INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 2, Number 1, Pages 27–41

THE FINEST LEVEL ACCELERATION OF MULTILEVEL AGGREGATION FOR MARKOV CHAINS

CHUN WEN, TING-ZHU HUANG, DE-AN WU, AND LIANG LI

Abstract. In this paper, we consider a class of new accelerated multilevel aggregation methods using two polynomial-type vector extrapolation methods, namely the reduced rank extrapolation (RRE) and the generalization of quadratic extrapolation (GQE) methods. We show how to combine the multilevel aggregation methods with the RRE and GQE algorithms on the finest level in order to speed up the numerical computation of the stationary probability vector for an irreducible Markov chain. Numerical experiments on typical Markov chain problems are reported to illustrate the efficiency of the accelerated multilevel aggregation methods.

Key words. Markov chains, multilevel aggregation, acceleration, vector extrapolation methods.

1. Introduction

The use of Markov chains is of interest in a wide range of applications, including information retrieval and web ranking [12, 28, 29, 32], queueing systems [16] and stochastic automata networks [14], as well as performance modeling of computer and communication systems, dependability and security analysis, and analysis of biological systems [40].

In this paper, we study a class of new accelerated multilevel aggregation methods which are efficient for computation of the stationary probability vector of an irreducible Markov chain. Mathematically, the problem to be solved is given by

$$Bx = x, \quad x_i \geq 0 \; \forall i, \quad ||x||_1 = 1, \tag{1}$$

where $B = (b_{ij}) \in \mathbb{R}^{n\times n}$ is a column stochastic matrix, i.e., $0 \leq b_{ij} \leq 1 \; \forall i, j$ and $e^{\mathrm{T}}B = e^{\mathrm{T}}$, with e the column vector with all elements as one, and $x \in \mathbb{R}^n$ is the stationary probability vector of Markov chains. In fact, if B is irreducible [5, 40], which is equivalent to the existence of a directed path from any vertex i to any other vertex j in the directed graph of B, then x is the unique solution to the linear system (1). Moreover, the stationary probability vector x of the Markov chains satisfies the inequality $x_i > 0 \; \forall i$.

For convenience, we rewrite the equation (1) as

$$Ax = 0, \tag{2}$$

with

$$A = I - B, \tag{3}$$

where I is an identity matrix, and A is a singular M-matrix with diagonal elements being the negative column sums of its off-diagonal elements. Hence, our interest is

Received by the editors June 21,2010 and, in revised form, November 30 ,2010 .
2000 *Mathematics Subject Classification.* 65C40, 60J22.
The authors would like to express their great thankfulness to the anonymous referees and Professor Yau Shu Wong for their constructive and helpful comments leading to the improvement of the paper. This research was supported by 973 Program (2007CB311002), NSFC (60973015), Sichuan Province Sci. & Tech. Research Project (2009SPT-1, 2009GZ0004).
This research was supported by NSFC (10926190, 60973015) and 973 Program (2007CB311002).

to solve the $n \times n$ homogeneous linear system (2) corresponding to an irreducible Markov chain.

Iterative procedures are commonly used numerical methods to compute the stationary probability vector for an irreducible Markov chain. Examples of iterative techniques include the power methods for calculating the dominant eigenvector of the matrix B [26–28, 33, 38], the Gauss-Seidel, SOR and SSOR iteration methods based on splitting of the matrix A [33, 34], the iterative aggregation/disaggregation algorithms for Markov chains [30, 31, 36, 41], the hybrid algorithm for queueing systems [48], and the well-known Krylov subspace methods such as the Arnoldi's algorithms [23, 45], BiCGSTAB and GMRES methods [33–35].

However, the iteration methods are likely to suffer from a slow convergence for some linear systems, for example, computing the principal eigenvector for Google matrix. Thus it is necessary to employ the idea of preconditioning. Philippe, Saad and Stewart considered three different incomplete factorizations: ILU0, ILUTH and ILUK as preconditioners for numerical solutions of Markov chain modelling in [33]. Virnik presented an algebraic multigrid preconditioner for M-matrices in [44]. Benzi and Uçar developed the block triangular and product preconditioners based on the alternating iteration [1] for M-matrices and the Markov chain problems in [2] and [3], respectively. In addition, the applications of circulant preconditioners for Markov chains had been report in [14, 15].

Recently, multilevel methods based on aggregation of the Markov states have been studied in the literature [17–20, 24, 25]. Isensee and Horton considered multi-level methods for the steady state solution of a continuous-time (CTMC) and discrete-time (DTMC) Markov chains in [25] and [24], respectively. De Sterck, Manteuffel, Mccormick, Nguyen and Ruge proposed a multilevel adaptive aggregation (MAA) method to calculate the stationary probability vector of Markov matrices in [18]. As already showed in [18], the multilevel method is a special case of the adaptive smoothed aggregation [8] and adaptive algebraic multigrid methods [7] for sparse linear systems.

Thereafter, De Sterck et al. proposed several strategies to accelerate the convergence of the multilevel aggregation methods. Such strategies include the application of a smoothing technique to the interpolation and restriction operators [17], and analyzing a recursive accelerated multilevel aggregation method by computing quadratic programming problems with inequality constraints[20]. In particular, a top-level acceleration of adaptive algebraic multilevel method was considered by finding a linear combination of previous fine-level iterates, so that it minimizes a functional over a subset of the probability vectors in [19]. The active set method from matlab's quadprog function [22] was used in their implementations.

Here, we consider a class of new accelerated multilevel aggregation methods by the use of two polynomial-type vector extrapolation methods: the reduced rank extrapolation (RRE) and the generalization of quadratic extrapolation (GQE) methods proposed by Sidi[38]. In fact, the idea to improve iteration methods by combining with vector extrapolation methods is not new; see [10, 11, 28, 39, 45]. This paper shows how to combine the multilevel aggregation methods with RRE and GQE algorithms on the finest level in order to speed up the numerical calculation of the stationary probability vector for an irreducible Markov chain. In numerical experiments, the accelerated multilevel aggregation methods are tested using three representative Markov chain problems. The problems include the nearly completely decomposable (NCD) Markov chains, which are difficult to solve since they consist of groups of states that are strongly interacted among each other but weakly interacted among the groups themselves. In particular, the necessary iteration counts increase rapidly with the size of the weak interactions approaching zero [24, 25].

The remainder of this paper is organized as follows. In Section 2, we briefly review the multilevel aggregation methods for the Markov chains [6, 42, 43]. Procedures to combine the multilevel aggregation methods with GQE and RRE algorithms on the finest level are presented in Section 3. Numerical simulations for three Markov chain problems are given in Section 4. Finally, conclusions are reported in Section 5.

2. Multilevel aggregation for Markov chains

In this section, we briefly review the multilevel aggregation methods to compute the stationary probability vector of Markov chains [6, 42, 43]. The main idea of a multilevel aggregation method is to transfer a large linear system into a smaller one by some aggregation strategies so that the numerical solution can be computed in an efficient way.

The basic multigrid algorithm for solving linear equations $Ax = b$ has been presented in [6, 42, 43]. Clearly, system (2) is a special case with $b = 0$. Using the same notations as in [6, 42, 43], let P_l be the full rank prolongation matrices of size $n_l \times n_{l+1}$, and R_l be the restriction operators of size $n_{l+1} \times n_l$, $l = 1, \cdots, L-1$. Here, the operators P_l and R_l are created by an automatic coarsening process, in which the coarse-level matrices are constructed by $A_{l+1} = R_l A_l P_l$, where $A_1 = A$. It is natural to number the levels so that the finest level is 1 and the coarsest level is L. Therefore, starting from the descriptions of the basic multigrid method for $Ax = b$, we have the following multilevel aggregation method for Markov chains, which is similar to that reported in [17–20].

Algorithm 1: Multilevel Aggregation Method, $x \leftarrow$ MA$(A, x, \nu_1, \nu_2, \alpha)$

```
(1) Pre-smoothing: Apply $\nu_1$ times $x \leftarrow N(\mathrm{Relax}(A, x))$.
(2) Construct $Q$ according to the automatic coarsening process described here.
    Obtain $R \leftarrow Q^{\mathrm{T}}$ and $P \leftarrow \mathrm{diag}(x)Q$.
(3) Form the coarse-level matrix $A_c \leftarrow RAP$, and compute $x_c \leftarrow Q^{\mathrm{T}}x$.
(4) If on the coarsest level, solve $A_c x_c = 0$ by a direct method. Otherwise apply
    $\alpha$ iterations of this algorithm $x_c \leftarrow \mathrm{MA}(A_c \mathrm{diag}(Q^{\mathrm{T}}x)^{-1}, x_c, \nu_1, \nu_2, \alpha)$.
(5) Coarse-level correction: $x \leftarrow P(\mathrm{diag}(Q^{\mathrm{T}}x)^{-1})x_c$.
(6) Post-smoothing: Apply $\nu_2$ times $x \leftarrow N(\mathrm{Relax}(A, x))$.
```

In Algorithm 1, $A = A_1$ is given in (2), x is an initial guess vector, A_c denotes the coarse-level matrix, and x_c is the corresponding coarse-level vector, Q is the aggregation matrix generated by the aggregation method, P and R are the prolongation and restriction operators, respectively. For the pre- and post-smoothing, the following weighted Jacobi method with weight ω is employed. Let the matrix A in (2) is split into the form

$$A = D - L - U,$$

where D is the diagonal part of A with $d_{ii} > 0$ $\forall i$, L, U are the negative strictly lower- and upper-triangular parts of A, respectively. Then the weighted Jacobi relaxation method can be written as

$$x \leftarrow N((1-\omega)x + \omega D^{-1}(L+U)x) \tag{4}$$

with the weight $\omega \in (0, 1)$. Here, we let $N(\cdot)$ denote the normalization operator defined by $N(x) := x/\|x\|_1$ $\forall x \neq 0$. Note that, a normalization is performed after each relaxation process to ensure that the fine-level iterates x_i can be interpreted as approximations to the stationary probability vector.

It is necessary to apply the direct solver for the coarsest level. The direct method used at step 4 of Algorithm 1 is based on the following theorem.

Theorem 1 (Theorem 4.16 in [4]). *If A is an irreducible singular M-matrix, then each of its principal submatrices other than A itself is a nonsingular M-matrix.*

If A_L is the coarsest-level operator, then we use the direct method presented in the coarsest-level algorithm below to solve the coarsest-level equation $A_L x_L = 0$.

Coarsest-level algorithm

Step 1. Compute $N_L := \text{size}(A_L, 1)$;

Step 2. Determine $A_{Lp} := A_L(1 : N_L - 1, 1 : N_L - 1)$;

Step 3. Determine $b_{Lp} := -A_L(1 : N_L - 1, N_L)$;

Step 4. Compute $x_{Lp} := A_{Lp} \setminus b_{Lp}$; let $x_{Lp}(N_L) = 1$; and set the coarsest-level solution $x_L = x_{Lp}/\|x_{Lp}\|_1$.

It is of vital importance to understand how the nodes are aggregated in step 2 of Algorithm 1, that is, we need to understand which nodes should be aggregated into a block and which nodes should be split between their neighbors. Here, we apply a neighborhood-based technique as our aggregation method since it is able to provide well-balanced aggregates of approximately equal size, and leads to a more regular coarsening throughout the automatic coarsening process; see [19, 20, 43].

Note that the neighborhood-based aggregation for Markov chains is based on the matrix scaled by the current iterate, i.e., $\overline{A} = A\text{diag}(x_k)$, rather than the original coefficient matrix A (for details see [18, 24, 25]), where diag$(\cdot)$ denotes a diagonal matrix formed with the current iterate x_k. Here, a node i is said to be strongly connected to node j in the graph of $\overline{A}$ if

$$-\overline{a}_{ij} \geq \theta \max_{k\neq i}\{-\overline{a}_{ik}\} \quad \text{or} \quad -\overline{a}_{ji} \geq \theta \max_{k\neq j}\{-\overline{a}_{jk}\}, \tag{5}$$

where θ is a strength of connection parameter. Suppose $\mathcal{N}_i$ is the set of all points which are strongly connected to i in the graph of $\overline{A}$ including node i itself. Then the neighborhood-based algorithm is given as follows [19, 20].

Algorithm 2: Neighborhood-Based Aggregation, $\{Q_j\}_{j=1}^J \leftarrow$ **NBA**$(\overline{A}, \theta)$

```
(1) Set $R = \{1, \cdots, n\}$ and $J = 0$.
(2) Assign entire neighborhoods to aggregates:
    for $i \in \{1, \cdots, n\}$, construct strong neighborhoods $\mathcal{N}_i$ based on (5),
    if $\mathcal{N}_i \subset R$ then $J \leftarrow J + 1$, $Q_J \leftarrow \mathcal{N}_i$, $\widetilde{Q}_J \leftarrow \mathcal{N}_i$, $R \leftarrow R \setminus \mathcal{N}_i$.
(3) Determine the remaining points in the most connected aggregates:
    while $R \neq \emptyset$, pick $i \in R$ and set $j = \underset{k=1,\cdots,J}{\text{argmax}}\, \text{card}(\mathcal{N}_i \cap \widetilde{Q}_k)$,
    set $Q_j \leftarrow Q_j \cup \{i\}$ and $R \leftarrow R \setminus \{i\}$.
(4) Construct the aggregation matrix $Q$:
    if $i \in Q_j,\ j = 1, \cdots, J$, then $Q_{ij} = 1$, otherwise $Q_{ij} = 0$.
```

Note that the aggregation matrix Q in Algorithm 2 has the following form

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{n \times J}, \tag{6}$$

where J denotes the number of aggregates. Clearly, from (6), the matrix Q has the properties that there exists only one element $Q_{ij} = 1$ in each row, but each column may have several elements $Q_{ij} = 1$, and the sum of the elements in the jth column denotes the number of the nodes combined into the jth aggregate.

3. Accelerated multilevel aggregation for Markov chains

In this section, we first give a short introduction to two polynomial-type vector extrapolation methods: the reduced rank extrapolation (RRE) and the generalization of quadratic extrapolation (GQE) methods. The key idea of our work is then presented. Finally, we show how to combine the multilevel aggregation methods with RRE and GQE algorithms.

Various kinds of vector extrapolation methods have been discussed in [39], and this includes the polynomial-type vector extrapolation methods such as the minimal polynomial extrapolation (MPE) of Cabay and Jackson [13] and RRE method of Eddy [21], and the epsilon vector extrapolation methods utilizing the scalar and vector epsilon methods of Wynn [46, 47], and the topological epsilon method of Brezinski [9].

It should be noted that the starting point of the vector extrapolation algorithms is to accelerate the convergence of the sequences $\{x_j\}$ generated from a fixed-point iterative method of the form

$$x_{j+1} = F(x_j), \quad j = 0, 1, \cdots; \quad F : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \tag{7}$$

where x_0 is an initial guess. In recent years, applications of the vector extrapolation methods to compute the stationary probability vector of Markov chains have been reported in [10, 11, 28, 38, 45]. Numerical simulations have also illustrated that the polynomial-type methods are in general more economic than the epsilon vector extrapolation methods with respect to the computing time and storage requirements. Therefore, two polynomial-type vector extrapolation methods, the RRE and GQE methods proposed in [38], are considered in this paper.

Here we give the definition of the RRE method.

Definition 1 (see [38]). *Let $\{x_j\}$ be a given sequence in $\mathbb{R}^n$, and $u_j = x_{j+1} - x_j$, choose $k \le n$ arbitrarily and define a matrix U via $U = [u_0, u_1, \cdots, u_k] \in \mathbb{R}^{n\times(k+1)}$. Let z be the least squares solution to the linear system $Uz = 0$ subject to the constraint $\sum_{j=0}^{k} z_j = 1$; this means that z is the solution to the problem*

$$\min_{z_0, z_1, \ldots, z_k} \| \sum_{j=0}^{k} z_j u_j \|_2, \quad \textit{subject to} \sum_{j=0}^{k} z_j = 1. \tag{8}$$

Then the RRE approximation for the exact solution of system (2) is given by

$$\widehat{x}_k = \sum_{j=0}^{k} z_j x_j. \tag{9}$$

Clearly, the approximate solution $\widehat{x}_k = Xz$ has the property

$$\|\widehat{x}_k\|_1 = \sum_{i=0}^{n} (\widehat{x}_k)_i = \sum_{i=1}^{n} \sum_{j=0}^{k} X_{ij} z_j = \sum_{j=0}^{k} z_j \sum_{i=1}^{n} X_{ij} = \sum_{j=0}^{k} z_j = 1$$

once it is checked to satisfy the condition that $\widehat{x}_k \geq 0$. This implies that the updated iterate is also a probability vector of Markov chains. Hence, starting from this point, we find that both the acceleration process considered in this paper and that reported in [1] are based on a linear combination of previous fine-level iterates, i.e., $\widehat{x}_k = Xz$, which is similar in principle to a preconditioned flexible Krylov wrapper applied to an additive iteration for a linear problem. The main difference between these accelerated methods lies in the computation of the vector z.

Note that the definition of RRE here is the most direct approach for our computational purposes. It is different from the original definition in [21]. We give the algorithm of RRE as follows [38].

Algorithm 3: The Reduced Rank Extrapolation Method (RRE)

```
(1) Input vectors $x_0, x_1, \cdots, x_{k+1}$.
(2) Compute $u_i = x_{i+1} - x_i, i = 0, 1, \cdots, k$, and set $U_k = [u_0, u_1, \cdots, u_k]$.
    Compute the QR-factorization of $U_k$, namely, $U_k = Q_k R_k$.
    Determine $R_{k-1} := R_k(1:k, 1:k)$, $Q_{k-1} := Q_k(:, 1:k)$.
(3) Solve the linear system
    $R_k^{\mathrm{T}} R_k d = e$, $d = [d_0, d_1, \cdots, d_k]^{\mathrm{T}}$, $e = [1, 1, \cdots, 1]^{\mathrm{T}} \in R^{k+1}$.
    Set $\lambda = \sum_{i=0}^{k} d_i$, compute $\gamma = [\gamma_0, \gamma_1, \cdots, \gamma_k]^{\mathrm{T}}$ by $\gamma_i = d_i/\lambda,\ i = 0, 1, \cdots, k$.
(4) Compute $\xi = [\xi_0, \xi_1, \cdots, \xi_{k-1}]^{\mathrm{T}}$ by
    $\xi_0 = 1 - \gamma_0$, $\xi_j = \xi_{j-1} - \gamma_j, j = 1, \cdots, k-1$.
    Comput $\widehat{x}_{k+1} = x_0 + Q_{k-1}(R_{k-1}\xi)$.
```

Next, we investigate another vector extrapolation method - the quadratic extrapolation method presented by Kamvar, Haveliwala, Manning and Golub for accelerating the computations of the dominant eigenvector for the PageRank; see [28]. Wu and Wei discussed the close connection with the Arnoldi's method [23] based on Ritz values in [45]. Moreover, Sidi reported that it was closely related to the MPE of Cabay and Jackson [13] and proposed a generalization of the quadratic extrapolation method along with the implementation of MPE; see [38]. Thus, for completeness, the algorithm of GQE is provided as follows [38].

Algorithm 4: The Generalization of Quadratic Extrapolation (GQE)

```
(1) Input the vectors $x_0, x_1, \cdots, x_{k+1}$.
(2) Compute $u_i = x_{i+1} - x_0, i = 0, 1, \cdots, k$, set $U_k = [u_0, u_1, \cdots, u_k]$.
    Compute the QR-factorization of $U_k$, namely, $U_k = Q_k R_k$.
    Determined $R_{k-1} := R_k(1:k, 1:k)$, $Q_{k-1} := Q_k(:, 1:k)$.
(3) Solve the linear system $R_{k-1} d = -Q_{k-1}^{\mathrm{T}} u_k$, $d = [d_0, d_1, \cdots, d_{k-1}]^{\mathrm{T}}$.
(4) Set $d_k = 1$ and compute $c = [c_0, c_1, \cdots, c_k]^{\mathrm{T}}$ by $c_i = \sum_{j=i}^{k} d_j, i = 0, 1, \cdots, k$.
```

(5) Compute $\widehat{x}_{k+1} = (\sum\limits_{i=0}^{k} c_i)x_0 + Q_k(R_k c)$.

Sidi has given the numerically fast, stable and efficient storage MPE and RRE algorithms in [37]. A common feature of the Algorithms 3 and 4 is that they both contain a QR-factorization in step 2 for $U_k = Q_k R_k$, where $Q_k \in \mathbb{R}^{n\times(k+1)}$ is unitary, and $R_k \in \mathbb{R}^{(k+1)\times(k+1)}$ is an upper triangular matrix with positive diagonal elements. Precisely, we have

$$Q_k = (q_0, q_1, \cdots, q_k) \in \mathbb{R}^{n\times(k+1)}, \quad Q_k^{\mathrm{T}} Q_k = I_{(k+1)\times(k+1)}.$$

$$R_k = \begin{pmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0k} \\ & r_{11} & r_{12} & \cdots & r_{1k} \\ & & r_{22} & \cdots & r_{2k} \\ & & & \ddots & \vdots \\ & & & & r_{kk} \end{pmatrix}, \quad r_{ii} > 0 \ \forall i.$$

To develop an efficient implementation for the QR-factorization, we apply the modified Gram-Schmidt process (MGS) to vectors $u_0, u_1, \cdots, u_k$, see [37, 38]. The MGS algorithm is given as follows.

MGS algorithm

Step 1. Compute $r_{00} = (u_0, u_0)^{1/2}$, and set $q_0 = u_0/r_{00}$;

Step 2. For $j = 1 : k$, set $u_j^{(0)} = u_j$;

Step 3. For $i = 1 : j$, compute $r_{ij} = (q_i, u_j^{(i)})$ and $u_j^{(i+1)} = u_j^{(i)} - r_{ij} q_i$;

Step 4. Compute $r_{jj} = (u_j^{(j)}, u_j^{(j)})^{1/2}$ and $q_j = u_j^{(j)}/r_{jj}$.

From Algorithms 3 and 4, we observe that the updated iterate $\widehat{x}_{k+1}$ is given only in terms of the previous iterates $x_j,\ j = 0, 1, \cdots, k+1$, and no other input is required. This is similar to the analytic properties of RRE and GQE methods [28, 38]. Hence, RRE and GQE methods can be used as effective accelerators to improve the convergence of the vector sequence $\{x_j\}$. In particular, given the vectors $x_0, x_1, \cdots, x_{k+1}$ and the QR-factorization in the MGS algorithm, the operation counts of Algorithms 3 and 4 consist of $\frac{1}{2}(k^2 + 5k + 2)$ vector additions, $\frac{1}{2}(k^2 + 5k)$ scalar-vector multiplications and $\frac{1}{2}(k^2 + 3k + 2)$ inner products [38].

The present work is motivated from the study of the excellent papers in [19, 20, 38]. In particular, De Sterck, Miller, Manteuffel and Sanders have improved the convergence of the multilevel aggregation methods by selecting a linear combination of previous iterates to minimize a functional within the space of probability vector (for details see [19]). Similar to the theoretical analysis in [19], we suppose there is a sequence of iterates $\{x_i\}_{i=1}^{\infty}$ obtained by Algorithm 1, where $x_i \geq 0$. Then at the kth outer iteration, let

$$X = [x_k, x_{k-1}, \cdots, x_{k-m+1}] \in \mathbb{R}^{n\times m}$$

be a matrix consisting of the last m iterates with x_k being the newest, where m is called the window size. As mentioned above, all the probability vectors x_i are nonnegative, thus the matrix X has the following properties:

$$x_i \geq 0 \text{ and } \|x_i\|_1 = 1,\ i = 1, 2, \cdots.$$

Combining with Definition 1, the problem to be solved has been transformed into computing the vector z satisfing $\sum_{i=1}^{m} z_i = 1$. Thus we obtain an updated probability vector $\widehat{x}_k = Xz = z_1 x_k + z_2 x_{k-1} + \cdots + z_m x_{k-m+1}$, which is a linear combination of the last m iterates. It has been reported in [19] that computing the vector z is equivalent to solving a small quadratic programming problem (QPP) for any window size $m \geq 2$, in which the active set method from matlab's quadprog function [22] has been used in the implementations.

Therefore, it is natural to consider a new accelerated multilevel aggregation method for the window size $m \geq 2$. Our main contribution is to show how to combine the multilevel aggregation method with RRE and GQE algorithms on the finest level to accelerate the numerical computation of the stationary probability vector for the Markov chains.

Algorithm 5: Accelerated Multilevel Aggregation Methods by RRE and GQE, $x \leftarrow \mathbf{AMA}(A, \widehat{x}_0, m, \epsilon)$

```
(1) Set $k = 1$, if no initial guess is given, choose $\widehat{x}_0$.
(2) Implement the multilevel aggregation method, $x_k \leftarrow \mathrm{MA}(A, \widehat{x}_{k-1}, \nu_1, \nu_2, \alpha)$.
(3) Set $m \leftarrow \min\{M, k\}$.
(4) Set $X \leftarrow [x_k, x_{k-1}, \cdots, x_{k-m-1}]$.
(5) Apply RRE and GQE algorithms to obtain $\widehat{x}_k$ from $X$, respectively.
(6) If $\|A\widehat{x}_k\|_1 > \|Ax_k\|_1$, then $\widehat{x}_k \leftarrow x_k$.
(7) Check convergence, if $\|A\widehat{x}_k\|_1/\|A\widehat{x}_0\|_1 < \epsilon$, then $\widehat{x}_k \leftarrow \widehat{x}_k/\|\widehat{x}_k\|_1$, otherwise
    set $k \leftarrow k + 1$ and go to step 2.
```

Note that, in Algorithm 5 m is the window size and ϵ is a prescribed tolerance. In particular, there exists a difference in constructing the matrix X between our Algorithm 5 and the algorithm of [1]. From Algorithms 3 and 4, the RRE and GQE methods require $k + 2$ vectors $x_0, x_1, \cdots, x_{k+1}$ as inputs. That is to say, when the window size is $m = 2$, four approximate probability vectors given in the matrix $X = [x_0, x_1, x_2, x_3] \in \mathbb{R}^{n\times 4}$ are needed as input in the GQE and RRE algorithms. The window size $m = 2$ corresponds to the quadratic vector extrapolation method presented in [28]. Hence, the matrix X is given as the form of the step 4 in Algorithm 5. The efficiency of the accelerated multilevel aggregation algorithms will be demonstrated shortly.

4. Numerical simulations

In this section, we report numerical results obtained using a Matlab 7.0.1 implementation on Window-XP with 2.93GHz 64-bit processor and 2GB memory. The main goal is to examine the new accelerated multilevel aggregation algorithms and to show the speed up for numerical solutions of stationary probability vector for the Markov chains. Here we set the window size as $m = 2, 3, 4$. Three typical Markov chain problems studied in [17, 24, 25] have been used in our experiments. We compare the efficiency of the unaccelerated V-cycles ($\alpha = 1$ in Algorithm 1) and the unaccelerated W-cycles ($\alpha = 2$ in Algorithm 1). For convenience, we let W_GQE and W_RRE denote the applications of GQE and RRE methods with W-cycles.

Without loss of generality, special sets of the parameters are employed and they are taken from [17, 19, 20]. As mentioned early, the weighted Jacobi method is used as the pre- and post-smoothing in Algorithm 1. Let $\nu_1 = \nu_2 = 1$ and set the relaxation parameter $\omega = 0.7$ in the experiments. Note that, even though the values work well for all tests considered here, they are likely to be problem-dependent. The coarsest-level solution is implemented by the coarsest-level algorithm and the strength of connection parameter is chosen as $\theta = 0.25$. The initial guess in Algorithm 5 is generated by random sampling with a uniform (0,1) distribution and normalized to one in the one norm. All iterations are terminated when $\mathrm{res} = \|Ax_k\|_1/\|Ax_0\|_1 \leq 10^{-5}$ with x_k the current approximate solution.

Numerical results for various accelerated multilevel aggregation methods are now reported, where "n" is the problem size, "lev" is the number of levels in the multilevel aggregation methods, "it" denotes the iteration counts less than 100, "CPU" denotes the computing time, and "C_{op}" is the operator complexity of the last cycle, which is defined as the sum of the nonzero entries in all operators on all levels, divided by the nonzero entries in the fine-level operator, i.e., $C_{op} = \frac{\sum_{l=1}^{L} \mathrm{nnz}(A_l)}{\mathrm{nnz}(A_1)}$ with $\mathrm{nnz}(A)$ is the number of nonzero entries in A. In addition, if the convergence factor for the kth iteration is defined by $\gamma_k := \|Ax_k\|_1/\|Ax_{k-1}\|_1$, then the effective convergence factor can be defined by $\gamma_{eff} := \gamma^{1/C_{op}}$, where γ is the geometric mean of the sequence $\{\gamma_k\}$.

The first problem is a one-dimensional (1D) Markov chain with identical transition rates, which often results from a $M/M/1$ queueing system with identical arrival and service rates. The simplest graph of this test problem is illustrated in Fig. 1. In fact, the iteration counts needed to solve this problem increases sharply when the problem size is increasing, since in principle the probability has to be distributed evenly throughout the chain and a number of iterations are required to propagate [24, 25]. Numerical results are presented in Tables 1, 2 and 3.

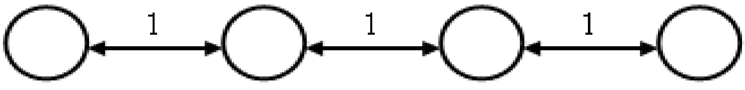

Fig. 1: Graph for one-dimensional Markov chain with identical transition rates.

		V-cycles				W-cycles			
n	lev	C_{op}	γ_{eff}	it	res	C_{op}	γ_{eff}	it	res
243	2	1.5007	0.7138	20	9.6963e-6	2.0014	0.7766	20	9.6963e-6
729	3	1.7506	0.9177	64	9.9026e-6	3.0018	0.9231	41	9.0712e-6
2187	4	1.8754	0.9570	>100	2.8835e-5	4.0017	0.9602	58	9.5623e-6
4374	5	1.9378	0.9641	>100	9.1344e-5	5.0015	0.9789	87	9.7252e-6

Table1: Unaccelerated V-cycles and W-cycles for one-dimensional Markov chain.

From Table 1, we observe that the unaccelerated W-cycles is more effective than the unaccelerated V-cycles, and it requires less iterations for the one-dimensional Markov chain. In addition, the convergence factor of the W-cycles is comparable to that of the V-cycles, but the operator complexity of the W-cycles is larger than that of the V-cycles.

Table 2 presents the numerical results of the accelerated W-cycles with GQE and RRE methods for the one-dimensional Markov chain with the window size m: $m = 2, 3, 4$. Clearly, both the effective convergence factor and the iteration counts of the accelerated W-cycles are less than those based on the unaccelerated W-cycles. However, the operator complexities are identical. In particular, window size $m = 3$ and $m = 4$ provide the most effective acceleration for the GQE and RRE methods. The accelerated W-cycles with GQE method is superior to that with RRE method. For instance, when the problem size is 2187 or 4374, comparing with the results using the unaccelerated W-cycles shown in Table 1, the iterations has been reduced by 63% for W_GQE, and 57% for W_RRE. Hence, we conclude that the accelerated

Method			Window size					
W_GQE			2		3		4	
n	lev	C_{op}	γ_{eff}	it	γ_{eff}	it	γ_{eff}	it
243	2	2.0014	0.7014	12	0.6716	11	0.6818	12
729	3	3.0018	0.8342	18	0.8553	21	0.8523	19
2187	4	4.0017	0.9162	28	0.9041	24	0.8947	22
4374	5	5.0015	0.9573	43	0.9606	47	0.9450	33
W_RRE			2		3		4	
n	lev	C_{op}	γ_{eff}	it	γ_{eff}	it	γ_{eff}	it
243	2	2.0014	0.6733	13	0.6943	13	0.7014	14
729	3	3.0018	0.8569	22	0.8449	20	0.8592	22
2187	4	4.0017	0.9239	30	0.9070	25	0.9014	23
4374	5	5.0015	0.9569	42	0.9517	37	0.9519	37

Table 2: Accelerated W-cycles with GQE and RRE methods for one-dimensional Markov chain.

		W-cycles	W_GQE			W_RRE		
n	lev	CPU	2	3	4	2	3	4
27	2	0.2500	0.2030	0.1870	0.2030	0.2030	0.2030	0.2030
81	2	0.5470	0.3600	0.3590	0.3280	0.3750	0.3590	0.4540
243	2	1.3280	0.8440	0.7810	0.8440	0.9060	0.9060	0.9690
486	3	14.890	6.5470	6.9070	6.1720	7.9530	6.9070	7.6410
729	3	33.016	14.953	17.359	15.766	18.156	16.562	18.109
1458	4	293.30	142.49	137.47	113.53	166.49	132.22	112.63

Table 3: CPU of unaccelerated W-cycles and accelerated W-cycles with GQE and RRE methods for one-dimensional Markov chain.

multilevel aggregation methods with GQE and RRE produce faster convergence than that of the unaccelerated methods.

In Table 3, we report the CPU time using the unaccelerated W-cycles and accelerated W-cycles with GQE and RRE methods. It is clear that compared to the unaccelerated W-cycles, significant saving in computing time ranging from 20%–62% is achieved using the accelerated W-cycles with GQE and RRE methods.

For the second test problem, we consider a birth-death chain with invariant birth and death rates as shown in Fig. 2. Generally speaking, this kind of birth-death chain is used in the study of demographics, queueing theory, performance engineering or in biology [5, 40]. This test problem has also been discussed in [17, 24, 25]. Here, we choose $\mu = 0.96$ in the simulations. Numerical results for this problem are given in Tables 4, 5 and 6.

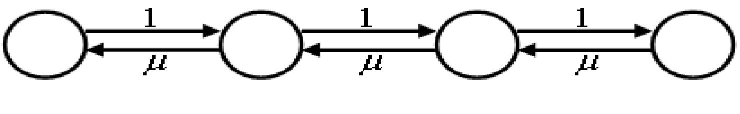

Fig. 2: Graph for a birth-death chain with $\mu = 0.96$.

		V-cycles				W-cycles			
n	lev	C_{op}	γ_{eff}	it	res	C_{op}	γ_{eff}	it	res
27	2	1.5063	0.7163	21	6.8180e-6	2.0127	0.7791	21	6.8180e-6
81	3	1.7552	0.9259	69	9.9695e-6	3.0166	0.9288	43	8.9192e-6
243	4	1.8831	0.9648	>100	1.3287e-4	4.0481	0.9695	75	9.4890e-6
1024	5	1.9355	0.9704	>100	3.0908e-4	7.0430	1.5757	92	8.9849e-6
2048	7	1.9827	0.9694	>100	2.0776e-4	17.625	1.1949	>100	0.5676

Table 4: Unaccelerated V-cycles and W-cycles for the birth-death chain with $\mu = 0.96$.

Method		Window size								
W_GQE		2			3			4		
n	lev	C_{op}	γ_{eff}	it	C_{op}	γ_{eff}	it	C_{op}	γ_{eff}	it
27	2	2.0127	0.6678	12	2.0127	0.6691	11	2.0127	0.6601	9
81	3	3.0166	0.8308	17	3.0166	0.8419	18	3.0166	0.8376	16
243	4	4.0481	0.9149	25	4.0261	0.8967	20	4.0481	0.8925	19
1024	5	4.9831	0.9611	45	7.0430	1.5757	92	4.9935	0.9607	40
2048	7	7.0443	0.9724	46	7.6734	0.9811	56	16.483	0.9974	93
W_RRE		2			3			4		
n	lev	C_{op}	γ_{eff}	it	C_{op}	γ_{eff}	it	C_{op}	γ_{eff}	it
27	2	2.0127	0.6619	12	2.0127	0.6854	12	2.0127	0.6521	11
81	3	3.0166	0.8501	20	3.0166	0.8299	17	3.0166	0.8398	18
243	4	4.0481	0.9241	31	4.0481	0.9294	28	4.0481	0.9067	24
1024	5	4.9831	0.9636	49	5.1160	0.9804	74	4.9831	0.9612	40
2048	7	6.9922	0.9815	62	6.9922	0.9814	62	7.1172	0.9849	63

Table 5: Accelerated W-cycles with GQE and RRE methods for the birth-death chain.

The numerical simulations presented in Tables 4 - 6 for the birth-death chain problem clearly demonstrate the effectiveness and improvement using the accelerated W-cycles method compared with the unaccelerated version. It is observed that the operator complexity increases when the problem size is increasing. For a fixed and large problem size, the window size m could affect the complexity. For example, when $n = 2048$, the operator complexity for the accelerated W-cycles with GQE method is given by 7.04, 7.67 and 16.48 for $m = 2, 3, 4$ respectively. The computing time listed in Table 6 also reveals that more time are required for the accelerated W-cycles with GQE and RRE methods when $m = 3$ compared to those needed with $m = 2$ and 4. For this test problem, the accelerated W-cycles with GQE and RRE using $m = 4$ produce the most efficient results, and significant reduction in CPU time is achieved compared to those needed using the unaccelerated W-cycles.

The final test problem is displayed in Fig. 3, which is a uniform chain with two weak links in the middle. The problem consists of groups of nodes that are strongly connected among each other but have two weak links in the middle of this chains. This is a typical nearly completely decomposable (NCD) Markov chain problem, and it has been discussed in [17, 24, 25]. Let $\tau = 1e-3$ in the numerical simulation, and the results are given in Tables 7, 8 and 9.

From Table 7, we note that the number of iterations of the unaccelerated W-cycles is less than that of the unaccelerated V-cycles. In addition, the effective

		W-cycles	W_GQE			W_RRE		
n	lev	CPU	2	3	4	2	3	4
27	2	0.2500	0.2030	0.2030	0.1870	0.2030	0.2030	0.2030
81	3	1.9070	0.8440	0.8910	0.8120	0.9690	0.8440	0.8900
243	4	13.906	4.8590	4.0310	3.8280	5.9850	5.3910	4.7500
486	4	37.953	17.500	22.344	13.562	16.937	22.844	14.656
729	4	66.640	36.687	55.343	34.766	34.969	56.250	29.891
1024	5	272.97	119.06	272.55	106.27	129.33	192.92	105.92

Table 6: CPU of unaccelerated W-cycles and accelerated W-cycles with GQE and RRE for the birth-death chain.

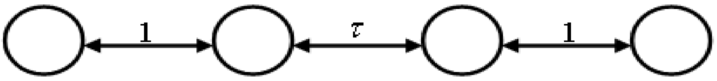

Fig. 3: Graph for a uniform chain with two weak links in middle($\tau = 1e - 3$).

		V-cycles				W-cycles			
n	lev	C_{op}	γ_{eff}	it	res	C_{op}	γ_{eff}	it	res
64	2	1.4947	0.7337	22	8.0129e-6	1.9895	0.7924	22	8.0129e-6
256	3	1.7467	0.9139	61	9.6932e-6	2.9896	0.9213	40	8.8346e-6
1024	4	1.8736	0.9575	>100	3.3384e-5	3.9928	0.9624	62	9.4829e-6
4096	5	1.9370	0.9639	>100	8.8132e-5	4.9958	0.9784	85	9.8224e-6
6144	6	1.9683	0.9652	>100	1.0263e-4	5.9938	0.9852	>100	1.2875e-6

Table 7: Unaccelerated V-cycles and W-cycles for the uniform chain with two weak links in middle ($\tau = 1e - 3$).

convergence factor of the unaccelerated W-cycles is comparable to that of the unaccelerated V-cycles, but the operator complexity of the unaccelerated W-cycles increases linearly as the number of levels in the multilevel aggregation methods is increasing.

Table 8 gives the numerical results of the accelerated W-cycles with GQE and RRE methods for the uniform chain with two weak links in the middle. The effective convergence factor and the iteration counts of the accelerated W-cycles are less than those of the unaccelerated W-cycles, but their operator complexities are identical. The most effective acceleration for the accelerated W-cycles with GQE and RRE methods corresponds when the window size $m = 4$.

Based on the CPU reported in Table 9, we observe that the computing time of the unaccelerated W-cycles has been reduced tremendously by about 27%–61% when the accelerated W-cycles with GQE and RRE methods are employed.

5. Conclusions

In this paper, we present a new approach for the finest level acceleration of multilevel aggregation methods for the solution of the stationary probability vector of an irreducible Markov chain. Our work is inspired by the excellent papers reported in[19, 20, 38]. Using a linear combination of the previous fine-level iterates, we construct a class of new accelerated methods to improve the convergence of

Method			Window size					
W_GQE			2		3		4	
n	lev	C_{op}	γ_{eff}	it	γ_{eff}	it	γ_{eff}	it
64	2	1.9895	0.6863	12	0.6636	11	0.6638	10
256	3	2.9896	0.8227	17	0.8324	18	0.8175	16
1024	4	3.9928	0.9165	28	0.9137	24	0.9071	24
4096	5	4.9958	0.9586	43	0.9530	39	0.9568	41
6144	6	5.9938	0.9818	79	0.9764	63	0.9715	53
W_RRE			2		3		4	
n	lev	C_{op}	γ_{eff}	it	γ_{eff}	it	γ_{eff}	it
64	2	1.9895	0.7137	14	0.6794	12	0.7044	13
256	3	2.9896	0.8442	20	0.8368	19	0.8317	18
1024	4	3.9928	0.9322	33	0.9164	27	0.9348	31
4096	5	4.9958	0.9588	44	0.9587	43	0.9503	36
6144	6	5.9938	0.9732	55	0.9722	52	0.9690	47

Table 8: Accelerated W-cycles with GQE and RRE methods for the uniform chain with two weak links in middle ($\tau = 1e - 3$).

		W-cycles	W_GQE			W_RRE		
n	lev	CPU	2	3	4	2	3	4
64	2	0.5310	0.3600	0.3440	0.3280	0.3910	0.3590	0.3750
128	2	0.8910	0.5470	0.5460	0.5160	0.5780	0.6090	0.5470
256	3	6.5930	2.9680	3.1090	2.7810	3.4380	3.2650	3.1100
512	3	17.375	7.8130	8.2350	7.7970	9.9850	8.2500	8.2500
1024	4	146.25	68.313	58.688	58.812	79.281	65.500	76.110
2048	5	1.21e+3	653.52	599.98	475.50	649.00	580.64	554.45

Table 9: CPU of unaccelerated W-cycles and accelerated W-cycles with GQE and RRE methods for the uniform chain with two weak links in middle ($\tau = 1e - 3$).

the unaccelerated V-cycles and W-cycles methods. Here, we consider two vector extrapolation methods: the reduced rank extrapolation (RRE) [38] and the generalized quadratic extrapolation (GQE) methods [28, 38]. We discuss how to efficiently combine the multilevel aggregation methods with the RRE and GQE algorithms on the finest level. To validate and to demonstrate the effectiveness of the developed accelerated multilevel aggregation methods, numerical simulations are carried out for three typical Markov chain problems. The computational results confirm that the accelerated W-cycles method is very efficient, and it provides a significant saving in computing time compared to the corresponding unaccelerated method.

References

[1] Benzi, M., Szyld, D.B., Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods, Numer. Math., 76 (1997) 309-321.

[2] Benzi, M., Uçar, B., Block triangular preconditioners for M-matrices and Markov chains, Electronic Trans. Numerical Analysis, 26 (2007) 209-227.

[3] Benzi, M., Uçar, B., Product preconditioning for Markov chain problems, in Proceedings of the 2006 Markov Anniversary Meeting, A. N. Langville and W. J. Stewart, eds., Raleigh, NC, 2006, (Charleston, SC), Boson Books, pp. 239-256.

[4] Berman, A., Plemmons, R.J., Nonnegative Matrices in the Mathematics Science, SIAM, Philadelpha, 1987

[5] Bini, D.A., Latouche, G., Meini, B., Numerical Methods for Structured Markov Chains, Oxford University Press, London, 2005.

[6] Brandt, A., McCormick, S., and Ruge. J., Algebraic multigrid (AMG) or sparse matrix equations. DJ Evans (Ed.), Sparsity and its Applications, 1984.

[7] Brezina, M., Falgout, R., Maclachlan, S., Manteuffel, T., Mccormick, S., and Ruge, J., Adaptive algebraic multigrid, SIAM J. Sci. Comput., 27 (2006) 1261-1286.

[8] Brezina, M., Falgout, R., Maclachlan, S., Manteuffel, T., Mccormick, S., and Ruge, J., Adaptive smoothed aggregation (αSA) multigrid, SIAM Rev., 47 (2005) 317-346.

[9] Brezinski, C., Généralisations de la transformation de Shanks, de la table de Padé, et de 1'ϵ-algorithme, Calcolo, 11 (1975) 317-360.

[10] Brezinski, C., Redivo Zaglia, M., Rational extrapolation for the PageRank vector, Math. Comp., 77 (2008) 1585-1598.

[11] Brezinski, C., Redivo Zaglia, M., The PageRank vector: properties, computation, approximation, and acceleration, SIAM J. Matrix Anal. Appl., 28 (2006) 551-575.

[12] Broder, A.Z., Lempel, R., Maghoul, F., Pedersen, J., Efficient PageRank approximation via graph aggregation, Inf Retrieval, 9 (2006) 123-138.

[13] Cabay, S., Jackson, L.W., A polynomial extrapolation method for finding limits and antilimits of vector sequences, SIAM J. Numer. Anal., 13 (1976) 734-752.

[14] Chan, R., Ching, W.K., Circulant preconditioners for stochastic automata networks, Numer. Math., 87 (2000) 35-57.

[15] Ching, W.K., Circulant preconditioners for failure prone manufacturing systems, Lin. Alg. Appl., 266 (1997) 161-180.

[16] Ching, W.K., Iterative methods for queueing systems with batch arrivals and negative customers, BIT, 43 (2003) 285-296.

[17] De Sterck, H., Manteuffel, T.A., McCormick, S.F., Miller, K., Pearson, J., Ruge, J., and Sanders, G., Smoothed aggregation multigrid for Markov chains, SIAM J. Sci. Comput., 32 (2010) 40-61.

[18] De Sterck, H., Manteuffel, T.A., Mccormick, S.F., Nguyen, Q., and Ruge, J., Multilevel adaptive aggregation for Markov chains, with applications to web ranking, SIAM J. Sci. Comput., 30 (2008) 2235-2262.

[19] De Sterck, H., Miller, K., Manteuffel, T., and Sanders, G., Top-level acceleration of adaptive algebraic multilevel methods for steady-state solution to Markov chains, Advances in Computational Mathematics, accepted, 2010.

[20] De Sterck, H., Miller, K., Sanders, G., and Winlaw, M., Recursively accelerated multilevel aggregation for Markov chains, SIAM J. Sci. Comput., 32 (2010) 1652-1671.

[21] Eddy, P.R., Extrapolating to the limit of a vector sequence, in: P. C. C. Wang (Ed.), Information Linkage Between Applied Mathematics and Industry, Academic Press, New York, 1979, pp. 387-396.

[22] Gill, P.E., Murray, W., and Wright, M.H., Practical optimization, Academic Press, London, UK, 1981.

[23] Golub, G.H., Greif, C., An Arnoldi-Type algorithm for computing PageRank, BIT, 46 (2006) 759-771.

[24] Isensee, C., Horton, G., A multi-level method for the steady state solution of discrete-time Markov chains, In Proceedings of the 2nd Balkan conference in informatics, pp. 413-420, Ohrid, Macedonia, November 2005.

[25] Isensee, C., Horton, G., A multi-level method for the steady state solution of Markov chains, In: Simulation and Visualization 2004, Magdeburg, Germany.

[26] Kamvar, S.D., Haveliwala, T.H., Golub, G.H., Adaptive methods for the computation of PageRank, Stanford University Technical Report, 2003.

[27] Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H., Exploiting the block structure of the web for computing PageRank, Stanford University Technical Report, 1999.

[28] Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H., Extrapolation methods for accelerating PageRank computations, In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[29] Langville, A.N., Meyer, C.D., A survey of eigenvector methods for web information retrieval, SIAM Review, 47 (2005) 135-161.

[30] Marek, I., Mayer, P., Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vector of stochastic matrices, Numerical Lin. Alg. Appl., 5 (1998) 253-274.

[31] Marek, I., Mayer, P., Convergence theory of some classes of iterative aggregation/disaggregation methods for computing stationary probability vector of stochastic matrices, Lin. Alg. Appl., 363 (2003) 177-200.

[32] Page, L., Brin, S., Motwani, R., and Winograd, T., The PageRank citation ranking: Bringing order to the web, Technical Report 1999-0120, Computer Science Department, Stanford, 1999.

[33] Philippe, B., Saad Y., and Stewart, W.J., Numerical methods in Markov Chain modeling, Operations Research, 40 (1992) 1156-1179.

[34] Saad, Y., Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, Philadelphia, PA, 2003.

[35] Saad, Y., Preconditioned Krylov subspace methods for the numerical solution of Markov chains, In W. J. Stewart, editor, Computations with Markov chains, pp. 49-64, Kluwer academic publishers, 1995.

[36] Schweitzer, P.J., Kindle, K.W., An iterative aggregation-disaggregation algorithm for solving linear equations, Appl. Math. Comput., 18 (1986) 313-354.

[37] Sidi, A., Efficient implementation of minimal polynomial and reduced rank extrapolation methods, J. Comput. Appl. Math., 36 (1991) 305-337. (Originally appeared as NASA TM-103240 ICOMP-90-20).

[38] Sidi, A., Vector extrapolation methods with applications to solution of large systems of equations and to PageRank computations, Computers and Math. Appl., 56 (2008) 1-24.

[39] Smith, D.A., Ford, W.F., Sidi, A., Extrapolation methods for vector sequence, SIAM Review, 29 (1987) 199-233.

[40] Stewart, W.J., An Introduction to the Numerical Solution of Markov Chains, Princeton University Press, Princeton, 1994.

[41] Stewart, W.J., Cao, W.L., Iterative aggregation/disaggregation techniques for nearly uncoupled Markov chains, J. ACM, 32 (1985) 702-719.

[42] Vaněk, P., Mandel, J., and Brezina, M., Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing, 56 (1996) 179-196.

[43] Vaněk, P., Mandel, J., and Brezina, M., Algebraic multigrid on unstructured meshes. Technical, Report X, Center for Computational Mathematics, Mathematics Department, 1994.

[44] Virnik, E., An algebraic multigrid preconditioner for a class of singular M-matrices, SIAM J. Sci. Comput., 29 (2007) 1982-1991.

[45] Wu, G., Wei, Y., An Arnoldi-Extrapolation algorithm for computing PageRank, J. Comput. Appl. Math., 2010.

[46] Wynn, P., Accelerated techniques for iterated vector and matrix problems, Math. Comp., 16 (1962) 301-322.

[47] Wynn, P., On a device for computing $e_m(S_n)$ transformation, Math. Tab. Other Aids Comput., 10 (1956) 91-96.

[48] Yuen, W. O., Ching, W.K., Ng, M.K., A hybrid algorithm for queueing systems, Calcolo, 41 (2004) 139-151.

School of Mathematical Sciences/Institute of Computational Science, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P. R. China
E-mail: wchun17@163.com (C. Wen), tingzhuhuang@126.com (T.-Z. Huang)