm:

$$\|v\| = \sum_{k \in E} |v_k|.$$

We now give a little more details about the computations for this example.

Table 1 gives us partial outputs of $(k, -z_k)$. The corresponding figure below shows that $-z_k$ decreases quickly for small k, but the convergence goes very slow for large k.

Table 1 Partial outputs of $(k, -z_k)$

k	$-z_k$	k	$-z_k$	k	$-z_k$
0	2.11289	14	0.877012	100	0.589332
1	1.42407	15	0.86311	120	0.574136
2	1.37537	16	0.850338	140	0.56279
3	1.22712	17	0.838548	160	0.554157
4	1.1711	18	0.827619	180	0.547529
5	1.10933	19	0.817449	200	0.542423
6	1.06711	20	0.807953	300	0.529909
7	1.02949	30	0.738257	400	0.526517
8	0.998685	40	0.694746	500	0.525603
9	0.971749	50	0.664453	600	0.525358
10	0.948331	60	0.641946	700	0.525292
11	0.927544	70	0.624473	800	0.525274
12	0.908975	80	0.610468	900	0.52527
13	0.892223	90	0.598963	$\geqslant$ 990	0.525268

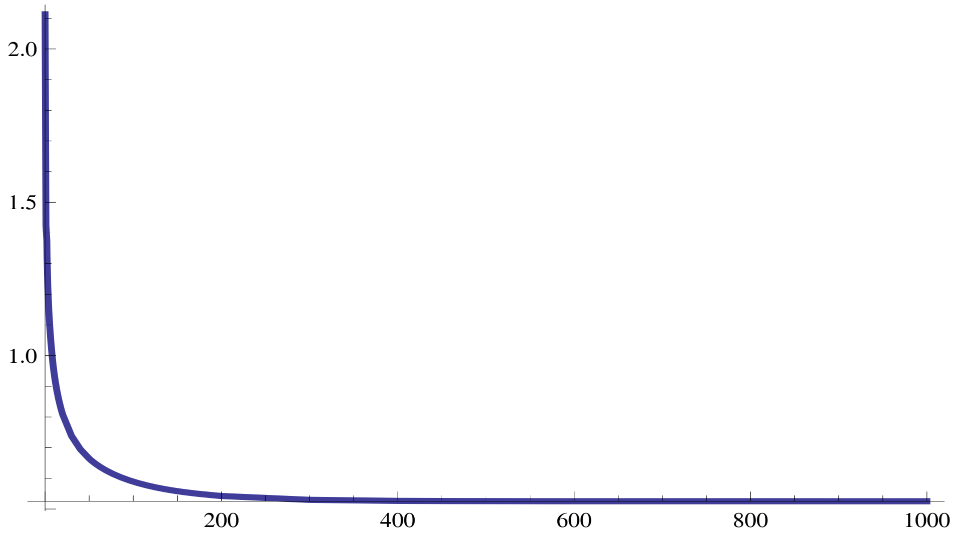

Fig. 1 Figure of $-z_k$ for $k = 0, 1, \ldots, 1000$

The advantage of the first algorithm is that it allows us to use a quite arbitrary positive initial vector v_0. The reason why the convergence of the example at the beginning steps goes quite fast is because we have used a very good initial v_0, as will be studied in Section 3. However, for larger k, the convergence becomes very slow, that is the limitation of this algorithm. From Fig. 1, it is clear that one may stop the computation at 300 iterations since then the results are almost the same. However, we keep going on until the six precisely significant digits as limited by a computer using Mathematica 9. The reason for doing so is for the comparison with other algorithms to be studied later.

Certainly, we expect the second algorithm to be more efficient. Now, what can we expect? Since this problem is often used in practice, we would be very happy if a new algorithm can reduce the number of iterations seriously, say, 500 for instance. Certainly, we would be very surprising if it can be reduced to 250. Let us think this question more carefully. Suppose that we are now interested in the maximal eigenvalue only, and suppose that we know it is located on $(0, 1)$ (actually, as we will see by Proposition 11 below, the maximal eigenvalue of $A := 113\,I + Q$ is located in $(0, 113)$). We may use the Golden Section Search (a famous method in optimization theory), its speed is about 0.618^{-1}. Then, to obtain the six precisely significant digits as in the last example, one needs at least 24 iterations since $10^{-6} \approx 0.618^{24}$. Suppose that we can adopt a faster algorithm, the Bisection Method. Then it requires about 20 iterations since $10^{-6} \approx 2^{-20}$. Hence, it is reasonable if an algorithm uses more than 20 iterations to arrive at the same precise level. Having this analysis in mind, we were shocked when the next result came to us.

Example 2 The matrix Q and the initial vector v_0 are the same as in the last example but we now adopt the ℓ^2-norm. The Rayleigh quotient iteration (applied to Q) starts at

$$z_0 = v_0^* Q v_0 \approx -0.78458$$

and then arrives at the same result as in the last example at the second step:

$$z_1 \approx -0.528215, \quad z_2 \approx -0.525268.$$

Example 2 is the main illustrating example (which will be further improved by Example 7 below) of this paper. It shows that the second algorithm can be extremely powerful. The key to this result is that we have chosen an efficient initial vector v_0 and then the resulting z_0 is also close to $\rho(Q)$. It may be the position to compare the use of ℓ^1-norm and ℓ^2-one. Let everything be the same as in the last example but replacing the ℓ^2-norm by the ℓ^1-one. Then the iteration starts at $z_0 \approx -0.367937$ and arrives at the same result at the third step:

$$z_1 \approx -0.509272, \quad z_2 \approx -0.52509, \quad z_3 \approx -0.525268.$$

The result comes with no surprising: it is easier to use the ℓ^1-norm in the computation but it is a little less efficient than using the ℓ^2-norm.

We have seen the power of the second algorithm. However, "too good" is dangerous. Each eigenvalue $\lambda_k \neq \rho(A)$ can be a pitfall of the algorithm provided either z_0 is close enough to λ_k or v_0 is close enough to the eigenvector g_k of λ_k. The next example illustrates the latter situation. For which a simpler v_0 deduces its corresponding z_0 to be more close to λ_2 rather than λ_3.

Here and in what follows, we often use the so-called Q-matrix

$$Q = (q_{ij} : i, j \in E),$$

which means that $q_{ij} \geqslant 0$ for every pair $i \neq j$ and $\sum_{j\in E} q_{ij} \leqslant 0$ for every $i \in E$. This implies the intrinsic use of probabilistic idea. For convenience, we often write by

$$0 < \lambda_0 < |\lambda_1| \leqslant |\lambda_2| \leqslant \cdots,$$

where $\{\lambda_j\}$ is the sequence of the eigenvalues of $-Q$. Then $\lambda_0 = -\rho(Q)$.

Example 3 The matrix Q is the same as the last example and we use again the ℓ^2-norm. Replace Q by $-Q$ (then the corresponding $z_k > 0$). Choose the initial vector v_0 to be the normalized uniform vector:

$$v_0 = \frac{1}{\sqrt{8}}\{1, 1, 1, 1, 1, 1, 1, 1\}.$$

Then with the particular choice given in the algorithm

$$z_0 = v_0^*(-Q)v_0 = 8,$$

we obtain the following output at the first 4 steps of the iterations:

$$z_1 \approx 4.78557, \quad z_2 \approx 5.67061, \quad z_3 \approx 5.91766, \quad z_4 \approx 5.91867 \approx \lambda_2.$$

The first two eigenvalues of $-Q$ are

$$\lambda_0 \approx 0.525268, \quad \lambda_1 \approx 2.00758, \quad \lambda_3 \approx 13.709,$$

respectively. Hence, the limit λ_2 is quite away from what we are interested in.

By the way, let us mention that in practice, we can stop our computation once the components of the first output v_1 have different signs, and try to choose a new initial pair (v_0, z_0). This is due to the fact that the maximal vector should be positive/negative up to a constant. Here, in the last example,

$$\begin{aligned} v_1 = (&-0.26762,\ 0.242432,\ -0.522646,\ -0.579319,\\ &-0.423469,\ -0.253452, -0.124365,\ -0.0425044)^*. \end{aligned}$$

Each of the components is negative except the second one.

The next example shows that we can still arrive at the expected result for a good initial z_0 even if v_0 is quite rough.

Example 4 Everything is the same as in the last example except

$$z_0 = 2.05768^{-1} \approx 0.485985.$$

Then $\{z_k\}$ approaches to λ_0 at the second step:

$$z_1 \approx 0.525998, \quad z_2 \approx 0.525268.$$

This paper is organized as follows. In the next section, we first review the five sources of the motivation for our problem. Then we recall the known convergence of these algorithms. From the above examples, we have seen that the second algorithm is much more attractive. To which, we need a careful design in choosing the initial pair (v_0, z_0). Clearly, an efficient initial pair is just a good estimate of the pair in advance. This itself is a hard topic in the study of eigenvalue problem and so it is understandable that the initial problem is still largely open in the eigenvalue computation theory. A complete, analytic (explicit) solution to this problem is presented in Section 3 first for tridiagonal matrices (after a suitable relabeling if necessary), and then for a class of more general matrices in terms of the so-called Lanczos tridiagonalization procedure. The main extension to the general situation is presented in Section 4 which consists of two subsections. In the first one, we concentrated on the construction of z_0 for a fixed simplest v_0. The second one in even more technical, in which we are mainly working on the construction of v_0. A number of examples are illustrated, case by case, for the results in the paper. It is remarkable that only the one-step iteration scheme, as illustrated by the two algorithms used above, is used in the paper. In Section 5, we make either additional proofs of some results in the main context, or additional remarks on related problems. In particular, we prove a convergence result of our approximating procedure for the principal eigenvalue of birth–death processes which have been studied for a long period up to now. A summary for the use of the algorithms up to Section 4 is given at the end of Section 5. The study on the next eigenpair is delayed to Section 6.

2 Motivation of problem and convergence of algorithms

In this section, for the reader's convenience, we recall briefly the motivation of our problem and the well-known convergence of the two algorithms introduced in the first section.

2.1 Motivation

It seems not necessary to mention the value of the study on the problem since the matrix eigenvalue computation is used almost everywhere. The next five sources reflect more or less the road where we started and finally arrive here.

Google's PageRank

When we search an expression from the network, a large number of webpages are collected. The question is how to output them on the screen of our computer. For this, we need to rank the pages. The procedure goes as follows. According to the connections of the websites, we get a nonnegative matrix A. To which we have the largest eigenvalue $\rho(A)$ and its corresponding positive left-eigenvalue. The normalized left-eigenvector gives us an order of the webpages, that is the PageRank as we required. Nowadays, there are a large number of publications on Google's PageRank, see for instance [12], in which the power iteration is studied but not the inverse iteration.

Global optimization of planned economy

Regarding the matrix A as a structure matrix in economy, Hua [11] proved that the optimal input of the planned economy is the left-eigenvector u of $\rho(A)$. Surprisingly, Hua [11] also proved that if one uses a different input rather than u, then the economy will go to collapse (i.e., some components of the product in the economic system will become less or equal to zero). Mathematically, this situation is very much the same as the last one, but in a completely different context. As far as I know, the practical algorithms for $(u, \rho(A))$ were not studied carefully during that period, except a formula was mentioned in [11]:

$$\rho(A) = \lim_{\ell\to\infty} \bigg(\frac{\text{Trace }(A^\ell)}{N+1}\bigg)^{1/\ell}.$$

Stationary distribution of time-discrete Markov chain

If A itself is a transition probability matrix, then the left-eigenvector corresponding to the largest eigenvalue one is nothing but the stationary distribution of the corresponding Markov chain. This explains the stability meaning in the two situations just discussed above. Based on this idea, we obtained a probabilistic proof of Hua's collapse theorem. Refer also to [4, Chapter 10] for additional story and related references.

Computing the stationary distribution of a given Markov chain is very important in practice and so has been studied quite a lot in the past years, including the so-called Markov Chain Monte Carlo (MCMC), perfect/backward coupling, and so on.

Exponential decay of time-continuous Markov chain

The maximal eigenvalue $\rho(Q)$ in Example 1 describes the exponential decay rate of the Markov chain with semigroup $(P_t = \mathrm{e}^{tQ} : t \geqslant 0)$. The present paper is based on our study on this topic, as will be seen from the subsequent sections.

Phase transitions

The last topic and the investigation on related stability speed are actually motivated from the study on phase transitions in statistical mechanics (cf. [3,4] for more references within). This is a challenge topic in mathematics since it is mainly in infinite-dimensional setting. To which, the mathematical tools are rather limited. Therefore, we have to look for new tools or develop some known traditional tools. To this end, we have already visited several branches of mathematics, including the computation theory. We are now glad to be able to say something on the last field after a long trip of the study.

In the second part of this section, we review some well-known facts on the convergence of the algorithms.

2.2 Convergence of algorithms

Here is the convergence of the power iteration. In this subsection, we suppose that the given matrix A (not necessarily nonnegative) has the dominant eigenvalue λ_0 (i.e., $|\lambda_0| > |\lambda_j|$ for all other eigenvalues λ_j) which is simple. The extension to the periodic situation is also possible, but is omitted here, one simply replaces the convergence of the original sequence by a subsequence.

Lemma 5 *Suppose that the initial vector v_0 has a nonzero component in the direction of the dominant eigenvector g. Then*

$$v_k = \frac{A^k v_0}{\|A^k v_0\|} \to g, \quad v_k^* A v_k \to \lambda_0, \quad k \to \infty.$$

Moreover,

$$\lim_{n\to\infty} \frac{A^n v_0}{A^{n-1} v_0} = \lambda_0,$$

where for given vectors u and v, the ratio u/v is understood as the quotient function of the functions u and v.

Proof Suppose that the eigenvalues are all different for simplicity. Otherwise, one simply uses the Jordan representation of matrices. Write

$$v_0 = \sum_{j=0}^{N} c_j g_j$$

for some constants (c_j) with $g_0 = g$. Then $c_0 \neq 0$ by assumption and

$$A^k v_0 = \sum_{j=0}^{N} c_j \lambda_j^k g_j = c_0 \lambda_0^k \bigg[g + \sum_{j=1}^{N} \frac{c_j}{c_0} \Big(\frac{\lambda_j}{\lambda_0}\Big)^k g_j \bigg].$$

Since $|\lambda_j/\lambda_0| < 1$ for each $j \geqslant 1$ and $\|g\| = 1$, we have

$$\frac{A^k v_0}{\|A^k v_0\|} \to \frac{c_0}{|c_0|} g, \quad k \to \infty,$$

and then

$$v_k^* A v_k \to g^* A g = g^* \lambda_0 g = \lambda_0, \quad k \to \infty.$$

We have thus proved the main assertion of the lemma. The proof of the last assertion is similar. □

Clearly, the convergence speed in the lemma is

$$\Big|\frac{\lambda_1}{\lambda_0}\Big|^k, \quad |\lambda_1| := \max\{|\lambda_j| \colon j > 0\}.$$

The next result is the convergence for the inverse iteration.

Lemma 6 *Under the assumption of the last lemma, for each z close to λ_0, we have*

$$v_k = \frac{(A - zI)^{-k} v_0}{\|(A - zI)^{-k} v_0\|} \to g, \quad v_k^* A v_k \to \lambda_0, \quad k \to \infty.$$

Moreover,

$$\lim_{n\to\infty} \frac{(A - zI)^{-n} v_0}{(A - zI)^{-n+1} v_0} = \frac{1}{\lambda_0 - z}.$$

Proof Note that for z close to λ_0, the dominant eigenvalue of the matrix $(A - zI)^{-1}$ is $(\lambda_0 - z)^{-1}$ with the same dominant eigenvector g as the one for A. The proof is very much the same as the previous one. □

The iteration given in Lemma 6 is called the inverse iteration. It is remarkable that the convergence speed in this lemma is

$$\Big|\frac{\lambda_0 - z}{\lambda_1 - z}\Big|^k \sim \Big|\frac{\lambda_0 - z}{\lambda_1 - \lambda_0}\Big|^k$$

when z is sufficiently close to λ_0. At the kth step, replacing z by the Rayleigh quotient approximation $z_k = v_k^* A v_k$, we obtain the Rayleigh quotient iteration as described in the first section. Clearly, the last algorithm is an acceleration of the inverse iteration. The price is that the initial z_0 has to be chosen close to λ_0 which is usually not explicitly known. Otherwise, if z_0 is chosen close to some $\lambda_j \neq \lambda_0$, then a similar proof of Lemma 6 shows that $v_k^* A v_k$ converges to the pitfall $\lambda_j \neq \lambda_0$. In practice, once $z = z_0$ is chosen in a suitable neighborhood of λ_0, the sequence $z = z_k$ converges to λ_0 rapidly, as illustrated by Examples 2 and 4. More precisely, Example 1 applies the power iteration to $A := 113I + Q$, its convergence speed is

$$\sim \Big(\frac{113-\lambda_1}{113-\lambda_0}\Big)^k \approx \Big(\frac{113-2.00758}{113-0.525268}\Big)^k, \quad k\to\infty.$$

Examples 2 and 4 use the Rayleight quotient iteration which has the convergence speed

$$\sim \prod_{j=0}^{k} \frac{\lambda_0 - z_j}{\lambda_1 - z_j}, \quad k\to\infty.$$

Since $z_k \to \lambda_0$, the last convergence is much fast than the previous one. Honestly, this still does not answer the reason why the inverse algorithm in Example 2 can achieve the six significant digits at the second iteration.

3 Efficient initials. Tridiagonal case

Again, assume that $A = (a_{ij})$ on $E = \{0, 1, \ldots, N\}$, $N < \infty$, is irreducible and having non-negative off-diagonal elements. Assume also that the matrix is tridiagonal (after a suitable relabeling if necessary): $a_{ij} = 0$ unless $|i-j| \leqslant 1$. By a shift $Q := A - mI$ if necessary, where I is the identity matrix on E and

$$m = \max_{i\in E} \sum_{j\in E} a_{ij},$$

one may assume that

$$Q = \begin{pmatrix} -(b_0+c_0) & b_0 & 0 & 0 & \cdots \\ a_1 & -(a_1+b_1+c_1) & b_1 & 0 & \cdots \\ 0 & a_2 & -(a_2+b_2+c_2) & b_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & a_N & -(a_N+c_N) \end{pmatrix},$$

where $a_i, b_i > 0$, $c_i \geqslant 0$ but $c_i \not\equiv 0$. Define

$$\mu_0 = 1, \quad \mu_n = \mu_{n-1}\frac{b_{n-1}}{a_n} = \frac{b_0 b_1 \cdots b_{n-1}}{a_1 a_2 \cdots a_n}, \quad 1 \leqslant n \leqslant N.$$

We now split our discussion into two cases.

Case 1 Let

$$c_0 = c_1 = \cdots = c_{N-1} = 0.$$

Then we may assume that $c_N > 0$. Otherwise, Q has the trivial maximal eigenvalue 0 with eigenvector with components being one everywhere. In this case, we rewrite c_N as b_N, ignoring the sequence (c_i), and define

$$\varphi_n = \sum_{k=n}^{N} \frac{1}{\mu_k b_k}, \quad 0 \leqslant n \leqslant N. \tag{3}$$

Case 2 Let some of c_i $(i = 0, 1, \ldots, N-1)$ be positive. Then, we need more work. Define

$$r_0 = 1 + \frac{c_0}{b_0}, \quad r_n = 1 + \frac{a_n + c_n}{b_n} - \frac{a_n}{b_n r_{n-1}}, \quad 1 \leqslant n < N,$$

$$h_0 = 1, \quad h_n = h_{n-1} r_{n-1} = \prod_{k=0}^{n-1} r_k, \quad 1 \leqslant n \leqslant N,$$

and additionally,

$$h_{N+1} = c_N h_N + a_N (h_{N-1} - h_N).$$

Finally, define

$$\varphi_n = \sum_{k=n}^{N} \frac{1}{h_k h_{k+1} \mu_k b_k}, \quad 0 \leqslant n \leqslant N, \tag{4}$$

with a convention that $b_N = 1$ to save our notation.

We remark that in the special case that $c_0 = c_1 = \cdots = c_{N-1} = 0$, by induction, it is easy to check that

$$r_0 = r_1 = \cdots = r_{N-1} = 1,$$

and hence,

$$h_0 = h_1 = \cdots = h_N = 1.$$

Furthermore, $h_{N+1} = c_N$. Thus, once replacing c_N by b_N, we return to (3) from (4).

To state our algorithm, we need one more quantity:

$$\delta_1 = \max_{0 \leqslant n \leqslant N} \left[\sqrt{\varphi_n} \sum_{k=0}^{n} \mu_k h_k^2 \sqrt{\varphi_k} + \frac{1}{\sqrt{\varphi_n}} \sum_{j=n+1}^{N} \mu_j h_j^2 \varphi_j^{3/2} \right].$$

Rayleigh quotient iteration in tridiagonal case *For given tridiagonal matrix A, define m, (a_i, b_i, c_i), (h_i), (φ_i), and δ_1 as above. Set*

$$\widetilde{v}_0(i) = h_i \sqrt{\varphi_i}, \quad 0 \leqslant i \leqslant N, \quad v_0 = \frac{\widetilde{v}_0}{\sqrt{\widetilde{v}_0^* \widetilde{v}_0}}, \quad z_0 = \frac{1}{\delta_1}.$$

At the kth step $(k \geqslant 1)$, solve the linear equation in w_k:

$$(-Q - z_{k-1} I) w_k = v_{k-1}, \tag{5}$$

and define

$$v_k = \frac{w_k}{\sqrt{w_k^* w_k}}, \quad z_k = v_k^* (-Q) v_k.$$

Then v_k converges to g and $m - z_k$ converges to $\rho(A)$ as $k \to \infty$.

It is an essential point that the choice of z_0 avoids the collapse since we have known that $\lambda_0(Q) = \lambda_{\min}(-Q)$ (the minimal eigenvalue of $-Q$) $\geqslant \delta_1^{-1}$ by [5, Corollary 3.3]. As an application of this result to Example 1, we have $c_i \equiv 0$ but $b_7 = 64$ and then $h_i \equiv 1$. We can define φ by (3) and then $\widetilde{v}_0 = \sqrt{\varphi}$ which is the one, up to a free factor $\sqrt{\varphi_0}$, used in Example 1. This is the meaning of "very good" claimed in the first section. We now compute the minimal eigenvalue of $-Q$ using not only $\widetilde{v}_0$ but also δ_1.

Example 7 The matrix Q and the vector $\widetilde{v}_0$ are the same as in Example 1:

$$(1,\ 0.587624,\ 0.426178,\ 0.329975,\ 0.260701,\ 0.204394,\ 0.153593,\ 0.101142)^*.$$

We have $\delta_1 = 2.05768$. Then, with the new $z_0 := \delta_1^{-1} \approx 0.485985$, the Rayleigh quotient iteration arrives at the expected estimate at the second step:

$$z_1 \approx 0.525313, \quad z_2 \approx 0.525268.$$

Comparing the approximation value of z_1 here and that in Example 2, it is clear that this result is sharper than Example 2 (see also the comment below Corollary 12).

Now, let us discuss the effectiveness of our algorithm with respect to the size N of the matrix. In computational mathematics, one often expects the number of iterations M grows up no more than N^α for some $\alpha > 0$. It is unusual if $M \approx \log N$ for large N. To this question, considering the basic Example 1 with varying N, the answer given below is worked out by Yue-Shuang Li using the algorithm introduced in this section and the software MatLab on a notebook. In the first line of Table 2, the reason we use $N+1$ rather than N is that the space is labeled starting at 0 but not 1.

Table 2 For different N, eigenvalue λ_0, its lower bound δ_1^{-1}, and z_1, z_2

$N+1$	$z_0 = \delta_1^{-1}$	z_1	$z_2 = \lambda_0$
100	0.348549	0.376437	0.376383
500	0.310195	0.338402	0.338329
1000	0.299089	0.32732	0.32724
5000	0.281156	0.308623	0.308529
7500	0.277865	0.305016	0.304918
10000	0.275762	0.30266	0.302561

Is it believable? Yes, we have justified the outputs in two different ways: in each case, first, the outputs starting from z_2 become the same (which actually coincides with the output of λ_0). Second, by using v_2, we can find upper/lower estimates $\overline{\xi}/\underline{\xi}$ of λ_0 such that $z_2 \in (\underline{\xi}, \overline{\xi})$, and moreover,

$$\frac{\overline{\xi}}{\underline{\xi}} \approx 1 + 10^{-5}.$$

The next example is due to Hua [11] in the study of economic optimization (cf. [4, Chapter 10]). Note that here we are studying the right-eigenvector, the matrix A used below is the transpose of the original one.

Example 8 *Let*

$$A = \frac{1}{100}\begin{pmatrix} 25 & 40 \\ 14 & 12 \end{pmatrix}.$$

Then

$$\rho(A) = \frac{37 + \sqrt{2409}}{200} \approx 0.430408.$$

With the initials:

$$v_0 \approx (0.429166,\ 0.220573)^*, \quad z_0 := \delta_1^{-1} \approx 0.212077,$$

the iteration arrives at the expected result at the second step $(n = 2)$*:*

$$0.65 - z_0 \approx 0.437923; \quad 0.65 - z_1 \approx 0.430603; \quad 0.65 - z_2 \approx 0.430408.$$

Proof First, we have $m = 65/100$ and then

$$Q = \begin{pmatrix} -\dfrac{2}{5} & \dfrac{2}{5} \\ \dfrac{7}{50} & -\dfrac{53}{100} \end{pmatrix}.$$

In this case, we ignore (c_i) but let $b_1 > 0$. Actually, we have

$$b_0 = \frac{2}{5}, \quad b_1 = \frac{39}{100}; \quad a_1 = \frac{7}{50}; \quad \mu_0 = 1,\ \mu_1 = \frac{20}{7}; \quad \varphi_0 = \frac{265}{78},\ \varphi_1 = \frac{35}{39}.$$

Therefore,

$$v_0 = \left(\sqrt{\frac{53}{67}}, \sqrt{\frac{14}{67}}\right), \quad z_0^{-1} = \frac{5(2809 + 40\sqrt{742}\,)}{4134}.$$

The conclusion now follows by our algorithm. □

An additional example for the algorithm presented in this section is delayed to Example 22.

Before moving further, let us introduce an algorithm for (and then a representation of) the solution to equation (5). This is mainly used in theoretic analysis rather than numerical computation. The idea is meaningful in a more general setup and comes from [9, Theorem 1.1, Proposition 2.6] plus a modification [6, Proposition 4.1]. Given a number $z \in \mathbb{R}$ and a vector v, consider the equation for the vector w:

$$Qw + zw = -v. \tag{6}$$

To do so, we need some notation. Fix i: $0 \leqslant i \leqslant N-1$, and set

$$\alpha_\ell^{(i)} = \frac{1}{b_{i+\ell}}\begin{cases} c_{i+\ell} - z + a_{i+\ell}, & 1 = \ell \leqslant N - i, \\ c_{i+\ell} - z, & 2 \leqslant \ell \leqslant N - i. \end{cases}$$

Next, define the vector $G^{(i)}_{\cdot,1}$ by $G^{(i)}_{\ell,1} = \alpha^{(i)}_{\ell}$ for $\ell = 1, 2, \ldots, N-i$ and define recursively in $k = 2, 3, \ldots, N-i$, the vector $G^{(i)}_{\cdot,\,k}$ by

$$G^{(i)}_{\ell,k} = G^{(i)}_{\ell,\,k-1} + \alpha^{(i+k-1)}_{\ell-k+1} G^{(i)}_{k-1,\,k-1}, \quad \ell = k, k+1, \ldots, N-i. \tag{7}$$

Note that here for computing $G^{(i)}_{\cdot,\,k}$, we use only $G^{(i)}_{\cdot,\,k-1}$ but not the others $G^{(i)}_{\cdot,\,j}$ with $j \leqslant k-2$.

Proposition 9 *Let $N \geqslant 1$ and $G^{(\cdot)}_{0,0} \equiv 1$. Then the solution $w = (w_k\colon k \in E)$ to equation* (6) *has the following representation*:

$$w_n = \frac{v_N + M_{N-1}(v)}{c_N - z + M_{N-1}(c_{\cdot} - z)}\,[1 + N_{n-1}(c_{\cdot} - z)] - N_{n-1}(v), \quad 0 \leqslant n \leqslant N,$$

where for each vector h, $N_{-1}(h) = 0$ and

$$M_{N-1}(h) = c_N \sum_{j=0}^{N-1} \frac{h_j}{b_j}\, G^{(j)}_{N-j,N-j},$$

$$N_n(h) = \sum_{j=0}^{n} \frac{h_j}{b_j} \sum_{k=0}^{n-j} G^{(j)}_{k,\,k}, \quad 0 \leqslant n < N.$$

The proof of this result is delayed to Section 5.

From now on, we are going to treat general real matrices. This is a hard task and will be the main goal of the next section. Here, we study a special case only. In computational mathematics, there is a well-known Lanczos tridiagonalization procedure making a matrix to be tridiagonal one. That is, for a given A, constructing a nonsingular B such that $B^{-1}AB =: T$ becomes a tridiagonal matrix. We will come back to the procedure soon. Here is an example (the details are omitted).

Example 10 Let

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 1 \\ 3 & 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{10} & 3/\sqrt{13} \\ 0 & 3/\sqrt{10} & -2/\sqrt{13} \end{pmatrix}.$$

Then

$$T = B^{-1}AB = \begin{pmatrix} 1 & 11/\sqrt{10} & 0 \\ \sqrt{10} & 25/11 & 20\sqrt{130}/143 \\ 0 & \sqrt{130}/11 & 8/11 \end{pmatrix}.$$

We have

$$\rho(A) = \rho(T) = 3 + \sqrt{5} \approx 5.23607.$$

Our algorithm arrives at the same result at the second step of the iterations $(n = 2)$:

$$(n = 0)\ 5.43937; \quad (n = 1)\ 5.23996; \quad (n = 2)\ 5.23607.$$

It is the position to recommend an improved algorithm as follows. The point is to use the inner product $(\cdot,\cdot)_\mu$ and norm $\|\cdot\|_\mu$ in the space $L^2(\mu)$ since (μ_k) may not be a constant as in Example 1.

Improved algorithm Given $\widetilde{v}_0$ and δ_1 as above, redefine $v_0=\widetilde{v}_0/\|\widetilde{v}_0\|_\mu$ and

$$z_0=\xi\delta_1^{-1}+(1-\xi)(v_0,-Qv_0)_\mu,\quad \xi\in[0,1].$$

For $k\geqslant 1$, define w_k as before but redefine

$$v_k=\frac{w_k}{\|w_k\|_\mu},\quad z_k=(v_k,-Qv_k)_\mu.$$

With $\xi=7/8$, Example 7 and Table 2 are improved as Table 2′.

Table 2′ Example 7 and Table 2 are improved using new z_0 with $\xi=7/8$

$N+1$	z_0	z_1	$z_2=\lambda_0$	upper/lower
8	0.523309	0.525268	0.525268	$1+10^{-11}$
100	0.387333	0.376393	0.376383	$1+10^{-8}$
500	0.349147	0.338342	0.338329	$1+10^{-7}$
1000	0.338027	0.327254	0.32724	$1+10^{-7}$
5000	0.319895	0.30855	0.308529	$1+10^{-7}$
7500	0.316529	0.304942	0.304918	$1+10^{-7}$
10000	0.31437	0.302586	0.302561	$1+10^{-7}$

The last column is the order of the ratio of the upper and lower bounds of λ_0 in terms of v_2, as will be explained below, above Example 13.

Table 3 gives two more examples.

Table 3 Outputs using improved z_0 with $\xi=7/8$

Example	z_0	z_1	$z_2=\lambda_0$
8	0.436733	0.430407	0.430408
10	5.36161	5.23578	5.23607

Appendix of Section 3 Algorithm for Lanczos tridiagonalization

For a given A, the aim is choosing a nonsingular Q such that

$$Q^{-1}AQ=T=\begin{pmatrix} c_1 & b_1 & \cdots & \cdots & 0\\ a_1 & c_2 & \ddots & & \vdots\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ \vdots & & \ddots & \ddots & b_{n-1}\\ 0 & \cdots & \cdots & a_{n-1} & c_n \end{pmatrix}.$$

Note that the notation here is somehow different from the other part of the paper. To do so, we use the following column partitionings:

$$Q=[q_1\mid q_2\mid\cdots\mid q_n],\quad (Q^{-1})^*=\widetilde{Q}=[\widetilde{q}_1\mid\widetilde{q}_2\mid\cdots\mid\widetilde{q}_n].$$

Let

$$q_0 = 0, \quad \widetilde{q}_0 = 0, \quad b_0 = 0, \quad a_0 = 0.$$

Choose unit vectors q_1 and $\widetilde{q}_1$ such that $\widetilde{q}_1^* q_1 = 1$. Define

$$\begin{aligned} c_k &= \widetilde{q}_k^* A q_k, \quad k \geqslant 1,\\ r_k &= (A - c_k I) q_k - a_{k-1} q_{k-1}, \quad k \geqslant 1,\\ \widetilde{r}_k &= (A - c_k I)^* \widetilde{q}_k - b_{k-1} \widetilde{q}_{k-1}, \quad k \geqslant 1,\\ b_k &= \|r_k\|_2, \quad a_k = \frac{\widetilde{r}_k^* r_k}{b_k}, \quad k \geqslant 1,\\ q_k &= \frac{r_{k-1}}{b_{k-1}}, \quad \widetilde{q}_k = \frac{\widetilde{r}_{k-1}}{a_{k-1}}, \quad k \geqslant 2. \end{aligned}$$

For Example 10, we simply choose

$$q = (1, 0, 0)^*, \quad \widetilde{q} = (1, 0, 0)^*.$$

Generally speaking, there is a question in choosing initial q_0 and $\widetilde{q}_0$. More generally, it should be meaningful to know for what A, the resulting matrix have positive a_k and b_k for every k.

4 Efficient initials. General case

A general algorithm for the efficient initials will be introduced later in the second subsection. The algorithm introduced in the next subsection is easier and quite general, but may be less efficient.

4.1 Fix uniformly distributed initial vector v_0

In this subsection, we fix the uniformly distributed initial vector

$$v_0 = \frac{(1, 1, \ldots, 1)}{\sqrt{N+1}}.$$

This is the easiest choice of v_0 since it does not use any information from the eigenvector g of $\rho(A)$ except its positivity property. On the other hand, this means that the choice is less efficient and it can be even broken as shown by Example 3. The effectiveness of this v_0 depends heavily on the choice of z_0. For which, here we introduce three effective choices.

Choice I Let $A = (a_{ij} \colon i, j \in E)$ be nonnegative and set $z_0 = \sup_{i\in E} A_i$, where $A_i = \sum_{j\in E} a_{ij}$. This universal choice comes from the fact that $\sup_{i\in E} A_i$ is an upper bound of $\rho(A)$, which can be seen by setting $x_i \equiv 1$ in the next result.

Proposition 11 *For a nonnegative irreducible matrix A with maximal eigenvalue $\rho(A)$, the Collatz–Wielandt formula holds*:

$$\sup_{x>0} \min_{i\in E} \frac{(Ax)_i}{x_i} = \rho(A) = \inf_{x>0} \max_{i\in E} \frac{(Ax)_i}{x_i}.$$

For the present (v_0, z_0), even though it is not necessary, one may replace (2) by

$$(z_{k-1}I - A)w_k = v_{k-1}. \tag{8}$$

This choice of z_0 avoids the collapse of the algorithm since

$$0 < z_0 - \rho(A) < |z_0 - \lambda|$$

for every eigenvalue $\lambda \neq \rho(A)$ of A.

Let us now introduce an important application of Proposition 11. First, if we replace A and $\rho(A)$ with $-Q$ and λ_0, respectively, the same conclusion holds, as shown in the next corollary (the proof is delayed to Section 5). Actually, the corollary holds in a much more general setup. Refer to [3, Theorem 9.5].

Corollary 12 *For Q-matrix, the Collatz–Wielandt formula becomes*

$$\sup_{x>0} \min_{i\in E} \frac{(-Qx)_i}{x_i} = \lambda_0(Q) = \inf_{x>0} \max_{i\in E} \frac{(-Qx)_i}{x_i}.$$

Thus, instead of the mean estimate given in these algorithm, we can produce pointwise estimates. To do so, we need only to compute the ratio $(-Q)v_k/v_k$. For instance, in Example 2, the ratio $(-Q)v_2/v_2$ is as follows:

0.525197, 0.5254, 0.52553, 0.525623, 0.525693, 0.525747, 0.525787, 0.525816.

Therefore, we obtain

$$0.525197 \leqslant \lambda_0 \leqslant 0.525816$$

and the ratio of the upper/lower bounds is ≈ 1.00118. Next, for Example 7, the ratio $(-Q)v_2/v_2$ is as follows:

0.525268, 0.525268, 0.525267, 0.525267, 0.525267, 0.525267, 0.525267, 0.525267.

Hence, we have

$$0.525267 \leqslant \lambda_0 \leqslant 0.525268$$

and the ratio of the upper/lower bounds is $\approx 1 + 10^{-6}$. Actually, if we apply the estimates given in [5, Theorem 2.4 (3)] (with $\operatorname{supp}(f) = E$)

$$z_2 \wedge \sup_{i\in E} \frac{f_i}{g_i} \geqslant \lambda_0 \geqslant \inf_{i\in E} \frac{f_i}{g_i},$$

$$g_i := \sum_{k\in E} \mu_k f_k \varphi_{i\vee k} = \varphi_i \sum_{k=0}^{i} \mu_k f_k + \sum_{k=i+1}^{N} \mu_k \varphi_k f_k,$$

$$\varphi_i := \sum_{k=i}^{N} \frac{1}{\mu_k b_k} \quad \text{(for this example, } \mu_k \equiv 1,\ b_i = (i+1)^2\text{)},$$

to the test function $f = v_2$ with a more precise output, the upper/lower bounds can be improved as $\approx 1 + 10^{-7}$. Hence, the estimate $\lambda_0 \approx 0.525268$ is indeed sharp up to the six precisely significant digits. This shows that the estimates in the latter example are much better than the former one.

Example 13 Let A be the same as in Example 10. Then $\rho(A) \approx 5.23607$ and $z_0 = 6$. The Rayleigh quotient iteration gives us

$$z_1 \approx 5.27273, \quad z_2 \approx 5.23639, \quad z_3 \approx 5.23607.$$

Example 14 Let

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}.$$

Then $\rho(A) \approx 36.2094$ and $z_0 = 58$. The Rayleigh quotient iteration gives us

$$z_1 \approx 37.3442, \quad z_2 \approx 36.2674, \quad z_3 \approx 36.2095, \quad z_4 \approx 36.2094.$$

Example 15 Let

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 3 & 14 & 11 & 0 \\ 9 & 10 & 11 & 1 \\ 5 & 6 & 7 & 8 \end{pmatrix}.$$

This matrix has complex eigenvalues:

$$24.0293, \ 7.72254, \ 1.1241 + 2.40522\,\mathrm{i}, \ 1.1241 - 2.40522\,\mathrm{i}.$$

Hence, $\rho(A) \approx 24.0293$ and $z_0 = 31$. The Rayleigh quotient iteration gives us

$$z_1 \approx 24.4393, \quad z_2 \approx 24.0385, \quad z_3 \approx 24.0293.$$

Example 16 Let Q be the same as in Example 1 and let

$$A = 113\,I + Q.$$

Then $z_0 = 113$. Recall that $\lambda_{\min}(-Q) \approx 0.525268$. For $k = 1, 2, 3$, the Rayleigh quotient iteration gives us $113 - z_k$ as follows:

$$113 - z_1 \approx 0.602312, \quad 113 - z_2 \approx 0.525463, \quad 113 - z_3 \approx 0.525268.$$

Alternatively, one may apply the algorithm directly to $-Q$ with $z_0 = 0$.

We remark that the algorithm is meaningful for any

$$z_0 \geqslant \sup_{i \in E} \sum_{j \in E} a_{ij}.$$

For instance, if we choose $z_0 = 200$ rather than $z_0 = 6$ used in Example 13, then the successive results of the iterations are as follows:

$$z_1 \approx 5.33546, \quad z_2 \approx 5.24182, \quad z_3 \approx 5.23608, \quad z_4 \approx 5.23607.$$

The convergence becomes slower as we can imagine. In other words, a larger initial z_0 is less efficient. In view of Proposition 11, we have

$$0 < \rho(A) \leqslant 113.$$

It seems that there is a large room for us to choose z_0. Yes or no? It is yes, since the last estimates are rather rough, each choice $z_0 \in [111.7, 113]$ is also available. The answer is also no, since if we choose $z_0 = 111.6$, then we will go to the pitfall λ_1 ($> \lambda_0$). Hence, it is rather sensitive to find a useful z_0 except Choice I. Noting that

$$\rho(A) \approx 113 - 0.525268,$$

the reason why the rough Choice I is still efficient for this model should be clear.

We have thus studied the model introduced in Example 1 six times with different initials. The results are collected in Table 4. Among them, the worst one is Example 3 and the best one is Example 7 which uses the whole power of the algorithm introduced in Section 3. The "Uniform" is the present Choice I and the "Auto" means automatic one given by the algorithm, as we will come back in Choice II below.

Table 4 Comparison of examples with different initials

same Q	v_0	z_0	# of iterations
Example 1	$\widetilde{v}_0$	power	10^3
Example 2	$\widetilde{v}_0$	auto	2
Example 3	uniform	auto	collapse
Example 4	uniform	δ_1^{-1}	2
Example 7	$\widetilde{v}_0$	δ_1^{-1}	2
Example 16	uniform	113	3

In conclusion, even though the present choice (v_0, z_0) may not be very efficient, but it works in a very general setup. This algorithm works even for a more general class of matrices, without assuming the nonnegative property, once you have an upper estimate of the largest eigenvalue of A. Clearly, for large-scale matrix, Choice I is meaningful only for the sparse ones.

Choice II Simply use the particular choice given in the Rayleigh quotient iteration: $z_0 = v_0^* A v_0$. This simple choice is quite natural and so is often used in practice. However, there is a dangerous here since v_0 is chosen roughly, the algorithm may lead to an incorrect limit, as illustrated by Example 3.

With the present z_0, the computation results for Examples 13–15 are listed in Table 5.

Table 5 Output (z_1, z_2, z_3) of Examples 13–15

Example	z_1	z_2	$z_3 = \lambda_0$
13	5.24183	5.23608	5.23607
14	35.8428	36.2127	36.2094
15	23.7316	24.0317	24.0293

Combining (z_1, z_2) here with those given in the last part, it is clear that the present choice of z_0, once works, is better than Choice I.

Choice III This is based on a comparison technique. For given $A = (a_{ij})$ having the property $a_{i,i+1} + a_{i+1,i} > 0$ for every i, we introduce the symmetrized matrix $(A + A^*)/2$. (This symmetrizing procedure may be omitted if both $a_{i,i+1} > 0$ and $a_{i+1,i} > 0$ for every i.) Denote by $(\alpha_i, \beta_i, \gamma_i)$ the tridiagonal part (where γ_i are the diagonal elements) taken from the symmetrized matrix. By assumption, we have $\alpha_i > 0$ and $\beta_i > 0$. We can then follow the last section to choose a z_0 first for the tridiagonal matrix and then regarding it as an approximation of z_0 for the original A. One may worry that we have lost too much in the last step. Yes, it may be so. However, the key is to avoid the collapse. The smaller estimate z_0 of $\lambda_{\min}(-Q)$ is not really serious since the algorithm can repair it rapidly, as shown by the next example.

Example 17 Let A be the same as in Example 15. Then

$$\frac{1}{2}(A + A^*) = \begin{pmatrix} 1 & 5/2 & 9/2 & 5/2 \\ 5/2 & 14 & 21/2 & 3 \\ 9/2 & 21/2 & 11 & 4 \\ 5/2 & 3 & 4 & 8 \end{pmatrix}.$$

From this, we obtain a tridiagonal matrix

$$T = \begin{pmatrix} 1 & 5/2 & 0 & 0 \\ 5/2 & 14 & 21/2 & 0 \\ 0 & 21/2 & 11 & 4 \\ 0 & 0 & 4 & 8 \end{pmatrix},$$

and then

$$Q = T - 27I = \begin{pmatrix} -26 & 5/2 & 0 & 0 \\ 5/2 & -13 & 21/2 & 0 \\ 0 & 21/2 & -16 & 4 \\ 0 & 0 & 4 & -19 \end{pmatrix}.$$

According to what we did in Section 3, we have $z_0 \approx 1/0.321526$ for $-Q$. Then, we have

$$z_0 \approx 27 - \frac{1}{0.321526}$$

for T. This is regarded as an approximation of z_0 for A. Starting from here and using the Rayleigh quotient iteration, we obtain the successive approximation of $\rho(A)$ as follows:

$$z_1 \approx 24.0125, \quad z_2 \approx 24.0293,$$

as we expected. Picking up the tridiagonal part directly from A (without using the symmetrizing procedure), the same approach leads to the following output:

$$z_0 \approx 28 - \frac{1}{0.23307} \approx 23.7094, \quad z_1 \approx 23.9901, \quad z_2 \approx 24.0293.$$

Let us remark that the three choices of z_0 in this subsection are independent of the initial v_0 used here and so can be also used in the next subsection. Certainly, there are other approaches can be used to deduce an approximation of the required z_0. For instance, Cheeger's approach [3, §9.5], which is meaningful in a very general setup. Since it takes account of all subset of E (except the emptyset), the number of computations is of order 2^N. This approach as well as the capacitary one (cf. [4, Chapter 7]) needs to be simplified to fit the present setup. In practice, one often uses Proposition 11 or Corollary 12 to get an upper/lower bound in terms of a suitable test sequence (x_i). Refer also to [4, Theorem 3.6] which uses test weights. These approaches depend heavily on the working models.

4.2 Efficient initial vector v_0

In general, it is much more difficult to choose an efficient initial v_0 than z_0. Here is our algorithm.

A general algorithm

Let $A = (a_{ij} \colon i, j \in E)$ be a given irreducible matrix having nonnegative off-diagonal elements. Once again, denote by $\rho(A)$ the maximal eigenvalue of A. If $A_i := \sum_{j\in E} a_{ij}$ is a constant (independent of $i \in E$), then we have $\rho(A) \equiv A_i$ with right-eigenvector $\mathbb{1}$ (its components are all equal to 1). From now on, we assume that A_i are not a constant.

We introduce our algorithm in four steps.

Step 1 When $A_i \leqslant 0$ for every $i \in E$, one can jump from here to Step 2 below by setting $Q = A$. Otherwise, let $\max_{i\in E} A_i > 0$. Define

$$Q = A - \big(\max_{i\in E} A_i\big) I.$$

Then the sum of each row of Q is less or equal to zero and at least one of the rows is less than zero since A_i is not a constant. Now, if

$$Q_0 = Q_1 = \cdots = Q_{N-1} = 0$$

but $Q_N < 0$ ($Q_k := \sum_j q_{kj}$), then one can jump from here to Step 3 with $h_i \equiv 1$.

Step 2 Assume that $Q_k < 0$ for some $k \leqslant N - 1$. Denote by

$$h = (h_0, h_1, \ldots, h_N)^*$$

with $h_0 = 1$ the solution to the equation

$$Q^{\setminus N\text{'s row}} h = 0,$$

where $Q^{\setminus k\text{'s row}}$ is obtained from Q removing its k's row $(q_{k0}, q_{k1}, \ldots, q_{kN})$. In the case that

$$c_N + \sum_{j\leqslant N-1} q_{Nj}\Big(1 - \frac{h_j}{h_N}\Big)$$

is much smaller than

$$\sum_{j\leqslant N-1} q_{Nj}\frac{h_j}{h_N}$$

(say, 1 : 100 for instance), one can jump from here to (10) with $x_i \equiv 1$ (cf. Example 21 in the case of $b_4 = 0.01$).

Step 3 Let $(h_i\colon i \in E)$ be constructed in the last step. Define $q_i = -q_{ii}$, $i \in E$. Let $x = (x_0, x_1, \ldots, x_N)^*$ (with $x_0 = 1$) be the solution to the equation

$$x^{\setminus \text{0's row}} = P^{\setminus \text{0's row}}\, x, \tag{9}$$

where

$$P = (p_{ij}\colon i, j \in E)\colon \quad p_{ii} = 0, \quad p_{ij} = \frac{q_{ij}h_j}{q_i h_i}, \quad j \neq i;$$

or in the matrix form,

$$P = \text{Diag}((q_i h_i)^{-1}) Q \text{Diag}(h_i) + I.$$

Refer to the comments below Examples 21 and 22 for the constraint $x_0 = 1$. Here, the sequence (x_i) is an extension of (φ_i) used in Section 3 (cf. Lemma 24 below).

Step 4 We are now ready to state our algorithm as follows. Define a (column) vector $\widetilde{v}_0$ with components

$$\widetilde{v}_0(i) = h_i\sqrt{x_i}, \quad i = 0, 1, \ldots, N. \tag{10}$$

Let

$$v_0 = \frac{\widetilde{v}_0}{\sqrt{\widetilde{v}_0^*\widetilde{v}_0}}, \quad z_0 = v_0^*(-Q)v_0.$$

In general, for $k \geqslant 1$, let w_k be the solution to the equation

$$(-Q - z_{k-1}I)w_k = v_{k-1},$$

and define

$$v_k = \frac{w_k}{\sqrt{w_k^* w_k}}, \quad z_k = v_k^*(-Q)v_k.$$

Then z_k and v_k are approximations of the minimal eigenvalue $\lambda_0 = \lambda_{\min}(-Q)$ of $-Q$ and its eigenvector, respectively. If we replace $-Q$ by A everywhere in this step, then the resulting z_k and v_k are approximations of $\rho(A)$ and its eigenvector g, respectively. Obviously, from Step 1, it follows that

$$\lambda_{\min}(-Q) + \rho(A) = \max_{i\in E} A_i.$$

Hence,

$$\lambda_0 = \lambda_{\min}(-Q) > \alpha \Longleftrightarrow \rho(A) \leqslant \max_{i\in E} A_i - \alpha.$$

This gives the relationship of a lower estimate of λ_0 and an upper estimate of $\rho(A)$.

Example 18 *Let A be given in Example* 10. *Then*

$$\rho(A) = 3 + \sqrt{5} \approx 5.23607.$$

Our algorithm here gives us

$$z_1 \approx 5.23883, \quad z_2 \approx 5.23607.$$

Proof Since $\max_i A_i = 6$, we have

$$Q = A - 6\,I = \begin{pmatrix} -5 & 2 & 3 \\ 1 & -4 & 1 \\ 3 & 2 & -5 \end{pmatrix}.$$

Next, we have

$$h_0 = 1, \quad h_1 = \frac{4}{7}, \quad h_2 = \frac{9}{7},$$

and

$$x_0 = 1, \quad x_1 = \frac{7}{9}, \quad x_2 = \frac{49}{81}.$$

From these, we obtain

$$\widetilde{v}_0 = (1,\, h_1\sqrt{x_1},\, h_2\sqrt{x_2}\,)^* = \Big(1,\, \frac{4}{3\sqrt{7}},\, 1\Big)^*.$$

Now, with

$$v_0 = \frac{\widetilde{v}_0}{\sqrt{\widetilde{v}_0^*\widetilde{v}_0}}, \quad z_0 = v_0^* A v_0 \approx 5.11616,$$

we can apply the Rayleigh quotient iteration in two steps to obtain the conclusion. □

Example 19 *Let A be the same as in Example* 14. *Then $\rho(A) \approx 36.2094$. By using* (10),

$$v_0 = (0.348213,\, 0.244601,\, 0.389728,\, 0.816719)^*,$$

the Rayleigh quotient iteration starts at $z_0 \approx 34.4924$ and gives us

$$z_1 \approx 36.1469, \quad z_2 \approx 36.2095, \quad z_3 \approx 36.2094.$$

Proof We have

$$Q = A - 58I = \begin{pmatrix} -57 & 2 & 3 & 4 \\ 5 & -52 & 7 & 8 \\ 9 & 10 & -47 & 12 \\ 13 & 14 & 15 & -42 \end{pmatrix}.$$

Next, we have

$$h_0 = 1, \quad h_1 = \frac{59}{27}, \quad h_2 = \frac{91}{27}, \quad h_3 = \frac{287}{27}.$$

Furthermore, we have

$$x_0 = 1, \quad x_1 = \frac{189}{1829}, \quad x_2 = \frac{7155}{64883}, \quad x_3 = \frac{243}{4991}.$$

Then the conclusion follows from the iteration. □

Example 20 *Let A be the same as in Example* 15. *Then* $\rho(A) \approx 24.0293$. *By using the algorithm in Section* 4.2, *the Rayleigh quotient iteration starts at* $31 - z_0 \approx 22.6424$ *and gives us for* $k = 1, 2, 3$,

$$31 - z_k \approx 24.1046, \quad 24.0298, \quad 24.0293,$$

respectively.

Proof We have

$$Q = A - 31I = \begin{pmatrix} -30 & 2 & 0 & 0 \\ 3 & -17 & 11 & 0 \\ 9 & 10 & -20 & 1 \\ 5 & 6 & 7 & -23 \end{pmatrix}.$$

Then, we have

$$h_0 = 1, \quad h_1 = 15, \quad h_2 = \frac{252}{11}, \quad h_3 = \frac{3291}{11};$$
$$x_0 = 1, \quad x_1 = \frac{3691}{76575}, \quad x_2 = \frac{1694}{45945}, \quad x_3 = \frac{7447}{3360111};$$
$$v_0 = (0.140655,\ 0.463208,\ 0.61873,\ 0.61873).$$

The conclusion follows by the algorithm. □

It is interesting to compare this example with Examples 15 and 17.

Actually, to show that our algorithm is reasonable, one may ignore the part using the H-transform and jump to the last step on Q-matrix since the transform does not change the spectrum. Thus, one needs to compare the maximal eigenvector g and its approximation (x_i). As mentioned before, this depends heavily on the rate $b_N = c_N$. Here is an example of sparse matrix.

Example 21 Let

$$Q = \begin{pmatrix} -3 & 2 & 0 & 1 & 0 \\ 4 & -7 & 3 & 0 & 0 \\ 0 & 5 & -5 & 0 & 0 \\ 10 & 0 & 0 & -16 & 6 \\ 0 & 0 & 0 & 11 & -11 - b_4 \end{pmatrix}.$$

Corresponding to different b_4, the maximal eigenvector g (normalized so that the first component to be one) and its approximation $(\sqrt{x_i}\,)$ (up to a positive constant) are given in Table 6.

Table 6 For different b_4, vectors g and $(\sqrt{x_i}\,)$ (Example 21)

b_4	g	$\sqrt{x}$ up to a constant
0.01	(1, 1.00011, 1.00017, 0.999498, 0.998616)*	(1, 1, 1, 0.999728, 0.999274)*
1	(1, 1.00992, 1.0149, 0.955637, 0.877794)*	(1, 1, 1, 0.9759, 0.934353)*
100	(1, 1.08011, 1.1211, 0.656961, 0.0652116)*	(1, 1, 1, 0.805682, 0.253629)*

The corresponding output of our algorithm is given in Table 7.

Table 7 For different b_4, eigenvalue λ_0 and z_1, z_2, z_3 (Example 21)

b_4	z_1	z_2	$z_3 = \lambda_0$
0.01	0.000278573	0.000278686	
1	0.0236258	0.0245174	0.0245175
100	0.200058	0.182609	0.182819

Our original purpose to design the Q-matrix in the last example is for a test of sparse matrix. The solution $x_0 = x_1 = x_2 = 1$ leads us to think about the transition machinery of the Q-matrix. Here is the graphic structure of the Q-matrix:

② ⇆ ① ⇆ ⓪ ⇆ ③ ⇆ ④.

As we will see at the end of Section 5, x_i is the probability of the process first hitting 0 starting from i (which is exactly the probabilistic meaning of the construction of (x_i) given in our general algorithm). Now, starting from 2, there is only one way to go to 0, and hence x_2 should be equal to 1. So does x_1. From this graph, it follows that the matrix is indeed tridiagonal after a relabeling (simply exchange the labels ② and ⓪):

⓪ ⇆ ① ⇆ ② ⇆ ③ ⇆ ④.

As a comparison, we present the next result using the algorithms given in Sections 4 and 3, respectively.

Example 22 Let

$$Q = \begin{pmatrix} -5 & 5 & 0 & 0 & 0 \\ 3 & -7 & 4 & 0 & 0 \\ 0 & 2 & -3 & 1 & 0 \\ 0 & 0 & 10 & -16 & 6 \\ 0 & 0 & 0 & 11 & -11-b_4 \end{pmatrix}.$$

Corresponding to different b_4, the maximal eigenvector g and its approximation $(\sqrt{x_i}\,)$ are given in Table 8.

Table 8 For different b_4, vectors g and $(\sqrt{x_i}\,)$ (Example 22)

b_4	g	$\sqrt{x}$ up to a constant
0.01	$(1, 0.999944, 0.999833, 0.999331, 0.998449)^*$	$(1, 0.999819, 0.999682, 0.99941, 0.998956)^*$
1	$(1, 0.995096, 0.98532, 0.941608, 0.864908)^*$	$(1, 0.984848, 0.973329, 0.949871, 0.909433)^*$
100	$(1, 0.963436, 0.89198, 0.585996, 0.0581675)^*$	$(1, 0.91325, 0.842344, 0.678661, 0.213643)^*$

The corresponding output (z_k) of the algorithm in Section 4 is given in Table 9.

Table 9 For different b_4, eigenvalue λ_0 and z_1, z_2, z_3 (Example 22)

b_4	z_1	z_2	$z_3 = \lambda_0$
0.01	0.000278548	0.000278686	
1	0.0234222	0.0245174	0.0245175
100	0.13342	0.182541	0.182819

The output (z_k) of the algorithm in Section 3 is given in Table 10.

Table 10 For different b_4, eigenvalue λ_0, its lower bound δ_1^{-1} and z_1, z_2 (Example 22)

b_4	$z_0 = \delta_1^{-1}$	z_1	$z_2 = \lambda_0$
0.01	0.00027867	0.000278686	
1	0.0244003	0.024519	0.0245175
100	0.179806	0.182912	0.182819
10^6	0.191917	0.195239	0.195145

Once again, one sees the efficiency of our algorithm.

Comparing the last two examples, especially their g and $\sqrt{x_i}$, it is obvious that the latter is better than the former one. This suggests us to choose the starting point 0 carefully. Here is an easier way to do so. First, define a sequence $\{E_\ell\}$ of level sets as follows. Let $E_0 = \{N\}$ and $E_1 = \{i \in E \colon a_{iN} > 0\}$. At the kth step, set

$$E_k = \{i \in E \setminus (E_0 + E_1 + \cdots + E_{k-1}) \colon \exists\, j \in E_{k-1} \text{ such that } a_{ij} > 0\}.$$

The procedure should be stopped at m if $E_{m+1} = \emptyset$. Because of the irreducibility, each $i \in E$ should belong to one of the level sets. Finally, regard one of $i_m \in E_m$ satisfying

$$a_{i_m j_{m-1}} = \min\{a_{ij} \colon i \in E_m,\ j \in E_{m-1}\}$$

as our initial 0. However, for initial $\widetilde{v}_0$, in practice, it is not necessary to relabeling the states as we did in Example 22. What we need is only replace the constraint $x_0 = 1$ by $x_{i_m} = 1$ (at the same time, "removing the first line" is replaced by "removing the i_m's line" in constructing the required matrix) in solving (x_i) without change the original matrix A or Q. One may need the relabeling in computing δ_1 defined in Section 3.

To conclude this subsection, we introduce a new construction of z_0 based on v_0 defined by our general algorithm. It is an extension of $z_0 = \delta_1^{-1}$ given in Section 3. To do so, we use Q, (h_i), and (x_i) defined at the beginning of this subsection. Let $\widetilde{Q}_0$ be the matrix obtained from

$$\widetilde{Q} := \mathrm{Diag}(h_i)^{-1} Q \mathrm{Diag}(h_i)$$

by modifying the last diagonal element $\widetilde{q}_{N,N}$ so that the sum of its last row becomes zero (i.e., removing the killing c_N). Next, let $\mu := (\mu_0, \mu_1, \ldots, \mu_N)$ with $\mu_0 = 1$ be the solution to the equation

$$\mu \widetilde{Q}_0 = 0.$$

Since there are only N variables $\mu_1, \mu_2, \ldots, \mu_N$, one may get the solution μ from the equation

$$\widetilde{Q}^{*\,\backslash \text{the last row}} \mu^* = 0.$$

Here, we remark that for a large class of Q-matrix Q, there is an explicit representation of μ in terms of the non-diagonal elements of Q, refer to [3, Chapter 7]. Now, our new initial z_0 is defined to be δ_1^{-1}:

$$\delta_1 = \frac{1}{1-x_1} \max_{0 \leqslant n \leqslant N} \left[\sqrt{x_n} \sum_{k=0}^{n} \mu_k \sqrt{x_k} + \frac{1}{\sqrt{x_n}} \sum_{n+1 \leqslant j \leqslant N} \mu_j x_j^{3/2} \right]. \tag{11}$$

In contrast to the above examples which use only the automatic $z_0 = v_0^* A v_0$ (or $z_0 = v_0^*(-Q)v_0$), here we use (11). Remember that this initial z_0 is for $-Q$, when we go back to the original A, its initial becomes $\max_{i \in E} \sum_{j \in E} a_{ij} - z_0$. The outputs of Examples 18–20 using δ_1^{-1} are listed in Table 11.

Table 11 Outputs of Examples 18–20 using δ_1^{-1}

Example	z_0	z_1	z_2	$z_3 = \lambda_0$
18	5.90016	5.22268	5.23611	5.23607
19	57.2719	36.236	36.2097	36.2094
20	30.3886	23.7436	24.0347	24.0293

Finally, we have an improved algorithm (for Q) as stated in Section 3 (below Example 10) based on the use of $L^2(\mu)$ and the convex combination:

$$z_0 = \xi \delta_1^{-1} + (1-\xi)(v_0, -Qv_0)_\mu, \quad \xi \in [0,1].$$

The outputs of Examples 18–20 using the new z_0 with $\xi = 1/3$ are listed in Table 12.

Table 12 Outputs of Examples 18–20 using new z_0 with $\xi = 1/3$

Example	z_0	z_1	z_2	$z_3 = \lambda_0$
18	5.04169	5.24358	5.23608	5.23607
19	35.4952	36.2657	36.2095	36.2094
20	24.0583	24.0213	24.0293	

This combination becomes more serious when N is large since in that case $(v_0, -Qv_0)_\mu$ is often an upper bound of λ_0, which may be much closer to other $\lambda_j \neq \lambda_0$ and so the algorithm would converge to λ_j but not λ_0. Certainly, the convex combination idea is also meaningful for the first two choices of z_0 introduced in the first subsection.

5 Additional remarks and proofs

In this section, we first prove a new result related to our earlier study. Then we present some proofs of the results given in the last two sections. Finally, we will make some remarks on the results studied so far in the previous sections.

The next result solves an open question kept in our mind for many years. For a given birth–death matrix Q on E with $c_0 = c_1 = \cdots = c_{N-1} = 0$ and $b_N := c_N > 0$, and a positive function f on E, define

$$I\!I(f)(i) = \frac{1}{f_i} \sum_{j=i}^{N} \frac{1}{\mu_j b_j} \sum_{k=0}^{j} \mu_k f_k, \quad i \in E.$$

Proposition 23 *For Q and $I\!I$ given above, let f_1 (> 0 on E) be arbitrarily given function and define successively $f_{n+1} = f_n I\!I(f_n)$. Then this algorithm coincides with the inverse iteration given in Lemma 6 with $z = 0$, even for infinite N. Furthermore, we have*

$$\lambda_0 = \lambda_{\min}(-Q) = \lim_{n\to\infty} I\!I(f_n)(i)^{-1}$$

for each $i \in E$. In particular, we have

$$\lim_{n\to\infty} \min_{i\in E} I\!I(f_n)(i) = \frac{1}{\lambda_0} = \lim_{n\to\infty} \max_{i\in E} I\!I(f_n)(i).$$

Proof Consider the Poisson equation: $-Qf = g$ for a given g. The solution is given by $f = gI\!I(g)$ ([5, (2.7)–(2.9)]). It can be also written as $f = (-Q)^{-1}g$. By setting $g = f_1$ and $f = f_2$, it follows that

$$f_2 = (-Q)^{-1} f_1 = f_1 I\!I(f_1).$$

Now, by iteration, we get

$$f_{n+1} = (-Q)^{-n} f_1 = f_n I\!I(f_n), \quad n \geqslant 1.$$

We have thus proved the first assertion. Therefore,

$$I\!I(f_n) = \frac{f_{n+1}}{f_n} = \frac{(-Q)^{-n}(f_1)}{(-Q)^{-n+1}(f_1)} \to \frac{1}{\lambda_0}, \quad n \to \infty,$$

by the last assertion of Lemma 6 with $z = 0$. The last assertion of the proposition then follows since on a finite set, the pointwise convergence implies the uniform one. □

We remark that the last proposition is meaningful once the Poisson equation $-Qf = g$ is solvable. In parallel, Lemma 6 improves the approximating procedures studied in [5] and related publications.

Now, we turn to prove Proposition 9 and Corollary 12.

Proof of Proposition 9 (a) First, we follow the setup and notation in [9] (where a more general situation is studied) for a moment. Define

$$M_{N-1}(h) = \sum_{k=0}^{N-1} \widetilde{q}_N^{(k)} \sum_{j=0}^{k} \frac{\widetilde{F}_k^{(j)} h_j}{q_{j,j+1}},$$

$$N_n(h) = \sum_{k=0}^{n} \sum_{j=0}^{k} \frac{\widetilde{F}_k^{(j)} h_j}{q_{j,j+1}}, \quad 0 \leqslant n < N.$$

Then the solution given in [9, Proposition 2.6] can be rewritten as

$$g_n = \frac{f_N + M_{N-1}(f)}{c_N + M_{N-1}(c.)} [1 - N_{n-1}(c.)] + N_{n-1}(f), \quad N_{-1} := 0, \quad 0 \leqslant n \leqslant N.$$

By an exchange of the order of the summations, we can rewrite M_n and N_n as follows:

$$M_{N-1}(h) = \sum_{j=0}^{N-1} \frac{h_j}{q_{j,j+1}} \sum_{k=j}^{N-1} \widetilde{q}_N^{(k)} \widetilde{F}_k^{(j)},$$

$$N_n(h) = \sum_{j=0}^{n} \frac{h_j}{q_{j,j+1}} \sum_{k=j}^{n} \widetilde{F}_k^{(j)}, \quad 0 \leqslant n < N.$$

Here, for finite N, the element $q_{N,N+1}$ is replaced by c_N by our convention. Thus, by [9, (1.1)], we get

$$M_{N-1}(h) = c_N \sum_{j=0}^{N-1} \frac{h_j}{q_{j,j+1}} \widetilde{F}_N^{(j)}.$$

By [6, Proposition 4.1], we have $\widetilde{F}_{i+m}^{(i)} = G_{m,m}^{(i)}$. It follows that

$$M_{N-1}(h) = c_N \sum_{j=0}^{N-1} \frac{h_j}{q_{j,\,j+1}} G_{N-j,N-j}^{(j)},$$

$$N_n(h) = \sum_{j=0}^{n} \frac{h_j}{q_{j,j+1}} \sum_{k=0}^{n-j} G_{k,k}^{(j)}, \quad 0 \leqslant n < N.$$

Applying this solution to the birth–death context and setting $f = -v$, $g = w$, replacing the original $c.$ used in [9] by $z - c.$, we obtain

$$g_n = \frac{-v_N - M_{N-1}(v)}{z - c_N + M_{N-1}(z - c.)} [1 - N_{n-1}(z - c.)] - N_{n-1}(v), \quad 0 \leqslant n \leqslant N.$$

Equivalently,

$$g_n = \frac{v_N + M_{N-1}(v)}{c_N - z + M_{N-1}(c. - z)} [1 + N_{n-1}(c. - z)] - N_{n-1}(v), \quad 0 \leqslant n \leqslant N.$$

This gives us the required conclusion. □

Proof of Corollary 12 The proof is quite straightforward. Choose m large enough such that

$$A := mI + Q$$

is a nonnegative matrix. Then $-Q = mI - A$. Hence,

$$\lambda_0(Q) = m - \rho(A).$$

The proof now is a direct application of the Collatz–Wielandt formula:

$$\begin{aligned} m - \rho(A) &= m - \inf_{x>0} \max_i \frac{(Ax)_i}{x_i} = \sup_{x>0} \min_i \frac{(-Qx)_i}{x_i}, \\ m - \rho(A) &= m - \sup_{x>0} \min_i \frac{(Ax)_i}{x_i} = \inf_{x>0} \max_i \frac{(-Qx)_i}{x_i}. \end{aligned}$$ □

It is now ready to make some additional remarks on the results in the previous sections. The two algorithms as well as their convergence and the Collatz–Wielandt formula can be found easily from Wikipedia. From which, one knows that the Power Iteration was first appeared in 1929 [14] and the Inverse Iteration appeared in 1944 [15]. These algorithms are taught for undergraduate students on the course of computations and are included in many books, see for instance [10,13,16]. In particular, Appendix of Section 3 is modified from [10, pp. 584, 585].

We now say a few words about the unusual word "complete" used at the end of the first section for the results obtained in Section 3. Actually, this is one of the 16 situations with $N \leqslant \infty$ we have worked out so far to have a unified estimation of the principal eigenvalue:

$$(4\delta)^{-1} \leqslant \delta_1^{-1} \leqslant \lambda_0 \leqslant {\delta_1'}^{-1} \leqslant \delta^{-1} \tag{12}$$

for some constants δ, δ_1, and δ_1', where δ_1 is the one we have used in Section 3 for the initial z_0. Besides, we often have in practice that $1 \leqslant \delta_1/\delta_1' \leqslant 2$. Thus, the efficiency of the initial (v_0, z_0) introduced in Section 3 comes with no surprising. More precisely, the initial (v_0, z_0) is taken from the first step of our approximating procedure: [5, Theorem 3.3 (1), (3.4)]. Example 1 here is a truncated one from [5, Example 3.6] where $N = \infty$, $\lambda_0 = 1/4$, and $\delta_1 = 4$ which is sharp. Certainly, this is still not enough to claim that we can arrive at such a precise approximation in the second iteration. The story on the estimation of the principal eigenvalue, or more general on the estimation of the stability speed is too long to talk here and so the author is planning to publish a survey article [7]. For earlier progress, refer to [4] which includes a lot of information up to 2004, or a more recent paper [5].

Next, we discuss the sequence $(h_0, h_1, \ldots, h_N)$ used in Sections 3 and 4. The role of the sequence is to keep the same spectrum of the original Q and its H-transform $\widetilde{Q}$:

$$\widetilde{Q} = \mathrm{Diag}(h_i)^{-1} Q \mathrm{Diag}(h_i). \tag{13}$$

Certainly, Q and $\widetilde{Q}$ have the same diagonals. Next, define

$$P = (p_{ij} \colon i, j \in E) := \mathrm{Diag}(q_i^{-1})\, \widetilde{Q} + I, \tag{14}$$

which is the matrix used in Section 4. Note that even though the sequence (c_i) in the original Q can be non-zero, the resulting $\widetilde{c}_k = 0$ for every $k < N$ but $\widetilde{c}_N > 0$ for the matrix $\widetilde{Q}$. For a given measure μ, set $\widetilde{\mu} = h^2 \mu$ (i.e., $\widetilde{\mu}_i = h_i^2 \mu_i$ for each $i \in E$), the transform $\widetilde{f} = f/h$ gives us an isometry between $L^2(\mu)$ and $L^2(\widetilde{\mu})$ and then an isospectrum of Q on $L^2(\mu)$ and $\widetilde{Q}$ on $L^2(\widetilde{\mu})$. This technique is due to [8]. See also [6]. Now, if $\widetilde{g}$ is an approximating eigenvector corresponding to $\widetilde{\lambda}_0$ of $\widetilde{Q}$, then, $g := h\widetilde{g}$ is an approximating eigenvector corresponding to λ_0 of Q, due to the isospectral property of Q and $\widetilde{Q}$. Because

$$\|\widetilde{g}\|_{L^2(\widetilde{\mu})} = \|g\|_{L^2(\mu)}, \quad \big(\widetilde{g}, \widetilde{Q}\widetilde{g}\big)_{\widetilde{\mu}} = (g, Qg)_\mu,$$

by [8], we have

$$\frac{\big(\widetilde{g}, -\widetilde{Q}\widetilde{g}\big)_{\widetilde{\mu}}}{\|\widetilde{g}\|_{L^2(\widetilde{\mu})}} = \frac{(g, -Qg)_\mu}{\|g\|_{L^2(\mu)}} = \frac{g^*(-Q)g}{\sqrt{g^* g}}. \tag{15}$$

Here, we assume that $\mu_k \equiv 1$ for simplicity. This means that we can estimate the maximal eigenpair (λ_0, g) of Q in terms of the one $\big(\widetilde{\lambda}_0, \widetilde{g}\big)$ of $\widetilde{Q}$. More precisely, the maximal eigenvalue $\widetilde{g}$ of $\widetilde{Q}$ is approximated by φ in the context of Section 3 (or by $x = (x_i)$ in Section 4). Now, in Section 3 for instance, $\widetilde{v}_0 = h\sqrt{\varphi}$ is an approximation of the maximal eigenvector g of Q. With $v_0 = \widetilde{v}_0 / \sqrt{\widetilde{v}_0^* v_0}$, equation (15) leads to our first approximation of λ_0:

$$v_0^*(-Q)v_0 = z_0.$$

Now, our task is to show that the sequence (x_i) defined in Section 4 is an extension of (φ_i) given in Section 3. To this end, recall that the matrix $\widetilde{Q}$ defined by (13) is again a Q-matrix. Hence, the matrix $P = (p_{ij} \colon i, j \in E)$ defined by (14) is just the embedding chain of $\widetilde{Q}$. Note that here $p_{ii} = 0$ for each $i \in E$. By the construction of (h_i), we have $\sum_{j\in E} p_{ij} = 1$ for each $i \leqslant N-1$ but $\sum_{j\in E} p_{Nj} < 1$, refer to [8]. The equation for (x_i) in (9) can be rewritten as

$$x_n = \sum_{j\in E} p_{ij} x_j, \quad 1 \leqslant n \leqslant N, \quad x_0 = 1. \tag{16}$$

In probabilistic language, the solution (x_i) (or the minimal solution (x_i^*) when $N=\infty$) to equation (16) is the probability of first hitting 0 of the Q-process with Q-matrix $\widetilde{Q}$ or its embedding sub-Markov chain with transition matrix $P=(p_{ij})$, starting from i. Refer to [3, Lemma 4.46].

We are now going to prove the following result.

Lemma 24 *For birth–death matrix, the solution* (x_i) *to equation* (16) *coincides with* (φ_i) (*up to a constant*) *used in Section* 3.

Before prove Lemma 24, let us discuss the relation of these sequence with the recurrence of the Markov chain in the case of $N=\infty$. First, it is known by [3, Theorem 4.55 (1) and the second line of p. 161] that a birth–death process is recurrent if and only if

$$b_0 \sum_{n=1}^{\infty} \frac{a_1 a_2 \cdots a_n}{b_1 b_2 \cdots b_n} = b_0 \sum_{n=1}^{\infty} \frac{1}{\mu_n b_n} = \infty.$$

For simplicity, set

$$F_n^{(0)} = \frac{a_1 a_2 \cdots a_n}{b_1 b_2 \cdots b_n}, \quad n \geqslant 1.$$

The sequence $\{F_n^{(0)}\}_{n\geqslant 1}$ is a very special case of $\{\widetilde{F}_n^{(j)}\}_{n\geqslant 1}$ used in the proof of Proposition 9. Refer to [9] and [3, §4.5] for more details. Note that (φ_n) is just the tail series of $\sum_{n=1}^{\infty} F_n^{(0)}$ provided $N=\infty$. On the other hand, by [3, Lemma 4.46], the process is recurrent if and only if the minimal solution (x_i^*) to the equation (16),

$$x_n = \frac{b_n}{a_n+b_n}\, x_{n+1} + \frac{a_n}{a_n+b_n}\, x_{n-1}, \quad n \geqslant 1, \quad x_0 := 1,$$

is equal to one identically. Rewrite the equation as

$$x_n - x_{n+1} = \frac{a_n}{b_n}\,(x_{n-1} - x_n), \quad n \geqslant 1.$$

By induction, it follows that

$$x_n - x_{n+1} = F_n^{(0)}(x_0 - x_1), \quad n \geqslant 1.$$

Hence,

$$x_n - x_{N+1} = (x_0 - x_1)\sum_{k=n}^{N} F_k^{(0)}, \quad x_1 - x_n = (x_0 - x_1)\sum_{k=1}^{n-1} F_k^{(0)}, \quad n \geqslant 1.$$

Equivalently,

$$x_n - x_{N+1} = (x_0 - x_1)\sum_{k=n}^{N} F_k^{(0)}, \quad x_0 - x_n = (x_0 - x_1)\sum_{k=0}^{n-1} F_k^{(0)}, \quad n \geqslant 0,$$

since $F_0^{(0)} = 1$ by convention. If $\sum_{k=0}^{\infty} F_k^{(0)} = \infty$, then from the second equation, we must have $x_1 = 1$ (since $x_0 = 1$) and then have the unique solution $x_i \equiv 1$. Therefore, the minimal solution $x_i^* \equiv 1$ and so the process is recurrent. Conversely, if $\sum_{k=0}^{\infty} F_k^{(0)} < \infty$, then from the first equation above, we obtain

$$x_0 - x_1 = \frac{x_0 - x_\infty}{\sum_{j=0}^{\infty} F_j^{(0)}},$$

and then

$$x_n - x_\infty = \frac{x_0 - x_\infty}{\sum_{j=0}^{\infty} F_j^{(0)}} \sum_{k=n}^{\infty} F_k^{(0)}, \quad n \geqslant 0.$$

Equivalently,

$$x_n = \frac{\sum_{k=n}^{\infty} F_k^{(0)}}{\sum_{j=0}^{\infty} F_j^{(0)}} + x_\infty \frac{\sum_{k=0}^{n-1} F_k^{(0)}}{\sum_{j=0}^{\infty} F_j^{(0)}}, \quad n \geqslant 0.$$

Clearly, for each given $x_\infty \in [0,1]$, using this formula, we obtain a solution (x_n) to the equation. Thus, the minimal solution should be as follows:

$$x_n^* = \frac{\sum_{k=n}^{\infty} F_k^{(0)}}{\sum_{j=0}^{\infty} F_j^{(0)}}, \quad n \geqslant 0,$$

which is clearly less than one for $n \geqslant 1$ since $\sum_{j=0}^{\infty} F_j^{(0)} < \infty$.

Proof of Lemma 24 For finite state $\{0, 1, \ldots, N\}$, since there is a killing $b_N > 0$, the minimal solution is as follows:

$$x_n^* = \frac{\sum_{k=n}^{N} F_k^{(0)}}{\sum_{j=0}^{N} F_j^{(0)}}, \quad n = 0, 1, \ldots, N.$$

In other words, up to a constant, we have

$$\varphi_n = \sum_{k=n}^{N} F_k^{(0)} = \frac{1}{1 - x_1^*} x_n^*, \quad n = 0, 1, \ldots, N.$$

That is what we required. □

Finally, we remark that the story for one-dimensional diffusions should be in parallel to Section 3. The algorithm presented in Section 4 may not be complete since the lack of an analog of (12).

Summary

This paper deals with the efficient initials for the Rayleigh quotient iteration. Here are suggestions for the use of the results in the previous sections of the paper on computing the maximal eigenpair.

(i) If the iterations are easy (small size of A, for instance), one simply adopts the simplest algorithm: Section 4.1 with Choice I, or more effectively, with the convex combination of Choice I and Choice II:

$$z_0 = \xi \max_{i\in E} A_i + (1-\xi) v_0^* A v_0, \quad \xi \in [0,1].$$

More especially, $\xi = 7/8$ for instance. Certainly, one may use Choice III for z_0.

(ii) If the given matrix is nearly tridiagonal (after a suitable relabeling if necessary) or the Lanczos tridiagonalization procedure is suitable, one use the method introduced in Section 3. The computation there is rather explicit and it works even for $N = \infty$.

(iii) In general, one uses the algorithm given in Section 4.2. Note that at each step of the Rayleigh quotient iteration, one has to solve a linear equation. Here, for the initials, we have to solve two more linear equations.

6 Next to maximal eigenpair

After an earlier version of the paper containing the first five sections was submitted, the author found a natural way to study the next to the maximal eigenpair. In this section, we restrict ourselves to the easier case that $A_i := \sum_{j\in E} a_{ij}$ is a constant. Then the maximal eigenpair is simply $(A_0, \mathbb{1})$ (where $\mathbb{1}$ is the constant function having value 1 everywhere), as mentioned before. By a shift if necessary, we return to the problem for a Q-matrix which is especially valuable since its next eigenvalue describes the ergodic rate of the corresponding Markov chain. In this setup, the minimal eigenpair ($\lambda_0 = 0$, $g_0 = \mathbb{1}$) of $-Q$ is known and we are looking for the next eigenpair (λ_1, g_1). Clearly, g_1 should be orthogonal to g_0 in $L^2(\pi)$-sense for the stationary distribution π of the process corresponding to the given matrix Q. This is the reason why we often use $v - \pi v$ in what follows for constructing a mimic of the eigenvector g_1. Besides, we need the assumption that $\lambda_1 > |\lambda_j|$ for every $j > 1$ to guarantee the convergence of our algorithms.

Once again, let us begin our study with a tridiagonal conservative Q-matrix

$$Q = \begin{pmatrix} -b_0 & b_0 & 0 & 0 & \cdots \\ a_1 & -(a_1+b_1) & b_1 & 0 & \cdots \\ 0 & a_2 & -(a_2+b_2) & b_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & a_N & -a_N \end{pmatrix},$$

where $a_i, b_i > 0$. Define $(\mu_k \colon k \in E)$ as in Section 3. Then we have the probability distribution $\pi = (\pi_0, \pi_1, \ldots, \pi_N)$: $\pi_k = \mu_k / \sum_{j\in E} \mu_j$. Again, denote by $(\cdot,\cdot)_\mu$ and $\|\cdot\|_\mu$ the inner product and norm in $L^2(\mu)$, respectively. Next, set

$$\varphi_n = \sum_{j\leqslant n-1} \frac{1}{\mu_j b_j}, \quad n \in E.$$

To define our initial v_0, let

$$\widetilde{v}_0 = (\sqrt{\varphi_0},\, \sqrt{\varphi_1},\, \ldots,\, \sqrt{\varphi_N}\,)^*, \quad \overline{v}_0 = \widetilde{v}_0 - \pi\widetilde{v}_0.$$

We can now introduce our algorithm in the present situation as follows. Choose initials

$$v_0 = \frac{\overline{v}_0}{\|\overline{v}_0\|_\mu}, \quad z_0 = \frac{(\overline{v}_0, -Q\widetilde{v}_0)_\mu}{\|\overline{v}_0\|_\mu^2}. \tag{17}$$

At the kth step ($k \geqslant 1$), let w_k be the solution to the equation

$$(-Q - z_{k-1}) w_k = v_{k-1}$$

and set

$$v_k = \frac{w_k}{\|w_k\|_\mu}, \quad z_k = (v_k, -Qv_k)_\mu.$$

We remark that here in defining v_k ($k \geqslant 1$), we do not need to use $w_k - \pi w_k$. The reason is as follows. If $\pi v = 0$ and w solves the equation

$$(-Q - z)w = v$$

for some constant $z \neq 0$, then

$$0 = \pi v = \pi(-Q - z)w = -z\pi w,$$

and so $\pi w = 0$. Therefore, we have $\pi w_k = 0$ for each $k \geqslant 1$ since so does the initial v_0: $\pi v_0 = 0$.

Instead of z_0 given in (17), there is another choice. Define

$$\eta_1 = \max_{0\leqslant i\leqslant N-1} \frac{1}{\mu_i b_i [\widetilde{v}_0(i+1) - \widetilde{v}_0(i)]} \sum_{j=i+1}^{N} \mu_j \overline{v}_0(j).$$

Then one may choose

$$z_0 = \eta_1^{-1} \tag{18}$$

as an initial.

Here, the initials $\widetilde{v}_0$ and z_0 are taken from [2, Theorem 2.2 (1)] or [4, Theorem 1.5 (2)]. Certainly, we can adopt the convex combination of those given in (17) and (18):

$$z_0 = \xi\eta_1^{-1} + (1 - \xi)(\overline{v}_0, -Q\widetilde{v}_0)_\mu \|\overline{v}_0\|_\mu^{-2}, \quad \xi \in [0, 1]. \tag{19}$$

We now consider an example modified from Example 1.

Example 25 Let $E=\{0,1,\ldots,7\}$ and

$$Q=\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & -5 & 2^2 & 0 & 0 & 0 & 0 & 0\\ 0 & 2^2 & -13 & 3^2 & 0 & 0 & 0 & 0\\ 0 & 0 & 3^2 & -25 & 4^2 & 0 & 0 & 0\\ 0 & 0 & 0 & 4^2 & -41 & 5^2 & 0 & 0\\ 0 & 0 & 0 & 0 & 5^2 & -61 & 6^2 & 0\\ 0 & 0 & 0 & 0 & 0 & 6^2 & -85 & 7^2\\ 0 & 0 & 0 & 0 & 0 & 0 & 7^2 & -7^2 \end{pmatrix}.$$

Then we have $\mu_k\equiv 1$, $\lambda_1(Q)\approx 0.820539$ with eigenvector

$$\approx(-3.95053,-0.708966,0.246859,0.649164,0.842169,0.93805,0.983254,1)^*.$$

Starting from $\overline{v}_0$:

$$(-4.79299,\ -0.0815238,\ 0.474589,\ 0.70372,\ 0.828504,\ 0.906932,\ 0.960767,\ 1)^*,$$

for different initial z_0, the outputs are given in Table 13.

Table 13 Outputs for different initial z_0 (Example 25)

choice	z_0	z_1	$z_2=\lambda_1$
(17)	0.902633	0.820614	0.820539
(18)	0.456343	0.8216	0.820539
(19)	0.724117	0.820629	0.820539

We remark that for this and the next example, the parameter ξ in (19) is specified to be $2/5$.

The next example has non-trivial (μ_k).

Example 26 Let

$$Q=\begin{pmatrix} -5 & 5 & 0 & 0 & 0\\ 3 & -7 & 4 & 0 & 0\\ 0 & 2 & -3 & 1 & 0\\ 0 & 0 & 10 & -16 & 6\\ 0 & 0 & 0 & 11 & -11 \end{pmatrix}.$$

Then

$$\mu_0=1,\quad \mu_1=\frac{5}{3},\quad \mu_2=\frac{10}{3},\quad \mu_3=\frac{1}{3},\quad \mu_4=\frac{2}{11}.$$

The eigenvalues of $-Q$ are as follows:

$$22.348,\quad 10.6857,\quad 5.92951,\quad 3.03673,\quad 0.$$

With

$$\widetilde{v}_0=\frac{1}{2\sqrt{5}}\,(0,2,\sqrt{7},\sqrt{13},\sqrt{23}\,)$$

for different initial z_0, the outputs are given in Table 14.

Table 14 Outputs for different initial z_0 (Example 26)

choice	z_0	z_1	$z_2 = \lambda_1$
(17)	3.84977	3.05196	3.03673
(18)	1.72924	3.05715	3.03673
(19)	3.00156	3.03675	3.03673

Next, consider the general conservative Q-matrices $Q = (q_{ij}\colon i, j \in E)$. Here, the conservativity means that $\sum_{j \in E} q_{ij} = 0$ for every $i \in E$. Next, define an auxiliary Q-matrix Q_1 which coincides with Q except replacing the element q_{NN} by cq_{NN}, where $c > 1$ is an arbitrary constant and is fixed to be 1000 in what follows for simplicity.

Following Section 4 (replacing Q by Q_1), let $(x_0, x_1, \ldots, x_N)$ (with $x_0 = 1$) be the solution to the equation

$$x^{\backslash 0\text{'s row}} = P^{\backslash 0\text{'s row}}\, x, \tag{20}$$

where

$$P = \mathrm{Diag}(q_0^{-1}, q_1^{-1}, \ldots, q_{N-1,N-1}^{-1}, (cq_{NN})^{-1})Q_1 + I.$$

To go further, we need $\mu = (\mu_0, \mu_1, \ldots, \mu_N)$ with $\mu_0 = 1$, which is the same as defined in Section 4: the solution to the equation

$$Q^{*\,\backslash\text{the last row}} \mu^* = 0.$$

Having x and μ at hand, we are ready to define our initials. For each $r \in [0, 1]$, to be specified later, define

$$\begin{gathered} \widetilde{v}_0 = (r, \sqrt{1 - x_1}, \sqrt{1 - x_2}, \ldots, \sqrt{1 - x_N}\,)^*, \quad \overline{v}_0 = \widetilde{v}_0 - \frac{\mu \widetilde{v}_0}{\sum_{k=0}^N \mu_k}, \\ v_0 = \frac{\overline{v}_0}{\|\overline{v}_0\|_\mu}, \quad z_0 = \frac{(\overline{v}_0, -Q\widetilde{v}_0)_\mu}{\|\overline{v}_0\|_\mu^2}. \end{gathered} \tag{21}$$

Because $\widetilde{v}_0$ depends on r, so do $\overline{v}_0$, v_0, and $z_0 =: z_0(r)$. Choose $r_0 \in [0, 1]$ so that

$$z_0(r_0) \approx \inf_{r \in [0,1]} z_0(r).$$

Corresponding to this specified r_0, we obtain our initials v_0 and z_0. This minimizing procedure in r is necessary for avoiding collapse since we are in a more sensitive situation than before. Then the iteration procedure is exactly the same as we used several times before.

The reason we adopt a large $c = 1000$ here is that for a larger c, its minimal eigenvalue $\lambda_0(Q_1)$ is closer to, but less than, the eigenvalue $\lambda_1(Q)$ we are interested. Refer to [1, Proposition 3.2] for more details. Thus, one may regard the former as an approximation of the latter. In other words, we can use an alternative initial

$$z_0 = \lambda_0(Q_1) \text{ or its estimates studied in previous sections.} \tag{22}$$

Certainly, one can define a convex combination of those given in (21) and (22) in an obvious way, but it is omitted here. The use of $\lambda_0(Q_1)$ seems necessary (especially for large N) to avoid some pitfall, as mentioned before.

The next example is interesting for which some of its eigenvalues are complex but the one we are interested is real.

Example 27 Let

$$Q = \begin{pmatrix} -30 & 30 & 0 & 0 \\ 1/5 & -17 & 84/5 & 0 \\ 11/28 & 275/42 & -20 & 1097/84 \\ 55/3291 & 330/1097 & 588/1097 & -2809/3291 \end{pmatrix}.$$

Then

$$Q_1 = \begin{pmatrix} -30 & 30 & 0 & 0 \\ 1/5 & -17 & 84/5 & 0 \\ 11/28 & 275/42 & -20 & 1097/84 \\ 55/3291 & 330/1097 & 588/1097 & -2809000/3291 \end{pmatrix}.$$

The eigenvalues of $-Q$ and $-Q_1$ are

$$29.8411 + 2.45214\,\mathrm{i}, \quad 29.8411 - 2.45214\,\mathrm{i}, \quad 8.17131, \quad 0,$$

and

$$853.548, \quad 29.8249 + 2.46241\,\mathrm{i}, \quad 29.8249 - 2.46241\,\mathrm{i}, \quad 7.34195,$$

respectively. Using (21) with $r_0 \approx 0.951$, the output is

$$z_0 \approx 7.73667, \quad z_1 \approx 8.15021, \quad z_2 \approx 8.17129, \quad z_3 \approx 8.17131.$$

While using (22), the output is

$$z_0 \approx 7.34195, \quad z_1 \approx 8.13216, \quad z_2 \approx 8.17124, \quad z_3 \approx 8.17131.$$

Here is one more example.

Example 28 Let

$$Q = \begin{pmatrix} -57 & 118/27 & 91/9 & 1148/27 \\ 135/59 & -52 & 637/59 & 2296/59 \\ 243/91 & 590/91 & -47 & 492/13 \\ 351/287 & 118/41 & 195/41 & -62/7 \end{pmatrix}.$$

Then

$$Q_1 = \begin{pmatrix} -57 & 118/27 & 91/9 & 1148/27 \\ 135/59 & -52 & 637/59 & 2296/59 \\ 243/91 & 590/91 & -47 & 492/13 \\ 351/287 & 118/41 & 195/41 & -62000/7 \end{pmatrix}.$$

The eigenvalues of $-Q$ and $-Q_1$ are

$$59.3118, \quad 58, \quad 47.5454, \quad 0,$$

and

$$8857.18, \quad 59.2467, \quad 58, \quad 38.7143,$$

respectively. Using (21) with $r_0 \approx 0.953$, the output is

$$z_0 \approx 47.5318, \quad z_1 \approx 47.5453, \quad z_2 \approx 47.5454.$$

While using (22), the output is

$$z_0 \approx 38.7143, \quad z_1 \approx 47.5343, \quad z_2 \approx 47.5453, \quad z_3 \approx 47.5454.$$

Acknowledgements The main results of the paper have been reported at Anhui Normal University, Jiangsu Normal University, the International Workshop on SDEs and Numerical Methods at Shanghai Normal University, Workshop on Markov Processes and Their Applications at Hunan University of Arts and Science, and Workshop of Probability Theory with Applications at University of Macau. The author acknowledges Professors Dong-Jin Zhu, Wan-Ding Ding, Ying-Chao Xie, Xue-Rong Mao, Xiang-Qun Yang, Xu-Yan Xiang, Jie Xiong, Li-Hu Xu, and their teams for very warm hospitality and financial support. The author also thanks Ms. Yue-Shuang Li for her assistance in computing large matrices. This work was supported in part by the National Natural Science Foundation of China (Grant No. 11131003), the "985" project from the Ministry of Education in China, and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Chen M F. Explicit bounds of the first eigenvalue. Sci China Ser A, 2000, 43(10): 1051–1059
2. Chen M F. Variational formulas and approximation theorems for the first eigenvalue. Sci China Ser A, 2001, 44(4): 409–418
3. Chen M F. From Markov Chains to Non-equilibrium Particle Systems. 2nd ed. Singapore: World Scientific, 2004
4. Chen M F. Eigenvalues, Inequalities, and Ergodic Theory. London: Springer, 2005
5. Chen M F. Speed of stability for birth–death processes. Front Math China, 2010, 5(3): 379–515
6. Chen M F. Criteria for discrete spectrum of 1D operators. Commun Math Stat, 2014, 2: 279–309
7. Chen M F. Unified speed estimation of various stabilities. Chinese J Appl Probab Statist, 2016, 32(1): 1–22
8. Chen M F, Zhang X. Isospectral operators. Commun Math Stat, 2014, 2: 17–32
9. Chen M F, Zhang Y H. Unified representation of formulas for single birth processes. Front Math China, 2014, 9(4): 761–796
10. Golub G H, van Loan C F. Matrix Computations. 4th ed. Baltimore: Johns Hopkins Univ Press, 2013
11. Hua L K. Mathematical theory of global optimization on planned economy, (II) and (III). Kexue Tongbao, 1984, 13: 769–772 (in Chinese)

12. Langville A N, Meyer C D. Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton: Princeton Univ Press, 2006
13. Meyer C. Matrix Analysis and Applied Linear Algebra. Philadelphia: SIAM, 2000
14. von Mises R, Pollaczek-Geiringer H. Praktische Verfahren der Gleichungsaufösung. ZAMM Z Angew Math Mech, 1929, 9: 152–164
15. Wielandt H. Beiträge zur mathematischen Behandlung komplexer Eigenwertprobleme. Teil V: Bestimmung höherer Eigenwerte durch gebrochene Iteration. Bericht B 44/J/37, Aerodynamische Versuchsanstalt Göttingen, Germany, 1944
16. Wilkinson J H. The Algebraic Eigenvalue Problem. Oxford: Oxford Univ Press, 1965