Research Report

KSTS/RR-15/003
March 9, 2015

Lanczos type method for computing PageRank

by

Kazuma Teramoto
Takashi Nodera

Kazuma Teramoto
Graduate School of Science and Technology
Keio University

Takashi Nodera
Department of Mathematics
Keio University

Department of Mathematics
Faculty of Science and Technology
Keio University

KSTS/RR-15/003
March 9, 2015

Lanczos type method for computing PageRank

Kazuma Teramoto[1] Takashi Nodera[2]

March 1, 2015

Abstract

Computing eigenvalues and eigenvectors of a large matrix is one of the most important tasks in numerical analysis. PageRank is a link analysis algorithm used by the internet search engine, Google, which ranks each document in the order of its relative importance in its database. In this paper, a new algorithm for computing the PageRank vector is proposed, using a combination of the Lanczos bi-orthogonalization algorithm with a semi-orthogonality and a SVD (singular-value decomposition). This method converges faster than the Arnoldi method requiring less computation time. The results of some numerical experiments have been documented to evaluate the effectiveness of our proposed Lanczos algorithm.

key words. PageRank, Lanczos method, semi-orthogonality, eigenvalue and eigenvector
AMS(MOS) subject classifications. 65F10, 65M12

Contents

1 Introduction

PageRank is the essential approach for ranking a Web page where the status of a page is determined according to its link structure on the Web. This model has been used by Google as a part of its contemporary search engine equipment. Nowadays, the precise ranking procedures and computation schemes used by Google are no longer public, but the

Algorithm 1: Arnoldi Method

```
Compute q_1 = q_0/||q_0||_2;
for j = 1, 2, ..., to m do
    Compute w_j = A q_j;
    for k = 1, 2... to j do
        h_kj = q_k^T w_j;
        w_j = w_j − h_kj q_k;
    end
    Compute h_{j+1,j} = ||w_j||_2;
    if h_{j+1,j} = 0 then
        stop and exit;
    else
        Set q_{j+1} = w_j / h_{j+1,j};
    end
end
```

PageRank model has taken on a life of its own and has received important consideration in the science and technology communities in the past ten years. PageRank is essentially a fixed distribution vector of the Markov chain wherein the transition matrix is a convex combination of the Web link graph and a precise rank-1 matrix. A major parameter in the model is a 'damping factor', a scalar that determines the weight given to the Web link graph in the model. The weighted PageRank constitutes the elements of the dominant eigenvector of the modified adjacency matrix as follows:

$$A = \alpha P + (1 - \alpha)E, \tag{1}$$

where P is a column stochastic matrix, α is a 'damping factor', and E is a rank one matrix. The specified derivation is detailed in a paper by Kamvar et al. [8].

More recently, the computation of the eigenpair (eigenvalue and eigenvector) of non-symmetric matrices have become one of the most important tasks in many science and technology applications. A typical example, nowadays, is the computation of PageRank based on the link structure of the Web. Due to the great size and sparsity of the matrix, factorization schemes are considered impractical, and iterative schemes are used, where the computation is dominated by matrix-vector products. Detailed descriptions of this problem are available, and the algorithms can be found in numerous references, e.g. [2, 7, 6, 8, 9, 10, 13].

The power method was firstly considered for computing PageRank. For detailed properties of PageRank using the power method, please refer to Kamvar et al. [8]. However, the power method has its disadvantages, i.e. for some given matrices, the power method converges very slowly. Although different methods have been suggested for accelerating convergence, there have been no significant improvements so far. For example, a procedure using orthogonalization such as the Arnoldi method has been suggested [7]. Teratomo et al. [13] have published studies which suggest the Lanczos method is an effective procedure to compute the PageRank vector. In this paper, we have studied a new algorithm for computing the PageRank vector, using a combined Lanczos bi-orthogonalization algorithm and SVD (Singular Value Decomposition).

Algorithm 2: Arnoldi-Type Method

Choose $\boldsymbol{q}_0$ with $\|\boldsymbol{q}_0\|_2 = 1$;
for $l = 1, 2, ...,$ *until convergence* **do**
    Compute $[Q_m, H_{m+1,m}]$=Arnoldi$(A, \boldsymbol{q}_0, m)$;
    Compute singular value decomposition $H_{m+1,m} - [I; 0] = U\Sigma V^T$;
    Compute $\boldsymbol{q}_0 = Q_m \boldsymbol{v}_m$;
    Compute $\boldsymbol{r} = \sigma_m Q_{m+1} \boldsymbol{u}_m$;
    if $\|\boldsymbol{r}\|_1 < TOL$ **then**
        stop and exit;
    end
end

The remainder of the paper is organized as follows: Section 2 includes a brief description of the Arnoldi method of the PageRank vector. In section 3, a new Lanczos algorithm with a SVD scheme is proposed. In section 4, the results of numerical experiments obtained through running MATLAB codes are shown. The conclusion follows.

2 Arnoldi Method

In this section, a brief introduction of the Arnoldi method [1] for computing the PageRank vector is given. The Arnoldi method, which uses Algorithm 1, builds an orthonormal basis for the Krylov subspace given by:

$$\mathcal{K}_m(A, \boldsymbol{q}_0) = \text{span}\{\boldsymbol{q}_0, A\boldsymbol{q}_0, ..., A^{m-1}\boldsymbol{q}_0\}, \tag{2}$$

where the Krylov subspace is restricted to a fixed dimension m and $\boldsymbol{q}_0$ is an initial vector which satisfies $\|\boldsymbol{q}_0\| = 1$. From Algorithm 1, the following relations hold:

$$\begin{aligned} AQ_m &= Q_m H_m + h_{m+1,m}\boldsymbol{q}_{m+1}\boldsymbol{e}_m^T, \\ Q_m^T A Q_m &= H_m, \end{aligned}$$

where $Q_m = [\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_m] \in R^{n\times m}$ is a column-orthogonal matrix, and $H_m = \{h_{i,j}\} \in R_{m\times m}$ is a Hessenberg matrix [7]. Since H_m is an orthogonal projection from A to $\mathcal{K}_m$, the eigenvalue of H_m can be used as an approximate eigenvalue of A. If $\boldsymbol{y}$ is the eigenvector of H_m, then $Q_m\boldsymbol{y}$ is the approximate eigenvector of A. This is true because it has been established that the largest eigenvalue of a PageRank matrix is 1. The Arnoldi-type method was proposed by Golub and Greif [5] and it will be referred to as Algorithm 2. In Algorithm 2, a singular value decomposition is computed normally, instead of the eigenvalue of H_m [7]. When m increases, the total computation cost of this method continuously increases with every cycle, while the total iterations decrease. It can be very difficult to choose m a priori and if too small a value is chosen, the convergence may stall. Consequently, it is difficult to choose the optimal value of m to minimize total computation cost (CPU time).

3 Lanczos Method with Semi-Orthogonality

In this section, a modified Lanczos method with semi-orthogonality is proposed. This algorithm is derived from using the result of Day [3]. This can be used when a non-

Hermitian matrix is converted to a tridiagonal matrix and one of the Krylov subspace methods. Given two starting vectors, this algorithm computes simultaneously, a basis for the following two subspaces, where, matrix A is a $n \times n$ PageRank matrix, $\boldsymbol{p}^*$ and $\boldsymbol{q}$ are initial vectors and the conjugate transpose is denoted by the superscript "*". Unlike the classical Lanczos method [13], this algorithm has an advantage because it can perform computations without having to set the value of m:

$$\mathcal{K}_m(A, \boldsymbol{q}) = \text{span}\{\boldsymbol{q}, A\boldsymbol{q}, ..., A^{m-1}\boldsymbol{q}\},$$
$$\mathcal{K}_m(A, \boldsymbol{p}^*) = \text{span}\{\boldsymbol{p}^*, \boldsymbol{p}^*A, ..., \boldsymbol{p}^*A^{m-1}\}.$$

Set $\boldsymbol{p}* = \boldsymbol{p}_1^*, \boldsymbol{q} = \boldsymbol{q}_1$. After j successful Lanczos steps, the matrices P and Q are produced. The rows of P_j^* span $\mathcal{K}_m^j(A, \boldsymbol{p}^*)$ and n the column of Q_j span $\mathcal{K}_m^j(A, \boldsymbol{q})$. The matrix $T_j = P_j^* A Q_j$ is tridiagonal. This means that $\Omega_j = P_j^* Q_j$ is diagonal. Let P_j^* and Q_j have full rank. If $P_j^* Q_j$ is invertible then,

$$\Pi_j = Q_j \Omega_j^{-1} P_j^* \tag{3}$$

and is an oblique projector ($\Pi_j^2 = \Pi_j$) onto Range(Q_j). Generally, it is not orthogonal ($\Pi_j^* \neq \Pi_j$), and

$$\{\boldsymbol{u}^*\Pi_j,\ \boldsymbol{u} \in \mathbf{C}^j\} = \text{Range}(P_j)^* \tag{4}$$

is also an oblique projector onto the dual space, where $\mathbf{C}^j$ is a j dimensional complex space. From equation (3) and (4), $\Pi_j A \Pi_j$ is a projection from A onto the dual Range(Q_j) and Range$(P_j)^*$. Assuming that P_j^* and Q_j exist, the description of A with regard to the basis $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$ is $Q_n^{-1} A Q_n$. $\Pi_n = I$ implies that $Q_n^{-1} = \Omega^{-1} P_n^*$ and $Q_n^{-1} A Q_n = \Omega_n^{-1} T_n$. The tridiagonal matrix $\Omega_j^{-1} T_j$ represents $\Pi_j A \Pi_j$ in the bases $(\boldsymbol{q}_1, ..., \boldsymbol{q}_j)$ and $(\omega_1^{-1}\boldsymbol{p}_1^*, ..., \omega_j^{-1}\boldsymbol{p}_j^*)$. The next step is to consider how to generate Lanczos vectors. Lanczos vectors can compute the following equations:

$$\beta_2 \boldsymbol{p}_2^* = \boldsymbol{p}_1^* A - \frac{\alpha_1}{\omega_1}\boldsymbol{p}_1^*, \tag{5}$$

$$\beta_{m+1}\boldsymbol{p}_{m+1}^* = \boldsymbol{p}_m^* A - \frac{\alpha_m}{\omega_m}\boldsymbol{p}_m^* - \frac{\gamma_m \omega_m}{\omega_{m-1}}\boldsymbol{p}_{m-1}^*, \tag{6}$$

$$\gamma_2 \boldsymbol{q}_2 = A\boldsymbol{q}_1 - \frac{\alpha_1}{\omega_1}\boldsymbol{q}_1, \tag{7}$$

$$\gamma_{m+1}\boldsymbol{q}_{m+1} = A\boldsymbol{q}_m - \frac{\alpha_m}{\omega_m}\boldsymbol{q}_m - \frac{\beta_m \omega_m}{\omega_{m-1}}\boldsymbol{q}_{m-1}. \tag{8}$$

The initial condition is $\omega_m = \boldsymbol{p}_m^* \boldsymbol{q}_m, \alpha_m = \boldsymbol{p}_m^* A \boldsymbol{q}_m$. With the intention of making the size of the Lanzcos vector 1, β and γ are scaled as follows:

$$\beta_{m+1} = \left|\left|\boldsymbol{p}_m^* A - \frac{\alpha_m}{\omega_m}\boldsymbol{p}_m^* - \frac{\gamma_m \omega_m}{\omega_{m-1}}\boldsymbol{p}_{m-1}^*\right|\right|_2, \tag{9}$$

$$\gamma_{m+1} = \left|\left|A\boldsymbol{q}_m - \frac{\alpha_m}{\omega_m}\boldsymbol{q}_m - \frac{\beta_m \omega_m}{\omega_{m-1}}\boldsymbol{q}_{m-1}\right|\right|_2. \tag{10}$$

When representing the recurrence formula in matrix form, the following equation holds:

$$P_m^* A = T_m \Omega^{-1} P_m^* + \boldsymbol{e}_m \beta_{m+1} \boldsymbol{p}_{m+1}^*, \tag{11}$$

KSTS/RR-15/003
March 9, 2015

Algorithm 3: Lanczos Method with Local Orthogonality

1 Start: $\boldsymbol{p}_1 = \boldsymbol{p}/\|\boldsymbol{p}\|_2, \boldsymbol{q}_1 = \boldsymbol{q}/\|\boldsymbol{q}\|_2, \omega_0 = 1, \beta_1 = \gamma_1 = 0, \omega_1 = \boldsymbol{p}_1^*\boldsymbol{q}_1$;
2 **for** $l = 1, 2, ..., MaxStep$ **do**
3 $\quad \boldsymbol{r}_i^* = \boldsymbol{p}_i^* B - \frac{\gamma_i \omega_i}{\omega_{i-1}} \boldsymbol{p}_{i-1}^*$,
4 $\quad \boldsymbol{s}_i = A\boldsymbol{q}_i - \boldsymbol{q}_{i-1}\frac{\beta_i \omega_i}{\omega_{i-1}}$;
5 $\quad \alpha_i = \boldsymbol{r}_i^* \boldsymbol{q}_i$;
6 $\quad \boldsymbol{r}_i^* := \boldsymbol{r}_i^* - \frac{\alpha_i}{\omega_i}\boldsymbol{p}_i^*,\ \boldsymbol{s}_i := \boldsymbol{s}_i - \boldsymbol{q}_i \frac{\alpha_i}{\omega_i}$;
7 $\quad \alpha_i^l = \boldsymbol{r}_i^* \boldsymbol{q}_i,\ \alpha_i^r = \boldsymbol{p}_i^* \boldsymbol{s}_i$;
8 $\quad \boldsymbol{r}_i^* := \boldsymbol{r}_i^* - \frac{\alpha_i^l}{\omega_i}\boldsymbol{p}_i^*,\ \boldsymbol{s}_i := \boldsymbol{s}_i - \boldsymbol{q}_i \frac{\alpha_i^r}{\omega_i}$;
9 $\quad \beta_{i+1} = \|\boldsymbol{r}_i^*\|_2,\ \gamma_{i+1} = \|\boldsymbol{s}_i\|_2$;
10 $\quad \boldsymbol{p}_{i+1}^* = \boldsymbol{r}_i^*/\beta_{i+1},\ \boldsymbol{q}_{i+1} = \boldsymbol{s}_i/\gamma_{i+1}$;
11 $\quad \omega_{i+1} = \boldsymbol{p}_{i+1}^* \boldsymbol{q}_{i+1}$;
12 $\quad$ check for breakdown:$|\omega_{m+1}| < (n + 10(m+1))\epsilon$;
13 $\quad$ check for convergence;
14 **end**

$$AQ_m = Q_m \Omega^{-1} T_m + \boldsymbol{q}_{m+1}\gamma_{m+1}\boldsymbol{e}_m^* \cdot, \tag{12}$$

The tridiagonal matrix can compute as follows:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2\omega_2 & & & \\ \gamma_2\omega_2 & \alpha_2 & \beta_3\omega_3 & & \\ & & \cdots & & \\ & & \gamma_{m-1}\omega_{m-1} & \alpha_{m-1} & \beta_m\omega_m \\ & & & \gamma_m\omega_m & \alpha_m \end{pmatrix}.$$

Using these equations, we can set the Lanczos algorithm as Algorithm 3. The local orthogonality is defined as follows:

Definition 3.1(Day [3]) Lanczos vectors $\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_i^*$ and $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_i$ satisfy the local orthogonality of the following equations in i steps:

$$|\cos\angle \boldsymbol{p}_i^* \boldsymbol{q}_{i-1}| \leq 4\epsilon, \tag{13}$$

$$|\cos\angle \boldsymbol{p}_{i-1}^* \boldsymbol{q}_i| \leq 4\epsilon. \tag{14}$$

In order to satisfy equations (13) and (14), it is necessary to maintain the bi-orthogonality between $\boldsymbol{p}_{i+1}$ and $\boldsymbol{q}_i$, and between $\boldsymbol{q}_{i+1}$, and $\boldsymbol{p}_i$, in line 6 and 7 of Algorithm 6. When the closely bi-orthogonal vectors are improved, then the cosine of the angles between the new vectors is detached from the dimension. These bi-orthogonalities are generally referred to as a local orthogonality.

The right Ritz vector which can compute this algorithm is the PageRank vector. The convergence criterion will be discussed at length in the next section.

3.1 Computing PageRank with Semi-Orthogonality

In this subsection, a modified Lanczos method for computing PageRank with a semi-orthogonality will be introduced. The eigenvalues of A are approximated using the eigenvalues of the pair (T_m, Ω_m). The pair (T_m, Ω_m) will be used to compute PageRank. The

Algorithm 4: Lanczos Method for Computing PageRank

```
Choose initial vectors p, q such that(p, q) = 1;
[T,P,Q]=Lanczos(A, p, q);
Compute singular value decomposition T_m − I = UΣV^T;
if Convergence condition satisfies then
    Compute PageRank vector Q_m v_m;
end
continue to Lanczos method;
```

eigentriplet of (T_m, Ω_m) is as follows:

$$\begin{aligned} \boldsymbol{u}_T^* T_m &= \theta_T \boldsymbol{u}_T^* \Omega_m, \\ T_m \boldsymbol{v}_T &= \theta_T \Omega_m \boldsymbol{v}_T. \end{aligned}$$

The eigentriplet of matrix A is approximated as follows:

$$\lambda \simeq \theta_T, \ \boldsymbol{y}^* \simeq \boldsymbol{u}_T^* P_m^*, \ \boldsymbol{x} \simeq Q_m \boldsymbol{v}_T.$$

where, θ_T is a Ritz value, $\boldsymbol{u}_T^* P_m^*$ is a left Ritz vector, and $Q_m \boldsymbol{v}_T$ is a right Ritz vector. In this paper, when a Lanczos algorithm converges, the right Ritz vector is regarded as the PageRank vector. The following equation has been used as the convergence criterion:

$$\frac{||A\boldsymbol{x} - \lambda \boldsymbol{x}||_2}{||A||_1} < \epsilon. \tag{15}$$

Here, the definition of the Ritz value and Ritz vector, $A\boldsymbol{x} - \lambda\boldsymbol{x}$ can be represented as follows:

$$A\boldsymbol{x} - \lambda \boldsymbol{x} \approx A Q_m \boldsymbol{v}_T - \theta_T Q_m \boldsymbol{v}_T. \tag{16}$$

Substitute equation(16) for equation (12), and the following equation is satisfied:

$$A\boldsymbol{x} - \lambda \boldsymbol{x} \approx \boldsymbol{q}_{m+1} \gamma_{m+1} \boldsymbol{e}_m^* \boldsymbol{v}_T.. \tag{17}$$

So even if only the value of $\boldsymbol{v}_T$ is known, it is possible to compute the convergence criterion easily. From the above, the modified Lanczos method with semi-orthogonality with SVD is Algorithm 4.

4 Numerical Experiments

In this section, the numerical results of the two methods described in the previous sections on the test problems were compared. All computing of the numerical experiments were run on a PC with 3.6 GHz and an eight-gigabyte memory using MATLAB R2012b. These results are shown to demonstrate the efficiency of the Lanczos algorithm with a semi-orthogonality. The test matrices, Death_Penalty and E-mail Enron, were obtained from Web pages: Stanford University Large Network Data Set Collection and University of Toronto Data sets for Link Analysis Ranking Experiments.

Table 1: Example 1, Iterations and Computation Time (4298×4298)

	Power		Arnoldi		Lanczos		LanLD	
c	IT	Time	IT	Time	IT	Time	IT	Time
0.85	55	1.06	8	0.453	10	0.677	5	0.235
0.90	76	1.48	9	0.516	10	0.677	6	0.264
0.95	119	2.28	11	0.625	10	0.677	5	0.235
0.99	226	4.33	13	0.765	10	0.677	7	0.289

Table 2: Example 2, Iterations and Computation Time (15000×15000)

	Arnoldi(m=10)		Lanczos		LanLD	
c	IT	Time	IT	Time	IT	Time
0.85	4	4.16	10	5.23	6	2.78
0.90	6	5.02	10	5.23	7	3.07
0.95	9	7.52	10	5.23	7	3.07
0.99	17	14.09	10	5.23	7	3.07

4.1 Example 1

The Power method, Arnoldi method, Lanczos method and the proposed method were applied to the 4298×4298 matrix which was downloaded from the following URL: `http://www.cs.toronto.edu/~tsap/experiments/download/download.html`. This matrix was made by converting the directed graph to the adjacency matrix. Eigenvalues of this matrix are dense, and the computation time of the power method is time consuming. There was no significant difference between the Lanczos method and the Arnoldi method, but when the value of c was large, the proposed Lanczos method converged faster than other methods.

4.2 Example 2

The Power method, the Arnoldi method, the Lanczos method and the proposed method were applied to the 15000×15000 E-mail Enron matrix which was downloaded from the following URL: `http://snap.stanford.edu/data/index.html`. This matrix was made by converting the directed graph to the adjacency matrix. This directed graph refers to e-mail server links. The numerical results are tabulated in Table 3. With the exception of when the Matrix size is larger than the previous one and when the value of c is large, convergence requires significantly more time. Otherwise, the proposed method is not significantly influenced by the value of c, and its convergence speed is faster.

5 Conclusions

In this paper, we proposed a new algorithm to compute PageRank, using a combination of the Lanczos method and SVD. Computation times were dependent on the number of the tridiagonal matrix's degree. Our numerical results showed that computation times were constant. The proposed method has advantages which are not critically dependent on α. According to our numerical experiments, the power method requires more time than other methods, so it is not a practical choice for computing PageRank. On the other hand, we must note that the Arnoldi method with SVD satisfies the stopping criterion faster than the power method. In terms of speed, however, our proposed Lanczos method was faster than the Arnoldi method.

References

[1] Arnoldi, W. E., "*The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem,*" Quarterly of Applied Mathematics 1951, Vol. 9, pp. 17–29.

[2] Bryan, K, and Leise, T., "*The $25,000,000,000 Eigenvector: the Linear Algebra Behind Google,*" SIAM Rev., Vol. 48, 2006, pp. 569–581. http://dx.doi.org/http://dx.doi.org/10.1137/050623280

[3] Day, D., "*An Efficient Implementation of the Nonsymmetric Lanczos Algorithm,*" SIAM J. Matrix Anal. Appl, Vol. 18, pp. 566–589, 1997. http://dx.doi.org/10.1137/S0895479895292503

[4] Elden, L. A., "*A Note on the Eigenvalues of the Google Matrix,*" Report LiTH-MAT-R-04-01, Linkoping University, 2003. http://arxiv.org/abs/math/0401177

[5] Golub, G. H. and Greif, C., "*An Arnoldi-type Algorithm for Computing PageRank,*" BIT Numerical Mathematics, Vol. 46, pp. 759–771, 2006. http://dx.doi.org/10.1007/s10543-006-0091-y

[6] Gang Wu., and Yimin Wei., "*A Power Arnoldi Algorithm for Computing PageRank,*" Numer. Linear Algebra Appl., Vol. 14, pp. 521–546, 2007. http://dx.doi.org/10.1002/nla.531

[7] Jung-Feng Yin, Guo-Jan Yin, and Ng, M., "*On Adaptively Accelerated Arnoldi Method for Computing PageRank,*" Numer. Linear Algebra Appl., Vol. 19. pp. 73–85, 2012. http://dx.doi.org/10.1002/nla.789

[8] Kamvar, S. D., Haveliwala, T. H., Manning, C. D. and Golub, G. H., "*Exploiting the Block Structure of the Web for Computing PageRank,*" Technical Report, SCCM-03-02, Stanford University, 2003. http://ilpubs.stanford.edu:8090/579/

[9] Langville, A., and Meyer, C., "*Google's PageRank and Beyond: The Science of Search Engine Ranking,*" Princeton University Press, 2006. http://press.princeton.edu/titles/8216.html

[10] Page, L., Brin, S., Motwani, R., and Winograd, T., "*The PageRank Citation Ranking: Bringing Order to the Web,*" Stanford Digital Libraries, 1999, http://dbpubs.stanford.edu:8090/pub/1999-66.

[11] Saad, Y., "*Iterative Methods for Sparse Linear Systems.*" SIAM, Second Edition, 2003. http://dx.doi.org/10.1137/1.9780898718003

[12] Saberi, N. H. and Ghazvini, H., "*Weighted Restarting Method in the Weighted Arnoldi Algorithm for Computing the Eigenvalues of a Nonsymmetric Matrix,*" Applied Mathematics and Computation, Vol. 175, pp. 1276–1287, 2006.
`http://dx.doi.org/10.1016/j.amc.2005.08.035`

[13] Teramoto, K. and Nodera, T., "*A Note on the Lanczos Algorithm for Computing PageRank,*" Computational Mathematics, Proceedings on Computational Geometry and Statistics, pp. 12–15, 2014.
`http://dx.doi.org/10.5176/2251-1911_CMCGS14.15`

Author addresses

1. **Kazuma Teramoto**, School of Fundamental Science and Technology, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan.
`mailto:trmt@keio.jp`

2. **Takashi Nodera**, Department of mathematics, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, Japan.
`mailto:nodara@math.keio.ac.jp`

Department of Mathematics
Faculty of Science and Technology
Keio University

Research Report

2014

[14/001] A. Larraín-Hubach, Y. Maeda, S. Rosenberg, F. Torres-Ardila,
Equivariant, strong and leading order characteristic classes associated to fibrations,
KSTS/RR-14/001, January 6, 2014

[14/002] Ryoichi Suzuki,
A Clark-Ocone type formula under change of measure for canonical Lévy processes,
KSTS/RR-14/002, March 12, 2014

2015

[15/001] Shiro Ishikawa,
Linguistic interpretation of quantum mechanics: Quantum Language,
KSTS/RR-15/001, January 22, 2015

[15/002] Takuji Arai, Ryoichi Suzuki
Local risk-minimization for Lévy markets,
KSTS/RR-15/002, February 2, 2015

[15/003] Kazuma Teramoto, Takashi Nodera
Lanczos type method for computing PageRank,
KSTS/RR-15/003, March 9, 2015

[15/004] Yoichi Matsuo, Takashi Nodera
Block symplectic Gram-Schmidt method,
KSTS/RR-15/004, March 9, 2015

[15/005] Yuto Yokota, Takashi Nodera
The L-BFGS method for nonlinear GMRES acceleration,
KSTS/RR-15/005, March 9, 2015