AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 6(1): 893–907.
DOI: 10.3934/math.2021053
Received: 01 September 2020
Accepted: 29 October 2020
Published: 04 November 2020

Research article

Acceleration of an adaptive generalized Arnoldi method for computing PageRank

Chun Wen[1,*], Qian-Ying Hu[2], Bing-Yuan Pu[3] and Yu-Yun Huang[4]

[1] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 610054, China
[2] School of Mathematical Sciences, Guizhou Normal University, Guiyang, 550025, China
[3] Department of Basic Courses, Chengdu Textile College, Chengdu, 611731, China
[4] School of Economic Mathematics, Southwestern University of Finance and Economics, Chengdu, 611130, China

* **Correspondence:** Email: wchun17@163.com.

Abstract: By considering a weighted inner product, an adaptive generalized Arnoldi (GArnoldi) method was constructed by [13] for computing PageRank. In order to accelerate the adaptive GArnoldi method, this paper proposes a new method by using the power method with extrapolation process based on Google matrix's trace (PET) as an accelerated technique of the adaptive GArnoldi method. The new method is called as GArnoldi-PET method, whose implementation and convergence analysis are discussed in detail. Numerical experiments are used to illustrate the effectiveness of our proposed method.

Keywords: PageRank; generalized Arnoldi method; extrapolation; power method
Mathematics Subject Classification: 65F15, 65F10

1. Introduction

Using the hyperlink structure of web pages, Google's PageRank becomes one of the most successful methods for measuring the importance of each page [1]. From the viewpoint of numerical computations, the core of PageRank problems can be regarded as the problem of solving a dominant eigenvector of the Google matrix A:

$$Ax = x, \quad A = \alpha P + (1-\alpha)ve^{\mathrm{T}}, \quad \|x\|_1 = 1, \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the PageRank vector, $\alpha \in (0,1)$ is a damping factor, $e = [1, 1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^n$, $v = e/n$, $P \in \mathbb{R}^{n\times n}$ is a column-stochastic matrix, see [2] for details.

As an iterative method based on matrix-vector products, the power method is widely used for computing PageRank [1, 3]. However, when the damping factor α is close to 1, the power method suffers from slow convergence such that some accelerated techniques are developed. For example, based on the inner-outer iteration method proposed by Gleich et al. [4], Tian et al. [5] developed a general inner-outer iteration method for solving PageRank problems. Using the trace of the Google matrix A, Tan [6] introduced an extrapolation strategy and presented the power method with extrapolation process based on trace (PET) for improving the computation of PageRank problems.

On the other hand, Krylov subspace methods based on the Arnoldi process have been applied to compute PageRank problems. Golub and Greif [7] proposed an Arnoldi-type method by using the singular value decomposition (SVD), where the known largest eigenvalue 1 is considered as a shift such that the computation of the largest Ritz value is avoided. Wu and Wei [8] developed a Power-Arnoldi algorithm by periodically combining the power method with the thick restarted Arnoldi algorithm [9]. Hu et al. [10] proposed a variant of the Power-Arnoldi algorithm by employing the PET method.

Recently, the idea of introducing weighted inner products into an Arnoldi process has successfully been applied to many academic fields [11,12]. Yin et al. [13] proposed an adaptive generalized Arnoldi (GArnoldi) method for computing PageRank by applying a weighted inner product into an Arnoldi-type method. Wen et al. [14] developed an adaptive Power-GArnoldi algorithm by making use of the power method and the adaptive GArnoldi method together. Motivated by these works, with the aim of accelerating the adaptive GArnoldi method, a new method is proposed by periodically knitting the PET method with the adaptive GArnoldi method for PageRank problems. The new method is denoted as GArnoldi-PET method. Convergence performance of our proposed method is studied in detail, and numerical results are used to show its feasibility and effectiveness.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the PET method and the adaptive GArnoldi method for PageRank problems. In Section 3, we propose the GArnoldi-PET method and discuss its convergence. In Section 4, numerical results and comparisons are reported. Finally, conclusions are given in Section 5.

2. Previous work

In this section, we give simple introductions of the PET method and the adaptive GArnoldi method for computing PageRank.

2.1. The PET method for computing PageRank

Here, we first give the algorithmic version of the PET method for PageRank problems as follows, see [6] for more details.

Algorithm 1. The PET method

Input: an initial guess $x^{(0)}$, a prescribed tolerance *tol*, a positive integer m_1, $r = 1$ and $k = 0$.
Output: PageRank vector x.
1. Compute the number of dangling nodes l and $\mu = 1 + \alpha\left(\frac{l}{n} - 1\right)$.
2. Run the power iteration m_1 steps to obtain $x^{(m_1-1)}$ and $x^{(m_1)}$.
2.1. for $i = 1 : m_1$

2.2. $x^{(i)} = Ax^{(i-1)}$;
2.3. $r = \|x^{(i)} - x^{(i-1)}\|_2$;
2.4. $x^{(i)} = x^{(i)}/\|x^{(i)}\|_1$;
2.5. if $r \leq tol$, break; endif
2.6. end
3. Use the extrapolation scheme based on $x^{(m_1-1)}$, $x^{(m_1)}$ and μ:
3.1. $x^{(0)} = x^{(m_1)} - (\mu - 1)x^{(m_1-1)}$;
3.2. $x^{(0)} = x^{(0)}/\|x^{(0)}\|_1$;
3.3. $r = \|x^{(0)} - x^{(m_1)}\|_2$;
3.4. if $r \leq tol$, break; else, goto step 2; endif

Now, some illustrations of Algorithm 1 are given as follows.

- In step 1, the parameter μ is the trace of the Google matrix A.
- In step 2, the power method is run m_1 steps, which means the extrapolation technique is not employed to the power method in each iteration, but is used every m_1 power iterations.
- In step 3, we can see that the extrapolation strategy based on $x^{(m_1-1)}, x^{(m_1)}$ and μ is easy to implement as given in step 3.1.

2.2. The adaptive GArnoldi method for computing PageRank

As described in [13], let $G = (g_{ij})$ be an $n \times n$ symmetric positive definite (SPD) matrix, then the GArnoldi process based on a weighted inner product is presented as Algorithm 2.

Algorithm 2. The GArnoldi process

Input: an initial vector v_1, and the steps m of GArnoldi process, a SPD matrix G.
Output: V_m, H_m.
1. Compute $\widetilde{v}_1 = v_1/\|v_1\|_G$.
2. for $j = 1, 2, \cdots, m$
3. $q = A\widetilde{v}_j$;
4. for $i = 1, 2, \cdots, j$
5. $h_{i,j} = (q, \widetilde{v}_i)_G$, $q = q - h_{i,j}\widetilde{v}_i$;
6. end
7. $h_{j+1,j} = \|q\|_G$;
8. if $h_{j+1,j} = 0$, break; endif
9. $\widetilde{v}_{j+1} = q/h_{j+1,j}$;
10. end

In Algorithm 2, $(\cdot,\cdot)_G$ is a G-inner product defined as $(x, y)_G = x^{\mathrm{T}}Gy, \forall x \in \mathbb{R}^n, y \in \mathbb{R}^n$, and $\|\cdot\|_G$ is a G-norm defined by

$$\|x\|_G = \sqrt{(x,x)_G} = \sqrt{x^{\mathrm{T}}Gx} = \sqrt{x^{\mathrm{T}}Q^{\mathrm{T}}DQx} = \sqrt{\sum_{i=1}^{n} d_i(Qx)_i^2}, \quad \forall x \in \mathbb{R}^n, \tag{2.1}$$

where $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, $D = \text{diag}\{d_1, d_2, \cdots, d_n\}$ is a $n \times n$ diagonal matrix with $d_i > 0$, $i = 1, 2, \cdots, n$, and $G = Q^{\mathrm{T}}DQ$ is a diagonalized decomposition of G. Let $e_m \in \mathbb{R}^m$ be the m-th co-ordinate vector, then the GArnoldi process has the following relations [13]

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^{\mathrm{T}} = V_{m+1} H_{m+1,m},\ V_m^{\mathrm{T}} G A V_m = H_m,\ H_{m+1,m} = \begin{pmatrix} H_m \\ h_{m+1,m} e_m^{\mathrm{T}} \end{pmatrix},$$

where the matrix $V_k = [\widetilde{v}_1, \widetilde{v}_2, \cdots, \widetilde{v}_k]$ $(k = m, m+1)$ is an $n \times k$ G-orthogonal matrix, $H_m = (h_{ij})$ is an $m \times m$ Hessenberg matrix.

From Algorithm 2, it is obvious that different SPD matrices G will lead to different GArnoldi methods. Since every SPD matrix can be diagonalized, for simplicity, we let $G = \text{diag}\{d_1, d_2, \cdots, d_n\}$, $d_i > 0$, $i = 1, 2, \cdots, n$. It is seen that in each outer iteration of the GArnoldi method, we hope to find a vector v satisfying $\min\|Av - v\|_G$, where v is taken from a Krylov subspace $\mathcal{K}_m(A, v_1) = \text{span}(v_1, Av_1, \cdots, A^{m-1}v_1)$. Denote $r = Av - v \triangleq [r_1, r_2, \cdots, r_n]^{\mathrm{T}}$, it has $\min\|Av - v\|_G = \min\sqrt{\sum_{i=1}^n d_i r_i^2}$, which leads to a weighted least squares problem where d_i is actually the weight for the i-th component of residual r_i, $i = 1, 2, \cdots, n$. In order to speed up the computation of PageRank problems, Yin et al. [13] changed the weights adaptively according to the changing of the current residual corresponding to the approximate PageRank vector. Therefore, one choice of the matrix G is that

$$G = \text{diag}\{d_1, d_2, \cdots, d_n\},\ d_i = |r_i|/\|r\|_1,\ i = 1, 2, \cdots, n,$$

where r is the residual vector computed by the last calculation, and $\sum_{i=1}^n d_i = 1$. And the algorithmic version of the adaptive GArnoldi algorithm for computing PageRank is presented as Algorithm 3.

Algorithm 3. The adaptive GArnoldi method

Input: an initial vector $x^{(0)}$, the steps m of the GArnoldi process, a prescribed tolerance tol.
Output: PageRank vector x.
1. Set $G = I$.
2. for $i = 1, 2, \cdots$, until convergence,
3. Run Algorithm 2 for computing V_m, V_{m+1} and $H_{m+1,m}$.
4. Compute singular value decomposition $U\Sigma S^T = H_{m+1,m} - [I; 0]^{\mathrm{T}}$.
5. Compute $x = V_m s_m$, $r = \sigma_m V_{m+1} u_m$.
6. if $\|r\|_2 \le tol$, break; endif
7. Set $G = \text{diag}\{|r|/\|r\|_1\}$.
8. end

Note that, in step 5 of Algorithm 3, s_m and u_m denote the right and left singular vector of $H_{m+1,m} - [I; 0]^{\mathrm{T}}$ associated with the minimal singular value σ_m, respectively.

3. The GArnoldi-PET method for computing PageRank

In order to accelerate the computation of PageRank problems, we develop a new method by combining the PET method with the adaptive GArnoldi method. The new method is called GArnoldi-PET method. Here we first describe the construction of the GArnoldi-PET method, and then discuss its convergence.

3.1. The GArnoldi-PET method

As described in the subsection 2.1, based on the trace of the Google matrix A, an extrapolation strategy has been presented to speed up the convergence of the power method. Numerical experiments in [6] have illustrated that the PET method has a faster convergence than the power method when the damping factor α is close to 1. On the other hand, since the Arnoldi method is more computationally intense than applying the same number of iterations of the power method [8], thus it is natural to consider using the extrapolation strategy based on trace and the adaptive GArnoldi method together.

Similar to the construction of the Power-Arnoldi algorithm [8], the mechanism of our proposed method can be presented as follows: Given a unit positive vector $x^{(0)}$, and an approximate PageRank vector is obtained by iterating the adaptive GArnoldi method (Algorithm 3) for a few times (e.g., 2–3 times). If this approximate PageRank vector does not satisfy our prescribed tolerance, then we run the PET method to obtain another approximate vector with the resulting vector as the initial guess. If this approximate PageRank vector still can not satisfy our accuracy, then we return to Algorithm 3 with the new approximation as the starting vector. Repeating the above procedure until the described accuracy is achieved.

There is a problem about how to control the conversion between the PET method and the adaptive GArnoldi method. Many strategies have been developed to deal with this problem. Here, as given in [8], three parameters β, $restart$, $maxit$ are used to control the procedure. Let $\tau^{(curr)}$ be the residual norm of the current iteration, and $\tau^{(prev)}$ be the residual norm of the previous iteration. Computing $ratio = \tau^{(curr)}/\tau^{(prev)}$, if $ratio > \beta$, then $restart = restart + 1$. If $restart \geq maxit$, then we terminate the PET method and trigger the adaptive GArnoldi method. The specific implementation of the GArnoldi-PET method is given as follows.

Algorithm 4. The GArnoldi-PET method

Input: an initial guess $x^{(0)}$, the dimension of the Krylov subspace m, a prescribed tolerance tol, the parameters β, $maxit$ and m_1. Set $k = 1$, $restart = 0$, $\tau = 1$, $\tau_0 = \tau$, $\tau_1 = \tau$.
Output: PageRank vector x.
1. Compute the number of dangling nodes l and $\mu = 1 + \alpha\left(\frac{l}{n} - 1\right)$.
2. Run Algorithm 3 for a few times (2–3 times): iterate all steps of Algorithm 3 for the first run and steps 2–8 otherwise. If the approximation is satisfactory, then stop, else continue.
3. Run the modified PET method with the resulting vector $\tilde{x}_1$ as the initial guess, where $\tilde{x}_1$ is obtained from the adaptive GArnoldi method:
3.1. $restart = 0$;
3.2. while $restart < maxit$ & $\tau > tol$
3.3. $ratio = 0$; $\tau_0 = \tau$; $\tau_1 = \tau$;
3.4. while $ratio < \beta$ & $\tau > tol$
3.5. $x^{(k)} = Ax^{(k-1)}$; $x^{(k)} = x^{(k)}/\|x^{(k)}\|_1$;
3.6. $r = x^{(k)} - x^{(k-1)}$; $\tau = \|r\|_2$;
3.7. if $mod(k, m_1) = 0$
3.8. $x^{(0)} = x^{(k)} - (\mu - 1)x^{(k-1)}$; $x^{(0)} = x^{(0)}/\|x^{(0)}\|_1$;
3.9. $r = x^{(0)} - x^{(k)}$; $\tau = \|r\|_2$; $x^{(k)} = x^{(0)}$;
3.10. if $\tau \leq tol$, break; endif

3.11. end
3.12. $ratio = \tau/\tau_0$; $\tau_0 = \tau$; $k = k + 1$;
3.13. end
3.14. if $\tau/\tau_1 > \beta$, $restart = restart + 1$; endif
3.15. end
3.16. if $\tau \leq tol$, stop, else set $G = \text{diag}\left\{\frac{|r|}{\|r\|_1}\right\}$ and goto step 2.

Now, some remarks about the GArnoldi-PET method are given as follows.

- As shown in the step 3.16, the matrix G is adaptively changed according to the current residual.
- According to the construction of the GArnoldi-PET method, it is natural to treat the PET method as an accelerated technique for the adaptive GArnoldi method.
- In each iteration of the GArnoldi-PET method, the storage requirements are approximately $m + 1$ length$-n$ vectors in the adaptive GArnoldi method and two vectors in the PET method. Its main computational cost consists of m matrix-vector products, $\frac{m(m+1)}{2}$ inner products in the adaptive GArnoldi method and one matrix-vector product in the PET method.

3.2. Convergence analysis of the GArnoldi-PET method

Here we discuss the convergence analysis of the GArnoldi-PET algorithm. Particularly, we focus on the procedure when turning from the PET method to the adaptive GArnodli method.

Assume $\sigma(A)$ denote the set of eigenvalues of the Google matrix A, and its eigenvalues are arranged as $1 = |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Let $\mathcal{L}_{m-1}$ be the set of polynomials of degree not exceeding $m - 1$, and $\mathcal{K}_m(A, v_1)$ be a Krylov subspace. If $(\lambda_i, \varphi_i), i = 1, 2, \cdots, n$ are the eigenpairs of A, and $(\widetilde{\lambda}_j, \widetilde{y}_j), j = 1, 2, \cdots, m$ are the eigenpairs of H_m, then $\widetilde{\lambda}_j$ is often used to approximate λ_j, and $\widetilde{\varphi}_j = V_m\widetilde{y}_j$ is applied to approximate φ_j in the standard Arnoldi method. However, instead of using Ritz vectors $\widetilde{\varphi}_j$ as approximate eigenvectors, Jia [15] proposed a new strategy such that for each $\widetilde{\lambda}_j$, a unit norm vector $\widetilde{u}_j \in \mathcal{K}_m(A, v_1)$ satisfying the condition

$$\left\|(A - \widetilde{\lambda}_j I)\widetilde{u}_j\right\|_2 = \min_{u \in \mathcal{K}_m(A,v_1)} \left\|(A - \widetilde{\lambda}_j I)u\right\|_2 \tag{3.1}$$

is used to approximate φ_j. Here $\widetilde{u}_j$ is called a refined approximate eigenvector corresponding to λ_j. The convergence of the refined Arnoldi method is given as follows.

Theorem 1 [15]. Assume that $v_1 = \sum_{i=1}^{n} \gamma_i x_i$ with respect to the eigenbasis $\{x_i\}_{i=1,2,\cdots,n}$ in which $\|x_i\|_2 = 1, i = 1, 2, \cdots, n$ and $\gamma_i \neq 0$, let $S = [x_1, x_2, \cdots, x_n]$, and

$$\xi_j = \sum_{i \neq j} \left|\lambda_i - \widetilde{\lambda}_j\right| \cdot \frac{|\gamma_i|}{|\gamma_j|}.$$

Then

$$\left\|(A - \widetilde{\lambda}_j I)\widetilde{u}_j\right\|_2 \leq \frac{\sigma_{max}(S)}{\sigma_{min}(S)} \left(\left|\lambda_j - \widetilde{\lambda}_j\right| + \xi_j \min_{p \in \mathcal{L}_{m-1}, p(\lambda_j)=1} \max_{i \neq j} |p(\lambda_i)|\right),$$

where $\widetilde{u}_j$ is a refined approximate eigenvector as above, $\sigma_{max}(S)$ and $\sigma_{min}(S)$ are the largest and smallest singular value of the matrix S, respectively.

Before analyzing the convergence of the GArnoldi-PET algorithm, some useful conclusions are introduced as follows.

Lemma 1 [14]. Let $G = \mathrm{diag}\{d_1, d_2, \cdots, d_n\}$, $d_i > 0, 1 \le i \le n$, be a diagonal matrix. Then for any vector $x \in \mathbb{R}^n$, according to the definitions of the G-norm and the 2-norm, it has

$$\min_{1\le i\le n} d_i \cdot \|x\|_2^2 \le \|x\|_G^2 \le \max_{1\le i\le n} d_i \cdot \|x\|_2^2 . \tag{3.2}$$

Lemma 2 [10]. Let v_1 be the initial vector for the PET method, which is from the previous adaptive GArnoldi method. Then the PET iteration in Algorithm 4 produces the vector

$$v_1^{new} = \eta T^k v_1, T = A^{m_1-1}[A - (\mu - 1)I], \tag{3.3}$$

where $k \ge maxit$, η is the normalizing factor, μ is the trace of the matrix A, m_1 is a given number, T is called as the iterative matrix and I is an $n \times n$ identity matrix.

Theorem 2 [16]. Assume that the spectrum of the column-stochastic matrix P is $\{1, \lambda_2, \cdots, \lambda_n\}$, then the spectrum of the matrix $A = \alpha P + (1 - \alpha)ve^{\mathrm{T}}$ is $\{1, \alpha\lambda_2, \cdots, \alpha\lambda_n\}$, where $0 < \alpha < 1$, v is a vector with nonnegative elements such that $e^{\mathrm{T}}v = 1$.

Theorem 3 [17]. Let P be an $n \times n$ column-stochastic matrix. Let α be a real number such that $0 < \alpha < 1$. E is the $n \times n$ rank-one column-stochastic matrix $E = ve^{\mathrm{T}}$, where e is the n-vector of all ones and v is an n-vector whose elements are all non-negative and sum to 1, $A = \alpha P + (1 - \alpha)ve^{\mathrm{T}}$ is the $n \times n$ column-stochastic matrix, then its dominant eigenvalue $\lambda_1 = 1$, $|\lambda_2| \le \alpha$.

In the next cycle of the GArnoldi-PET method, v_1^{new} will be used as the initial vector for an m-step GArnoldi process, so that the new Krylov subspace

$$\mathcal{K}_m(A, v_1^{new}) = span(v_1^{new}, Av_1^{new}, \cdots, A^{m-1}v_1^{new})$$

is constructed. The following theorem shows the convergence of the GArnoldi-PET method.

Theorem 4. Assume that $v_1 = \sum_{i=1}^n \gamma_i x_i$ with respect to the eigenbasis $\{x_i\}_{i=1,2,\cdots,n}$ in which $\|x_i\|_2 = 1, i = 1, 2, \cdots, n$ and $\gamma_1 \ne 0$. Let $G = \mathrm{diag}\{d_1, d_2, \cdots, d_n\}$, $d_i > 0, i = 1, 2, 3, \cdots, n$, $S = [x_1, x_2, x_3, \cdots, x_n]$, and

$$\xi = \sum_{i=2}^{n} |\lambda_i - 1| \frac{|\gamma_i|}{|\gamma_1|}, \quad \zeta = \sqrt{\frac{\max_{1\le i\le n} d_i}{\min_{1\le i\le n} d_i}}.$$

Then

$$\|(A - I)u\|_G \le \frac{\xi \cdot \zeta}{\sigma_{min}(S)} \cdot \left(\frac{\alpha^{m_1-1}(\alpha - \mu + 1)}{2 - \mu}\right)^k \cdot \min_{p\in\mathcal{L}_{m-1}, p(\lambda_1)=1} \max_{\lambda\in\sigma(A)/\{\lambda_1\}} |p(\lambda)| ,$$

where u is taken from the Krylov subspace $\mathcal{K}_m(A, v_1)$, $\mu = 1 + \alpha\left(\frac{l}{n} - 1\right)$ with the number of dangling nodes l, $k \ge maxit$, $\sigma_{min}(S)$ is the smallest singular value of the matrix S.

Proof. Let $\{1, \pi_2, \cdots, \pi_n\}$ be the eigenvalue set of the matrix P, then $\{1, \lambda_2 = \alpha\pi_2, \cdots, \alpha_n = \alpha\pi_n\}$ is the eigenvalue set of the matrix A by Theorem 2. According to (3.3), we have

$$Tx_i = \begin{cases} (2 - \mu)x_i, & i = 1 \\ \lambda_i^{m_1-1}(\lambda_i - \mu + 1)x_i, & i = 2, 3, \cdots, n \end{cases}.$$

Let φ_i $(i = 1, 2, \cdots, n)$ be eigenvalues of the iterative matrix T. Then, we have

$$\varphi_i = \begin{cases} 2 - \mu, & i = 1 \\ \lambda_i^{m_1-1}(\lambda_i - \mu + 1), & i = 2, 3, \cdots, n \end{cases}. \tag{3.4}$$

Since $\mu = 1 + \alpha\left(\frac{l}{n} - 1\right)$ with the number of dangling nodes l, it has $1 - \mu = \alpha\left(1 - \frac{l}{n}\right) \geq 0$. On the other hand, according to Theorem 3 and $1 = |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, it has $|\lambda_i| \leq \alpha$ and

$$-\alpha + \mu - 1 \leq -\alpha + 1 - \mu \leq \lambda_i + 1 - \mu \leq \alpha + 1 - \mu, i = 2, \cdots, n.$$

Hence, we obtain $|\lambda_i - \mu + 1| \leq \alpha - \mu + 1, i = 2, \cdots, n$, and

$$|\varphi_i| \leq \alpha^{m_1 - 1}(\alpha - \mu + 1) < 2 - \mu, \quad i = 2, 3, \cdots, n. \tag{3.5}$$

For any $u \in \mathcal{K}_m(A, v_1^{new})$, there exists $q(x) \in \mathcal{L}_{m-1}$ such that

$$\begin{aligned}
\|(A - I)u\|_G &= \min_{q \in \mathcal{L}_{m-1}} \frac{\|(A - I)q(A)v_1^{new}\|_G}{\|q(A)v_1^{new}\|_G} \\
&= \min_{q \in \mathcal{L}_{m-1}} \frac{\|(A - I)q(A)\eta T^k v_1\|_G}{\|q(A)\eta T^k v_1\|_G} \\
&= \min_{q \in \mathcal{L}_{m-1}} \frac{\|(A - I)q(A)T^k\gamma_1 x_1 + \sum_{i=2}^n (A - I)q(A)T^k\gamma_i x_i\|_G}{\|\sum_{i=1}^n q(A)T^k\gamma_i x_i\|_G} \\
&= \min_{q \in \mathcal{L}_{m-1}} \frac{\|\sum_{i=2}^n (\lambda_i - 1)q(\lambda_i)\varphi_i^k\gamma_i x_i\|_G}{\|\sum_{i=1}^n q(\lambda_i)\varphi_i^k\gamma_i x_i\|_G},
\end{aligned} \tag{3.6}$$

where we used the facts that $\lambda_1 = 1$, $Ax_i = \lambda_i x_i, Tx_i = \varphi_i x_i, i = 1, 2, \cdots, n$. According to (3.2) and (3.5), for the numerator of (3.6), it has

$$\begin{aligned}
\left\|\sum_{i=2}^n (\lambda_i - 1)q(\lambda_i)\varphi_i^k\gamma_i x_i\right\|_G &\leq \sqrt{\max_{1 \leq i \leq n} d_i} \cdot \left\|\sum_{i=2}^n (\lambda_i - 1)q(\lambda_i)\varphi_i^k\gamma_i x_i\right\|_2 \\
&\leq \sqrt{\max_{1 \leq i \leq n} d_i} \cdot \sum_{i=2}^n |\lambda_i - 1| \cdot |\varphi_i^k| \cdot |\gamma_i| \cdot |q(\lambda_i)| \\
&\leq \sqrt{\max_{1 \leq i \leq n} d_i} \cdot \sum_{i=2}^n [\alpha^{m_1 - 1}(\alpha - \mu + 1)]^k \cdot |\lambda_i - 1| \cdot |\gamma_i| \cdot |q(\lambda_i)|.
\end{aligned} \tag{3.7}$$

For the denominator of (3.6), it has

$$\begin{aligned}
\left\|\sum_{i=1}^n q(\lambda_i)\varphi_i^k\gamma_i x_i\right\|_G^2 &\geq \min_{1 \leq i \leq n} d_i \cdot \left\|\sum_{i=1}^n q(\lambda_i)\varphi_i^k\gamma_i x_i\right\|_2^2 \\
&\geq \min_{1 \leq i \leq n} d_i \cdot \sigma_{min}^2(S) \cdot \sum_{i=1}^n |\varphi_i^k|^2 \cdot |\gamma_i|^2 \cdot |q(\lambda_i)|^2.
\end{aligned} \tag{3.8}$$

Combining (3.5), (3.7) and (3.8) into (3.6), we have

$$\begin{aligned}
\|(A - I)u\|_G &\leq \min_{q \in \mathcal{L}_{m-1}} \frac{\sqrt{\max_{1 \leq i \leq n} d_i} \cdot \sum_{i=2}^n [\alpha^{m_1 - 1}(\alpha - \mu + 1)]^k \cdot |\lambda_i - 1| \cdot |\gamma_i| \cdot |q(\lambda_i)|}{\sqrt{\min_{1 \leq i \leq n} d_i \cdot \sigma_{min}^2(S) \sum_{i=1}^n |\varphi_i^k|^2 \cdot |\gamma_i|^2 \cdot |q(\lambda_i)|^2}} \\
&\leq \frac{1}{\sigma_{min}(S)} \cdot \sqrt{\frac{\max_{1 \leq i \leq n} d_i}{\min_{1 \leq i \leq n} d_i}} \cdot \min_{q \in \mathcal{L}_{m-1}} \frac{\sum_{i=2}^n [\alpha^{m_1 - 1}(\alpha - \mu + 1)]^k \cdot |\lambda_i - 1| \cdot |\gamma_i| \cdot |q(\lambda_i)|}{(2 - \mu)^k \cdot |\gamma_1| \cdot |q(\lambda_1)|}
\end{aligned}$$

$$\leq \frac{1}{\sigma_{min}(S)} \cdot \sqrt{\frac{\max_{1\leq i\leq n} d_i}{\min_{1\leq i\leq n} d_i}} \cdot \left(\frac{\alpha^{m_1-1}(\alpha-\mu+1)}{2-\mu}\right)^k \cdot \min_{q\in\mathcal{L}_{m-1}} \sum_{i=2}^{n} |\lambda_i - 1| \frac{|\gamma_i|}{|\gamma_1|} \cdot \frac{|q(\lambda_i)|}{|q(\lambda_1)|},$$

where $\frac{\alpha^{m_1-1}(\alpha-\mu+1)}{2-\mu} < 1$. Let $p(\lambda) = q(\lambda)/q(1)$, where $p(1) = 1$, then we get

$$\|(A - I)u\|_G \leq \frac{\xi \cdot \zeta}{\sigma_{min}(S)} \cdot \left(\frac{\alpha^{m_1-1}(\alpha-\mu+1)}{2-\mu}\right)^k \cdot \min_{p\in\mathcal{L}_{m-1}, p(\lambda_1)=1} \max_{\lambda\in\sigma(A)/\{\lambda_1\}} |p(\lambda)|.$$

□

4. Numerical experiments

In this section, we test the effectiveness of the GArnoldi-PET method and compare it with the PET method (Algorithm 1) [6], the Power-Arnoldi algorithm (called as PA) [8] and the adaptive GArnoldi method (called as A-Arnoldi) [13] in terms of the number of matrix-vector products (Mv) and the computing time in seconds (CPU). All the numerical results are obtained by using MATLAB 2018b on the Windows 10 (64 bit) operating system with 1.7 GHz Intel(R) Core(TM) i5 CPU and RAM 4.00 GB.

In Table 1, we list the characteristics of test matrices including the matrix size (n), the number of nonzero elements (nnz), the number of dangling nodes ($numd$) and the density (den) which is defined by $den = \frac{nnz}{n\times n} \times 100$.

Table 1. The characteristic of test matrices.

Name	n	nnz	$numd$	den
wb-cs-stanford	9,914	36,854	2,861	0.375×10^{-1}
web-Stanford	281903	2,312,497	172	0.291×10^{-2}
wikipedia-20051105	1,634,989	19,753,078	72,556	0.739×10^{-3}

For a fair comparison, all algorithms use the same initial guess $x^{(0)} = v = e/n$ with $e = [1, 1, \cdots, 1]^{\mathrm{T}}$. The tolerance is chosen as $tol = 10^{-8}$. The values of the damping factor α are 0.990, 0.993, 0.995 and 0.997, respectively. The parameter $\beta = \alpha - 0.1$. We run the thick restarted Arnoldi procedure, with the number of approximate eigenpairs p, two times per cycle in the Power-Arnoldi method. Similarly, we run the adaptive GArnoldi procedure two times per cycle in the GArnoldi-PET method. In addition, for describing the efficiency of the GArnoldi-PET method, we define

$$\text{speedup} = \frac{\text{CPU}_{\text{PA}} - \text{CPU}_{\text{GArnoldi-PET}}}{\text{CPU}_{\text{PA}}} \times 100\%.$$

Example 1. The first test matrix is the wb-cs-stanford matrix, which contains 9914 pages, 36854 links and 2861 dangling nodes. It is available from https://sparse.tamu.edu/Gleich/wb-cs-stanford. In this example, we set the parameters $m = 5, p = 3, maxit = 6$ and $m_1 = 40$. Numerical results of the PET method, the adaptive GArnoldi method, the Power-Arnoldi algorithm and the GArnoldi-PET algorithm are reported in Table 2. Figure 1 plots the convergence history of the four methods with different values of α.

From Table 2, we can see that the Power-Arnoldi algorithm works better than the PET method and the adaptive GArnoldi method in terms of the number of matrix-vector products and the computing time. However, the GArnoldi-PET algorithm performs the best. For example, when $\alpha = 0.997$, the Power-Arnoldi algorithm needs 0.1473 seconds to reach the desired accuracy, while the GArnoldi-PET algorithm only uses 0.1038 seconds, and the speedup is 29.53%.

From Figure 1, it is easy to find that the GArnoldi-PET algorithm has a faster convergence speed than the PET method and Power-Arnoldi algorithm, even though its iteration counts are slightly inferior to the adaptive GArnoldi method. Obviously, only the number of iterations can not describe the whole story.

Table 2. Numerical results of the four methods on the wb-cs-stanford matrix.

α	PET	A-Arnoldi	PA	GArnoldi-PET
$\alpha = 0.99$				
Mv	712	290	169	158
CPU	0.1805	0.1589	0.0727	0.0692
speedup				4.81%
$\alpha = 0.993$				
Mv	960	350	238	194
CPU	0.2045	0.1862	0.0961	0.0801
speedup				16.65%
$\alpha = 0.995$				
Mv	1253	400	305	211
CPU	0.2916	0.1965	0.1325	0.0945
speedup				28.68%
$\alpha = 0.997$				
Mv	1804	530	362	255
CPU	0.3515	0.2778	0.1473	0.1038
speedup				29.53%

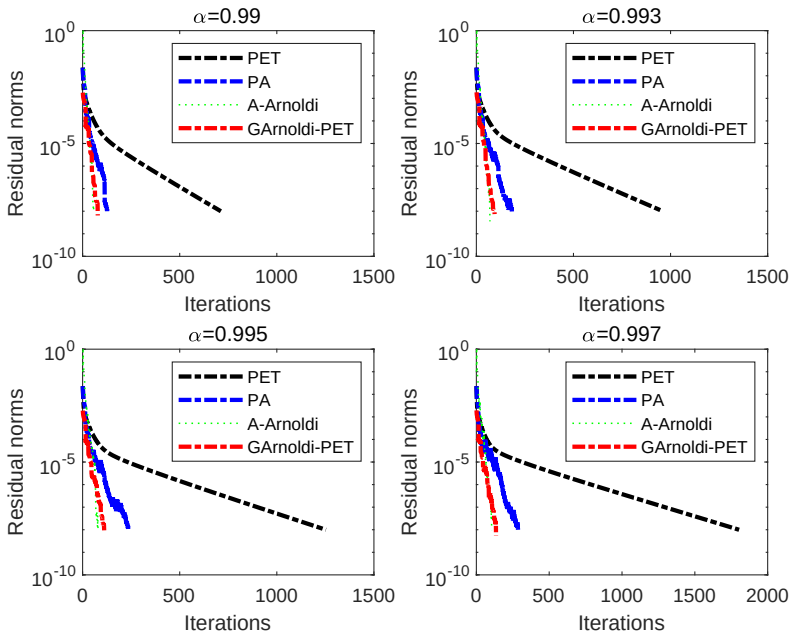

Figure 1. Convergence behaviors of the four methods on the wb-cs-stanford matrix.

Example 2. The second test matrix is the web-Stanford matrix, which contains 281903 nodes, 2312497 links and 172 dangling nodes. It is available from https://sparse.tamu.edu/SNAP/web-Stanford. In this example, we choose the parameters $m = 5, p = 3, maxit = 12$ and $m_1 = 35$. Numerical results of the PET method, the adaptive GArnoldi method, the Power-Arnoldi algorithm and the GArnoldi-PET algorithm are given in Table 3. Figure 2 depicts the convergence of the four methods with different values of α.

From Table 3, it observes that the GArnoldi-PET algorithm outperforms the other three methods in terms of the number of matrix-vector products and the computing time. Although the speedup is only 6.57% relative to the Power-Arnoldi algorithm when $\alpha = 0.993$. However, when α increases, e.g., $\alpha = 0.997$, the Power-Arnoldi algorithm needs 13.7626 seconds to reach the desired accuracy, the GArnoldi-PET algorithm only takes 10.7224 seconds, and the speedup becomes 22.09%.

From Figure 2, it shows that the GArnoldi-PET algorithm converges faster than the PET method and the Power-Arnoldi algorithm. When α is close to one, e.g., $\alpha = 0.995$ and $\alpha = 0.997$, the iteration counts of the GArnoldi-PET algorithm are less than those of the adaptive GArnoldi method. This suggests that our new algorithm has some potential.

Table 3. Numerical results of the four methods on the web-Stanford matrix.

α	PET	A-Arnoldi	PA	GArnoldi-PET
$\alpha = 0.99$				
Mv	717	715	303	273
CPU	11.4552	24.1209	7.4254	6.5659
speedup				11.58%
$\alpha = 0.993$				
Mv	966	905	395	349
CPU	14.6271	29.3734	9.0469	8.4525
speedup				6.57%
$\alpha = 0.995$				
Mv	1281	1185	464	397
CPU	19.6235	38.7536	11.6585	9.6601
speedup				17.14%
$\alpha = 0.997$				
Mv	1937	1795	601	481
CPU	29.5376	58.5102	13.7626	10.7224
speedup				22.09%

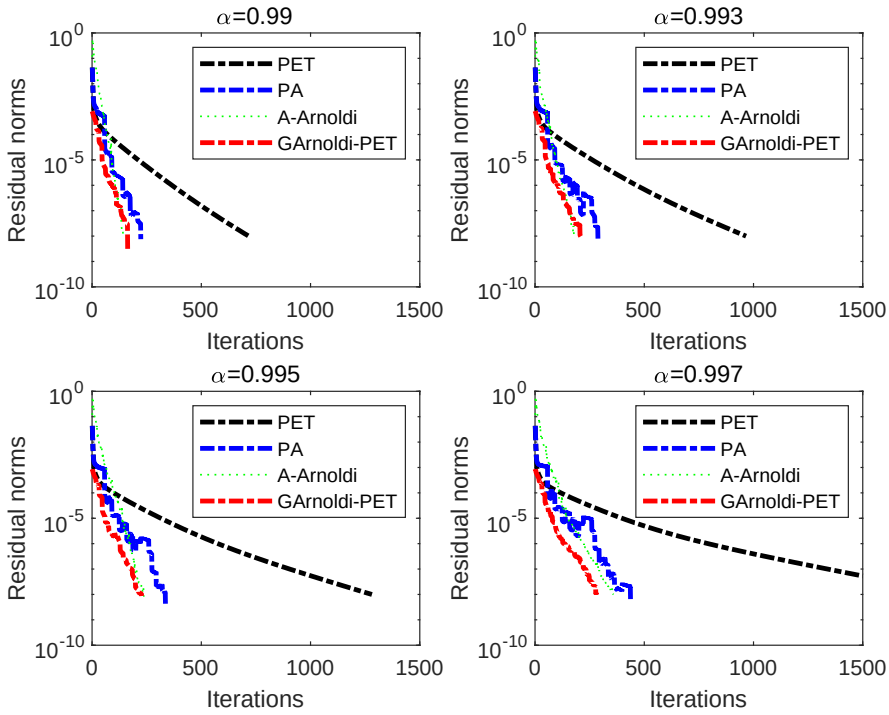

Figure 2. Convergence behaviors of the four methods on the web-Stanford matrix.

Example 3. The last test matrix is the wikipedia-20051105 matrix, which contains 1634989 nodes, 19753078 links and 72556 dangling nodes. It is available from https://sparse.tamu.edu/Gleich/wikipedia-20051105. In this example, we make the parameters $m = 5, p = 3, maxit = 8$ and $m_1 = 50$. Numerical results of the PET method, the adaptive GArnoldi method, the Power-Arnoldi algorithm and the GArnoldi-PET algorithm are listed in Table 4. Figure 3 shows the convergence curves of the four methods with different values of α.

From Table 4, we also see that the GArnoldi-PET algorithm makes great improvements on the PET method, the adaptive GArnoldi method and the Power-Arnoldi algorithm in terms of the number of matrix-vector products and the computing time. For a large damping factor such as $\alpha = 0.997$, the Power-Arnoldi algorithm takes 56.5771 seconds to reach the desired accuracy, while the GArnoldi-PET algorithm takes 41.9286 seconds to achieve the same accuracy, and the speedup is 25.89%.

From Figure 3, we again find that the GArnoldi-PET algorithm converges faster than the PET method, the adaptive GArnoldi method and the Power-Arnoldi algorithm. For different values of α, the iteration counts of the GArnoldi-PET algorithm are the least.

Table 4. Numerical results of the four methods on the wikipedia-20051105 matrix.

α	PET	A-Arnoldi	PA	GArnoldi-PET
$\alpha = 0.99$				
Mv	555	380	124	104
CPU	110.2113	113.5793	30.1254	25.1583
speedup				16.49%
$\alpha = 0.993$				
Mv	793	475	162	121
CPU	156.2311	139.5610	39.0841	29.0992
speedup				25.55%
$\alpha = 0.995$				
Mv	1111	595	171	141
CPU	221.5114	177.1960	43.4769	33.0091
speedup				24.08%
$\alpha = 0.997$				
Mv	1849	830	239	172
CPU	356.8225	247.7716	56.5771	41.9286
speedup				25.89%

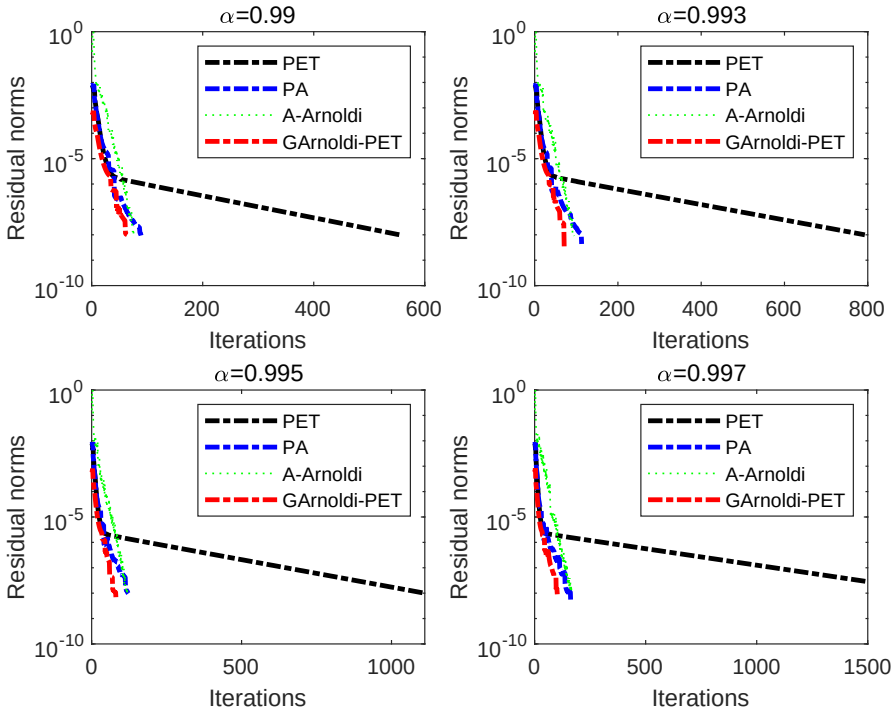

Figure 3. Convergence behaviors of the four methods on the wikipedia-20051105 matrix.

5. Conclusions

In this paper, by combining the PET method with the adaptive GArnoldi method, we propose a new method called as GArnoldi-PET method for accelerating the computation of PageRank problems. Its construction and theoretical analysis can be found in Section 3. Numerical results in Section 4 show that our proposed method is quite efficient and better than the existing methods, especially when the damping factor is close to 1. However, much research still needs further study. For example, determining the optimal choice of the parameters, or considering to use some preconditioning strategies as given in [18–20] for the GArnoldi method.

Acknowledgments

This research is supported by Science Challenge Project (TZ2016002-TZZT2019-B1.4), the Key Fund Project of Sichuan Provincial Department of Education (17za0003) and the Applied Basic Research Project of Sichuan Province (2020YJ0007). We would like to thank the anonymous referees for their valuable comments and suggestions.

Conflict of interest

All authors declare that there is no conflict of interest in this paper.

References

1. L. Page, S. Brin, R. Motwami, T. Winograd, *The PageRank citation ranking: Bringing order to the web, Technical report*, Computer Science Department, Stanford University, Stanford, CA, 1999.
2. S. Kamvar, T. Haveliwala, C. Manning, G. H. Golub, *Exploiting the block structure of the web for computing PageRank*, Technical Report, SCCM-03-02, Stanford University, 2003.
3. G. H. Golub, C. F. Van Loan, *Matrix Computations*, third ed., The Johns Hopkins University Press, Baltimore, London, 1996.
4. D. Gleich, A. Gray, C. Greif, T. Lau, An inner-outer iteration for computing PageRank, *SIAM J. Sci. Comput.,* **32** (2010), 349–371.
5. Z. L. Tian, Y. Liu, Y. Zhang, Z. Y. Liu, M. Y. Tian, The general inner-outer iteration method based on regular splittings for the PageRank problem, *Appl. Math. Comput.,* **356** (2019), 479–501.
6. X. Y. Tan, A new extrapolation method for pagerank computations, *J. Comput. Appl. Math.,* **313** (2017), 383–392 .
7. G. H. Golub, C. Greif, An Arnoldi-type algorithm for computing PageRank, *BIT.,* **46** (2006), 759–771.
8. G. Wu, Y. Wei, A Power-Arnoldi algorithm for computing pagerank, *Numer. Linear Algebra Appl.,* **14** (2007), 521–546.
9. R. Morgan, M. Zheng, A harmonic restarted Arnoldi algorithm for calculating eigenvalues and determining multiplicity, *Linear Algebra Appl.,* **415** (2006), 96–113.

10. Q. Y. Hu, C. Wen, T. Z. Huang, Z. L. Shen, X. M. Gu, A variant of the Power-Arnoldi algorithm for computing PageRank, *J. Comput. Appl. Math.,* **381** (2021), 113034.
11. A. Essai, Weighted FOM and GMRES for solving nonsymmetric linear systems, *Numer. Algorithms,* **18** (1998), 277–292.
12. H. S. Najafi, H. Ghazvini, Weighted restarting method in the weighted Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix, *Appl. Math. Comput.,* **175** (2006), 1276–1287.
13. J. F. Yin, G. J. Yin, M. Ng, On adaptively accelerated Arnoldi method for computing PageRank, *Numer. Linear Algebra Appl.,* **19** (2012), 73–85.
14. C. Wen, Q. Y. Hu, G. J. Yin, X. M. Gu, Z. L. Shen, An adaptive Power-GArnoldi algorithm for computing PageRank, *J. Comput. Appl. Math.,* **386** (2021), 113209.
15. Z. X. Jia, Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, *Linear Algebra Appl.,* **259** (1997), 1–23.
16. A. Langville, C. Meyer, *Googles PageRank and Beyond: The Science of the Search Engine Rankings*, Princeton University Press, 2006.
17. T. Haveliwala, S. Kamvar, *The second eigenvalue of the google matrix*, in: Proceedings of the Twelfth International World Wide Web of Conference, 2003.
18. F. Tudisco, C. Di Fiore, A preconditioning approach to the PageRank computation problem, *Linear Algebra Appl.,* **435** (2011), 2222–2246.
19. S. Cipolla, C. Di Fiore, F. Tudisco, Euler-Richardson method preconditioned by weakly stochastic matrix algebras: A potential contribution to Pagerank computation, *Electronic J. Linear Al.,* **32** (2017), 254–272.
20. G. Wu, Y. C. Wang, X. Q. Jin, A preconditioned and shifted GMRES algorithm for the PageRank problem with multiple damping factors, *SIAM J. Sci. Comput.,* **34** (2012), 2558–2575.

AIMS Press