BIT Numerical Mathematics
DOI: 10.1007/s10543-006-0091-y

AN ARNOLDI-TYPE ALGORITHM FOR COMPUTING PAGE RANK*

G. H. GOLUB[1,**] and C. GREIF[2,***]

[1] *SCCM Program, Gates 2B, Stanford University, Stanford CA 94305-9025, USA. email: golub@sccm.stanford.edu*

[2] *Department of Computer Science, The University of British Columbia, Vancouver B.C. V6T 1Z4, Canada. email: greif@cs.ubc.ca.*

Abstract.

We consider the problem of computing PageRank. The matrix involved is large and cannot be factored, and hence techniques based on matrix-vector products must be applied. A variant of the restarted refined Arnoldi method is proposed, which does not involve Ritz value computations. Numerical examples illustrate the performance and convergence behavior of the algorithm.

AMS subject classification (2000): 65F15, 65C40.

Key words: PageRank, power method, Arnoldi method, refined Arnoldi, singular value decomposition.

1 Introduction.

The iterative computation of Google's PageRank has been receiving a lot of attention in the last few years. The problem falls in the category of computing the stationary distribution vector of a finite Markov chain, defined as follows: compute a row vector π^T that satisfies $\pi^T = \pi^T A^T$ and $\|\pi\|_1 = 1$, where A^T is a row-stochastic matrix, i.e. a matrix whose entries are all between 0 and 1 and whose row-sums are all 1. For notational convenience, we will refer throughout to the equivalent problem of computing a column vector π for which $A\pi = \pi$. By the Perron–Frobenius theorem [2], if A is an irreducible nonnegative matrix then it has a positive real simple eigenvalue equal to its spectral radius, associated with a right nonnegative eigenvector. Furthermore, if A is strictly positive, then so is the eigenvector, and this is relevant to PageRank computations; see Section 2.

* Received October 26, 2004. Accepted in revised form April 18, 2006. Communicated by Lars Eldén.

** The work of this author was supported in part by the Department of Energy of the United States Government.

*** The work of this author was supported in part by the Natural Sciences and Engineering Research Council of Canada.

A comprehensive survey of numerical methods for Markov chains is provided in [28]. The computation of PageRank (see [19, 20, 21] and references therein) is a specific instance of the computation of a stationary distribution vector, with two distinct features. First, the matrices involved are so large that only a small set of computational tools can be applied. In particular, decomposition techniques cannot be considered, and algorithms based on matrix-vector products must be used. Secondly, the model incorporates a *damping factor* that defines a convex combination of the original matrix (after modifying zero-sum rows) and a rank-1 matrix. This effectively determines the magnitude of the second largest eigenvalue, and consequently the level of difficulty of the computation of the stationary vector, or the *PageRank* vector. The damping factor is a real positive number sufficiently close to (and always smaller than) 1. The smaller it is, the easier it is to compute the solution by simple means like the power method, and within a modest number of iterations. On the other hand, the closer the damping factor is to 1 the closer the matrix is to the original web link graph. If the largest eigenvalue (which is equal to 1) is not well separated from other eigenvalues, it is necessary to employ techniques that are more sophisticated than the power method.

The technique we discuss in this paper is based on the Arnoldi method [1]. Since the size of the problem restricts the scope of computational means, we consider a basic restarted approach. Our proposed algorithm is a variant of the refined Arnoldi method [15], with the feature that since the largest eigenvalue of the matrix is known, we use it as a shift and do not compute the largest Ritz value. As we show, the algorithm works well for a large range of values of the damping factor, and in particular values that are close to 1.

The remainder of this paper is organized as follows. In Section 2 we provide general background. In Section 3 we present the algorithm, discuss the computational work involved, and comment on the sensitivity of the PageRank problem. In Section 4 we provide experimental evidence of the effectiveness of our approach and in Section 5 we draw some conclusions.

2 Background.

2.1 Computing PageRank.

A detailed explanation of how the web link graph is formed can be found, for example, in [21]. The mathematical model for computing PageRank is based on the assumption that the importance of a webpage is determined by the importance of webpages that link to it.

Let w_i signify the importance (or *weight*) of page i. Then $w_i = \sum_{j \in B_i} \frac{w_j}{N_j}$, where B_i is the set of pages that link to page i and N_j is the number of outlinks from page j. Determining w_i is done iteratively. The initial guess is based on assuming that all pages are equally ranked, and in each iteration the importance of a page is re-calculated by taking into account the most recently assigned weights that the pages linking to it have. In mathematical terms, we set $w_i^{(0)} = \frac{1}{n}$,

and apply the iteration

$$w_i^{(k+1)} = \sum_{j \in B_i} \frac{w_j^{(k)}}{N_j}.$$

This iterative procedure amounts to applying the power method. The original PageRank algorithm was based on this approach, but was applied to a modified matrix rather than the original one. The modification addresses a few potential difficulties. One is the existence of *dangling nodes*: not all pages have outlinks. A second potential problem is the existence of *cyclic paths*: some pages might form a 'closed loop'. Indeed, any realistic web link graph is almost certainly reducible. This results in a reducible matrix that contains zero rows and has a non-unique stationary distribution vector.

Remedying the above difficulties is done as follows. First, all the zero rows of the matrix are modified so as to have a row-sum of 1 (see, e.g., [19] for details). Suppose the resulting matrix is P^T; then it is further modified to be

$$A(c) = cP + (1-c)E, \tag{2.1}$$

where E is a certain rank-1 matrix and c, the damping factor, is a positive parameter smaller than 1. In terms of the model, $1-c$ represents the probability that a surfer visiting a certain webpage will jump to a random page that is not an outlink of that page. The original choice of the damping factor was $c = 0.85$ [25, p. 11]. A possible choice of the $n \times n$ rank-1 matrix E is

$$E = e\, v^T; \quad e = [1, \ldots, 1]^T; \quad v = \frac{e}{n}. \tag{2.2}$$

The vector v is called the *personalization* or the *teleportation* vector and plays an important role in the model. A uniform v means that a random jump (not based on the available outlinks) is done to any available page with a uniform probability distribution. In our numerical experiments in this paper we take a uniform v, as in (2.2).

Notice that $A(c)$ (with $c < 1$) is a strictly positive matrix, and as such the stationary distribution vector is strictly positive. The actual PageRank vector is thus the positive stationary distribution vector of $A(c)$. The modification (2.1) makes it easier numerically to solve the problem, because while $P \equiv A(1)$ may have several eigenvalues on the unit circle, for $A(c), c < 1$, it is guaranteed that the second largest eigenvalue is equal to c [11]. A closed form characterization of $A(c)$ with identification of poles, the Jordan and the Schur canonical forms, are given in [12, 27]. The following useful result is proved in different ways in [27] and [6].

THEOREM 2.1. *Let P be a stochastic matrix with eigenvalues $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$. Then the eigenvalues of $cP + (1-c)eu^T$, where $0 < c < 1$ and u is a nonnegative vector with $e^T u = 1$ (i.e. a probability vector), are $\{1, c\lambda_2, c\lambda_3, \ldots, c\lambda_n\}$.*

The question what value of c is the 'correct' value and what gives a meaningful ranking is subject to ongoing investigation; see, e.g., [4] for a recent analysis. It

may therefore make sense to consider a large range of values. There are certain problems in which undamped web graphs are studied, see e.g. [30]. From a computational point of view, if c is sufficiently far from 1, as is the case for $c = 0.85$, then the separation of eigenvalues allows for an effective application of a simple technique like the power method. When the parameter is close to 1, the power method will perform poorly. It could then be accelerated; an effective technique of quadratic extrapolation has been derived in [19]. Further acceleration can be accomplished by adaptive techniques by which only components of the vector for which convergence has not yet been achieved are updated [18]. A variety of other effective methods can be found in the literature; see [13, 21] and references therein. The technique considered in this paper forms another possible approach of accelerating the computation of PageRank, and we will focus our attention on high values of c.

2.2 The Arnoldi algorithm.

We now briefly provide the essential details of the Arnoldi method for partial reduction to Hessenberg form. The algorithm is given in Figure 2.1. (See [5, p. 303].) It is a procedure of forming a truncated orthogonal similarity transformation. The input for the algorithm is a matrix A, an initial vector q and the number of steps k. (The matrix does not necessarily have to be provided explicitly; it is sufficient to have a routine that generates a matrix-vector product, for any given vector.) Upon exit, we have an $n \times (k+1)$ matrix Q_{k+1} with orthonormal columns and a $(k+1) \times k$ upper Hessenberg matrix $H_{k+1,k}$, that satisfy the relation $AQ_k = Q_{k+1}H_{k+1,k}$, where Q_k is the $n \times k$ matrix that contains the first k columns of Q_{k+1}. The columns of Q_k form an orthogonal basis for the Krylov subspace $\mathcal{K}^k(A, q) = \text{span}\{q, Aq, A^2q, \ldots, A^{k-1}q\}$. Many modern methods in numerical linear algebra rely on the Arnoldi algorithm for computing approximate solutions within Krylov subspaces. For more details on the Arnoldi method, see [5, 9, 26].

```
1 :  q_1 = q/‖q‖_2
2 :  for j = 1 to k
3 :       z = A q_j
4 :       for i = 1 to j
5 :            h_{i,j} = q_i^T z
6 :            z = z − h_{i,j} q_i
7 :       end for
8 :       h_{j+1,j} = ‖z‖_2
9 :       if h_{j+1,j} = 0, quit
10 :      q_{j+1} = z/h_{j+1,j}
11 :  end for
```

Figure 2.1: Arnoldi(A, q, k).

3 Arnoldi and refined Arnoldi algorithms for computing PageRank.

We first discuss two obvious applications of the restarted Arnoldi method that may give rise to certain difficulties when applied in the PageRank setting. We then introduce the algorithm we propose, which does not seem to have those problems.

A straightforward application of an explicitly restarted Arnoldi method is presented in Figure 3.1, where we pick a small dimension k, and repeatedly feed the Arnoldi algorithm the approximation computed in the previous truncated k-step procedure. Let us denote the eigenvalues of $H_{k,k}$, which are the *Ritz values*, by $\{\theta_i\}_{i=1}^{k}$. We need for our computation the dominant eigenpair, $\{\theta_M, v_M\}$, where $\theta_M = \max_i |\theta_i|$. A second possible approach is based on the refined Arnoldi algorithm [15, Section 3.2, Algorithm 1] and is presented in Figure 3.2. (See that paper and [16] for discussion and convergence analysis.) The approximations of the eigenvectors are not obtained by computing the eigenvectors of $H_{k,k}$ (the Ritz vectors). Rather, the singular vectors associated with the smallest singular values of $A - \theta_i I$ are *refined Ritz vectors*.

For computing PageRank, the two algorithms we have discussed so far may have the following potential difficulties. First, the largest Ritz value may be complex, and for such a large-scale problem it is undesirable to use complex

Set initial guess q and Arnoldi steps number k
Repeat
  $[Q_{k+1}, H_{k+1,k}] = Arnoldi(A, q, k)$
  Compute $H_{k,k} v_M = \theta_M v_M$
  Set $q = Q_k v_M$
Until $\|Aq - q\|_2 < \varepsilon$

Figure 3.1: A restarted Arnoldi algorithm for computing PageRank. The matrix $H_{k,k}$ is the square $k \times k$ matrix obtained by excluding the last row from $H_{k+1,k}$, and θ_M is the dominant eigenvalue of $H_{k,k}$, with associated eigenvector v_M.

Set initial guess q and Arnoldi steps number k
Repeat
  $[Q_{k+1}, H_{k+1,k}] = Arnoldi(A, q, k)$
  Compute $H_{k,k} v_M = \theta_M v_M$
  Compute $H_{k+1,k} - \theta_M \tilde{I} = U \Sigma V^T$
  Set $v = V_{*k}$
  Set $q = Q_k v$
Until $\sigma_{\min}(H_{k+1,k} - \theta_M \tilde{I}) < \varepsilon$

Figure 3.2: A restarted refined Arnoldi algorithm for computing PageRank. $\tilde{I}$ stands for an identity matrix augmented by a row of zeros. V_{*k} denotes the kth column of V. The matrix $H_{k,k}$ is the square $k \times k$ matrix obtained by excluding the last row from $H_{k+1,k}$, and θ_M is the dominant eigenvalue of $H_{k,k}$, with associated eigenvector v_M.

arithmetic if it can be avoided. Secondly, when the eigenvalues are not well separated (i.e. when c is close to 1), the largest Ritz value may converge slowly or irregularly to 1. The algorithm that we propose, presented in Figure 3.3, attempts to avoid these difficulties by incorporating a simple modification to the refined Arnoldi algorithm. We use the fact that the largest eigenvalue of A is known, and instead of computing the largest Ritz value we force the shift to be 1. Note that the smallest singular value of the shifted Hessenberg matrix is not equal to 0; rather, it converges to it throughout the iteration. Numerically we have observed that for our particular algorithm the smallest singular value converges to 0 more smoothly than the largest Ritz value converges to 1. Finally, since our algorithm does not entail Ritz value computations it is slightly faster than the refined Arnoldi algorithm. (Note though that the matrices involved are tiny and thus the savings are marginal.)

Set initial guess q and Arnoldi steps number k
Repeat
  $[Q_{k+1}, H_{k+1,k}] = Arnoldi(A, q, k)$
  Compute $H_{k+1,k} - \tilde{I} = U\Sigma V^T$
  Set $v = V_{*k}$
  Set $q = Q_k v$
Until $\sigma_{\min}(H_{k+1,k} - \tilde{I}) < \varepsilon$

Figure 3.3: Computation of PageRank using a variant of the refined Arnoldi algorithm that involves no Ritz values. $\tilde{I}$ stands for an identity matrix augmented by a row of zeros. V_{*k} denotes the kth column of V.

For variants of the Arnoldi algorithm it is natural to use the 2-norm as a measure of convergence. The stopping criterion for the algorithm (see Figure 3.3) does not entail any computational overhead, since the following holds.

PROPOSITION 3.1. *Consider the algorithm given in Figure* 3.3. *The stopping criterion* $\sigma_{\min}(H_{k+1,k} - \tilde{I}) < \varepsilon$ *is equivalent to* $\|Aq - q\|_2 < \varepsilon$.

The proof follows from [15, Theorem 3], with minor modifications, since

$$\begin{aligned}\min_{\|z\|_2=1} \|(A - I)Q_k z\|_2 &= \min_{\|z\|_2=1} \|Q_{k+1}(H_{k+1,k} - \tilde{I})z\|_2 \\ &= \min_{\|z\|_2=1} \|(H_{k+1,k} - \tilde{I})z\|_2 \\ &= \sigma_{\min}(H_{k+1,k} - \tilde{I}).\end{aligned}$$

Next, we consider the computational cost of the algorithm. Generally, the most computationally costly component of the algorithm are the matrix-vector products, which are applied k times in a k-step Arnoldi procedure. The cost largely depends on the number of nonzeros of the matrix. Inner products and other BLAS-1 operations are also performed in each iteration; see Table 3.1. In addition, we compute the singular value decomposition for the upper Hessenberg matrix, which is $\mathcal{O}(k^3)$ and is negligible, and the product $q = Q_k v$

(see Figure 3.3) which is approximately $2nk$ flops. (We note that the overall computational time depends on vectorization and parallelization in the particular computing environment and hence the table can only provide an indication of performance.) Storage is approximately $n \times (k+1)$ real entries. Thus, applying a certain number of steps of the proposed algorithm is more computationally intense than applying the same number of iterations of the power method, and we must keep k small. Note that some of the operations (e.g. inner products) are easily parallelizable and vectorizable. The hope is, then, that an Arnoldi-based algorithm will converge within fewer iterations than the power method in cases of a high damping factor, in which the power method converges slowly.

Table 3.1: Computational cost of the Arnoldi procedure. The first column refers to the line numbers in Figure 2.1. k is the number of Arnoldi steps taken, and ℓ is the average number of nonzeros per row in the matrix.

line	operation	times applied	cost (flops)
3	mat-vec products	k	$\approx 2nk\ell$
5	inner products	$k(k+1)/2$	$\approx nk(k+1)$
6	saxpy: $x + \alpha y$	$k(k+1)/2$	$\approx nk(k+1)$
8	2-norm computation	k	$\approx 2nk$
10	vector scaling	k	$\approx nk$

We end this section with a short discussion of the sensitivity of PageRank to the choice of the parameter c. Eigenvalue sensitivity analysis [23, 29], and in particular analysis for Markov chains [7, 10, 22] can be performed to assess how sensitive the stationary distribution eigenvector is to changes in the matrix. An investigation of the sensitivity of PageRank to the removal, addition, and change of pages can be found, for example, in [24]. An analysis of issues related to stability of PageRank is provided in [3], where a bound is given on the change in the 1-norm of the PageRank vector in relation to the change in the underlying web link graph. More observations on sensitivity can be found in [21, Section 7] and in [14].

Recall the definitions of $A(c), P, E, v$ and e in Equations (2.1) and (2.2). We know that $A(c)$ has an eigenvalue equal to 1: $A(c)x(c) = x(c)$, with $x(c)$ denoting PageRank. Differentiation with respect to c of the eigenvector equation gives $A'x + A\ x' = x'$, which is equivalent to $(I - A)x' = A'x$. By (2.1) we have $A' = P - E = \frac{1}{c}(A - E)$. It thus follows that $(I - A)x' = \frac{1}{c}(A - E)x$, which for E as in (2.2) and $\|x\|_1 = e^T x = 1$ can be further simplified to

$$(I - A)x' = \frac{1}{c}(x - v). \tag{3.1}$$

From (3.1) it is evident that the matrix involved is the same matrix that we are dealing with for computing the PageRank vector x. The linear system is singular

and consistent, and the right hand side contains the PageRank vector. Hence, accurately computing the sensitivity vector relies on accurate computation of PageRank. Notice also that since the PageRank vector x forms the null-space of $I - A$, a solution for x' may contain a component in the direction of x. The dependence of the sensitivity vector x' on the PageRank vector can be eliminated by projecting the vector onto the subspace orthogonal to x, applying a Gram-Schmidt step. Alternatively, imposing a condition such as $e^T x' = 1$ yields a nonsingular linear system whose associated matrix is $I - cP$. This matrix arises in linear system formulations of the PageRank problem, and its condition number is given in [17].

Computing x' may be meaningful since it is useful to know whether a small change of the parameter c will significantly affect the rank of a particular component of the PageRank vector. However, the computation of the sensitivity vector entails as much computational effort as the computation of PageRank.

4 Numerical experiments.

We have experimented with a few web matrices, and provide results for a representative subset; see Table 4.1. Since the 1-norm is a natural choice for the power method, in our comparisons we changed the stopping criterion given in Figure 3.3 and used the 1-norm instead, so as to perform a fair comparison.

Table 4.2 shows matrix-vector products for the `Stanford` matrix. Given the number of nonzeros of the matrix, k should be kept small, but we let it grow large just to illustrate the convergence behavior of the algorithm. For a fixed high value of c, as k increases there is a decrease in the iteration count, while storage requirements increase. Note that each Arnoldi iteration with k steps requires as many matrix-vector products, and hence the numbers in the table appear in integer multiples of k.

For c small ($c = 0.85$) it is evident that there is no improvement over the power method. This is consistent with similar observations about the effectiveness of the power method for low values of c in [19] for the quadratic extrapolation method and in [8] for approaches based on linear system formulations. On the other hand, for values of c close to 1 our algorithm proves much superior to the power method. For $c = 0.99$ we have approximately $66\%, 130\%$ and 230% more iterations for the power method compared to the algorithm with $k = 4, 8, 16$.

Table 4.1: Dimensions and number of nonzeros of some test matrices.

name	size	nz	avg nz per row
`Stanford`	281,903	2,312,497	8.2
`Stanford-Berkeley`	683,446	7,583,376	11.1
`wikipedia2005`	1,104,857	18,265,794	16.5
`edu`	2,024,716	14,056,641	6.9

Table 4.2: Matrix-vector products for various values of the damping factor c, for the 281,903 × 281,903 `Stanford` matrix. The stopping criterion was $\|x^{(k)} - Ax^{(k)}\|_1 < 10^{-7}$.

c	Power	$k = 4$	$k = 8$	$k = 16$
0.85	77	80	64	64
0.90	117	112	96	80
0.95	236	192	136	114
0.99	1165	700	504	352

Table 4.3: Matrix-vector products for various values of the damping factor c, for the 2,024,716 × 2,024,716 `edu` matrix. The stopping criterion was $\|x^{(k)} - Ax^{(k)}\|_1 < 10^{-7}$.

c	Power	$k = 8$
0.85	84	64
0.90	128	104
0.95	262	200
0.99	1331	920

Similar results were obtained for the `Stanford-Berkeley` matrix.

Table 4.3 shows the iteration counts for the larger `edu` matrix. In Figure 4.1 we take the value of the damping factor even further, to $c = 0.999$, and show for $k = 4$ that the algorithm yields substantial acceleration of convergence. The power method takes 7000 iterations to converge to a residual of approximately 5.7×10^{-5}. The algorithm with $k = 4$ takes 432 iterations, or 1728 matrix-

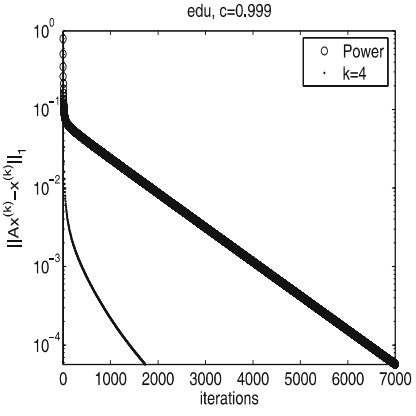

Figure 4.1: Convergence behavior for the 2,024,716 × 2,024,716 `edu` matrix, with $c = 0.999$.

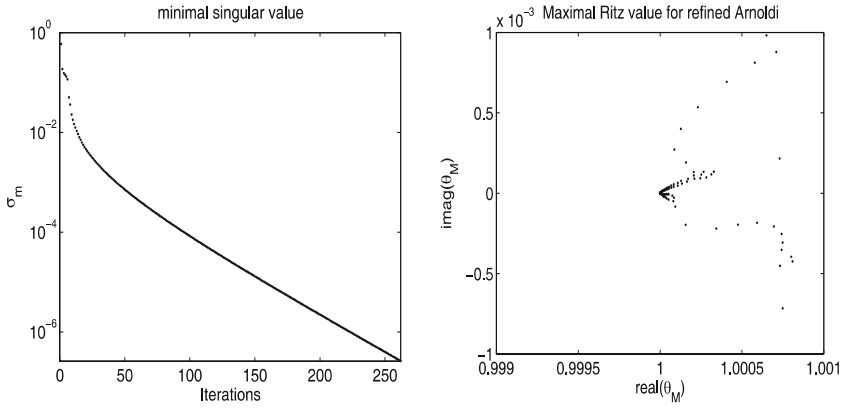

Figure 4.2: Extremal eigenvalues and singular values for the `edu` matrix, with $c = 0.99$ and $k = 4$. (Most of the Ritz values corresponding to the first 21 iterations were not within the bounds of the graph and are not shown.).

vector products, to converge to the same tolerance. In Figure 4.2 we show that while the smallest singular value converges smoothly to 0, the largest Ritz values throughout the iteration are typically complex. Their convergence in terms of magnitude is less smooth than the convergence of the minimal singular value. In the figure, the left graph shows the minimal singular values, using the algorithm. The right graph zooms on a neighborhood of 1 in the complex plane and shows the largest Ritz value of the refined Arnoldi algorithm throughout the iterative process.

Finally, we address the issue of the ranking as a function of the damping factor c, illustrating the potential importance of sensitivity analysis and of exploring robust algorithms (such as the Arnoldi approach) for various values of c. For the `wikipedia2005` matrix, we have looked at a few top ranked entries and how they

Table 4.4: Rankings of a few entries in `wikipedia2005` as a function of c. These rankings are for a particular test matrix, and do not necessarily reflect true rankings of Wikipedia pages..

Entry	$c = 0.85$	$c = 0.90$	$c = 0.95$	$c = 0.99$
United States	1	1	1	1
Race (U.S. Census)	2	2	4	20
United Kingdom	3	3	2	2
France	4	4	5	7
2005	5	5	11	10
Category: politics	13	7	6	5
Category: wikiportals	18	8	3	3

change with the damping factor. Table 4.4 illustrates this. Certain entries have their rank decrease as c grows whereas other entries have their rank go higher as c increases. Even close the top the sensitivity could be high. For example, while the top ranked entry stays at the top throughout, the second ranked entry for $c = 0.85$ has a much lower rank for $c = 0.99$, and on the other hand the entry ranked 18th for $c = 0.85$ is ranked third for $c = 0.99$.

5 Conclusions and future work.

We have introduced and explored the performance of an algorithm based on the Arnoldi method for computing PageRank. It is a variant of the refined Arnoldi method which computes PageRank without using Ritz values. The strength of Arnoldi-type methods lies in obtaining orthogonal vectors. It allows for effective separation of the PageRank vector from other approximate eigenvectors. The algorithm shows robust performance for PageRank computations that holds for a large variety of values of c.

For $c = 0.85$ our algorithm converges quickly but the separation of the second eigenvalue from the first is sufficiently good for the power method to perform extremely well. On the other hand, for high values of the damping factor the method we propose yields substantial computational savings (at the price of higher memory requirements). It should also be noted that the algorithm is applicable to any Markov chain, not only to the PageRank problem.

Acknowledgments.

We are grateful to David Gleich for his constructive comments on this manuscript, helpful discussions, and for providing us with a few web matrices, including the `wikipedia2005` matrix. We also thank Yahoo! Research Labs for the `edu` matrix, Lars Eldén for his input on eigenvalue sensitivity, and three anonymous referees for their helpful comments and suggestions.

REFERENCES

1. W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
2. A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, PA, 1994.
3. M. Bianchini, M. Gori, and F. Scarselli, *Inside PageRank*, ACM Trans. Internet Technol., 5 (2005), pp. 92–128.
4. P. Boldi, M. Santini, and S. Vigna, *PageRank as a function of the damping factor*, in Proceedings of the Forteenth International World Wide Web Conference, Chiba, Japan, pp. 557–566, ACM Press, New York, 2005.
5. J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
6. L. Eldén, *A note on the eigenvalues of the Google matrix*, Report LiTH-MAT-R-04-01, Linköping University, 2003.

7. R. E. Funderlic and C. D. Meyer, *Sensitivity of the stationary distribution vector for an ergodic Markov chain*, Linear Algebra Appl., 76 (1986), pp. 1–17.
8. D. Gleich, L. Zhukov, and P. Berkhin, *Fast parallel PageRank: a linear system approach*, Yahoo! Research Technical Report YRL-2004-038, available via `http://research.yahoo.com/publication/YRL-2004-038.pdf`, 2004.
9. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edn., Johns Hopkins University Press, Baltimore, MD, 1996.
10. G. H. Golub and C. D. Meyer, *Using the QR factorization and group inverse to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Algebra Discr., 17 (1986), pp. 273–281.
11. T. H. Haveliwala and S. D. Kamvar, *The second eigenvalue of the Google matrix*, Technical report 2003-20, Stanford University, available via `http://dbpubs.stanford.edu:8090/pub/2003-20`, 2003.
12. R. Horn and S. Serra-Capizzano, *Canonical forms for certain rank one perturbations and an application to the Google PageRanking problem*, Internet Math., to appear.
13. I. C. F. Ipsen and S. Kirkland, *Convergence analysis of an updating algorithm by Langville and Meyer*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 952–967.
14. I. C. F. Ipsen and R. S. Wills, *Mathematical properties and analysis of Google's PageRank*, Bol. Soc. Esp. Mat. Apl., 34 (2006) pp. 191–196.
15. Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.
16. Z. Jia and G. W. Stewart, *An analysis of the Rayleigh–Ritz method for approximating eigenspaces*, Math. Comp., 70 (2001), pp. 637–647.
17. S. D. Kamvar and T. H. Haveliwala, *The condition number of the PageRank problem*, Technical report 2003-36, Stanford University, available via `http://dbpubs.stanford.edu:8090/pub/2003-36`, 2003.
18. S. D. Kamvar, T. H. Haveliwala, and G. H. Golub, *Adaptive methods for the computation of PageRank*, Linear Algebra Appl., 386 (2004), pp. 51–65.
19. S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, 2003, available also as Stanford Technical Report SCCM-02-15, ACM Press, New York, 2003.
20. A. N. Langville and C. D. Meyer, *Deeper inside PageRank*, Internet Math., 1 (2005), pp. 335–380.
21. A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.
22. C. D. Meyer, *The character of a finite Markov chain*, in Linear Algebra, Markov Chains, and Queueing Models, IMA Vol. Math. Appl., C. D. Meyer and R. J. Plemmons, eds., pp. 47–58, vol. 48, Springer, Berlin, 1993.
23. C. D. Meyer and G. W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.
24. A. Y. Ng, A. X. Zheng, and M. I. Jordan, *Link analysis, eigenvectors and stability*, in Seventeenth International Joint Conference on Artificial Intelligence, pp. 903–910, Morgan Kaufmann, Seattle, WA, 2001.
25. L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: bringing order to the web*, Technical report 1999-66, Stanford University, Stanford Digital Libraries, available via
http://dbpubs.stanford.edu:8090/pub/1999-66, 1999.
26. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edn., SIAM, Philadelphia, PA, 2003.
27. S. Serra-Capizzano, *Jordan canonical form of the Google matrix: a potential contribution to the PageRank computation*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 305–312.

28. W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

29. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

30. D. Zhou, J. Huang, and B. Schölkopf, *Learning from labeled and unlabeled data on a directed graph*, in Proceedings of the 22nd International Conference on Machine Learning, Bonn, 2005, ACM Press, New York, 2005.