SIAM J. Matrix Anal. Appl.
Vol. 28, No. 2, pp. 551–575

THE PAGERANK VECTOR: PROPERTIES, COMPUTATION, APPROXIMATION, AND ACCELERATION[*]

CLAUDE BREZINSKI[†] AND MICHELA REDIVO-ZAGLIA[‡]

This work is dedicated to the memory of Prof. Germund Dahlquist

Abstract. An important problem in Web search is determining the importance of each page. After introducing the main characteristics of this problem, we will see that, from the mathematical point of view, it could be solved by computing the left principal eigenvector (the PageRank vector) of a matrix related to the structure of the Web by using the power method. We will give expressions of the PageRank vector and study the mathematical properties of the power method. Various Padé-style approximations of the PageRank vector will be given. Since the convergence of the power method is slow, it has to be accelerated. This problem will be discussed. Recently, several acceleration methods were proposed. We will give a theoretical justification for these methods. In particular, we will generalize the recently proposed Quadratic Extrapolation, and we interpret it on the basis of the method of moments of Vorobyev, and as a Krylov subspace method. Acceleration results are given for the various ϵ-algorithms, and for the E-algorithm. Other algorithms for this problem are also discussed.

Key words. PageRank, Web matrix, eigenvector computation, power method, Padé approximation, convergence acceleration

AMS subject classifications. 65F15, 65B99, 68U35

DOI. 10.1137/050626612

1. Introduction.

An important problem in Web search is classifying the pages according to their importance. In section 2, we formulate and discuss this problem in mathematical terms and explain how a rank is assigned to each page for creating the so-called PageRank vector. Various expressions of this vector are given in section 3. Since the PageRank vector is the dominant eigenvector of a stochastic and irreducible matrix, it can be computed by the power method, whose iterates are analyzed in section 4. The results of these two sections will justify the choices made for approximating of the PageRank vector and for accelerating the power method. Section 5 is devoted to the construction of Padé-style rational approximations of the PageRank vector. In section 6, we first present some general ideas on the acceleration of vector sequences by extrapolation. Procedures based on vector least squares extrapolation are discussed. Then, using this framework, we consider several algorithms which were recently proposed for accelerating the convergence of the power method [31]. Their effectiveness is theoretically justified. One of them is connected to Krylov subspace methods, and to the method of moments of Vorobyev [50, 8]. The application of the various ϵ-algorithms and of the E-algorithm to the PageRank problem is studied, and convergence acceleration results are proved. Finally, other possible acceleration techniques are considered.

[*] Received by the editors March 11, 2005; accepted for publication (in revised form) by V. Simoncini February 7, 2006; published electronically July 31, 2006.
http://www.siam.org/journals/simax/28-2/62661.html

[†] Laboratoire Paul Painlevé, UMR CNRS 8524, UFR de Mathématiques Pures et Appliquées, Université des Sciences et Technologies de Lille, France (Claude.Brezinski@univ-lille1.fr). The work of this author was partially supported by the project INTAS 03-51-6637.

[‡] Dipartimento di Matematica Pura ed Applicata, Università degli Studi di Padova, Italy (Michela.RedivoZaglia@unipd.it). The work of this author was supported by MIUR grant 2004015437.

Let us also mention that other classes of acceleration techniques, such as aggregation/disaggregation [30, 35, 25], lumping [37], adaptive methods [29], and the parallel computation of the PageRank vector [22, 32] are not discussed herein.

2. The problem. A query to a Web search engine often produces a very long list of answers because of the enormous number of pages (over 8 billion in Google's database). To help the surfer, these pages have to be listed starting from the most relevant ones. Google uses several metrics and strategies for solving this *ranking* problem.

The importance of a page is called its *PageRank*, and the *PageRank algorithm* [18, 41] is reportedly one of the main ingredients of Google's link analysis. A page is considered to be important if many other important pages are pointing to it. So, the importance of a page is determined by the importance of the other pages. This means that the row vector $\mathbf{r}^T$ of all PageRanks is only defined implicitly as the solution of a fixed-point problem, as we will see now.

Let $\deg(i) \geq 1$ be the outdegree (that is, the number of pages it points to) of the page i. Let $P = (p_{ij})$ be the matrix which describes the transitions between the pages i and j, where $p_{ij} = 1/\deg(i)$, $p_{ij} = 0$ if there is no outlink from page i to page j, and $p_{ii} = 0$.

The PageRank vector $\mathbf{r}^T$ satisfies $\mathbf{r}^T = \mathbf{r}^T P$, that is, $\mathbf{r} = P^T\mathbf{r}$, and it can be computed recursively by the standard power method

$$\mathbf{r}^{(n+1)} = P^T\mathbf{r}^{(n)}, \quad n = 0, 1, \ldots,$$

assuming that $\mathbf{r}$ is present in the spectral decomposition of $\mathbf{r}^{(0)}$. Unfortunately, this iterative procedure has convergence problems. It can cycle, or the limit can depend on the starting vector $\mathbf{r}^{(0)}$ [33].

To avoid these drawbacks, the original PageRank algorithm was revised.

First, since some pages have no outlink (*dangling* pages), P is not stochastic (some of its rows are zero). Different strategies were proposed to remedy this problem, but the most used one is to replace P by another matrix $\widetilde{P}$ as follows. Let $\mathbf{w} = (w_1, \ldots, w_p)^T \in \mathbb{R}^p$ be a probability vector, that is, such that $\mathbf{w} \geq 0$ and $\mathbf{e}^T\mathbf{w} = 1$ with $\mathbf{e} = (1, \ldots, 1)^T$, and p is the total number of pages. Let $\mathbf{d} = (d_i) \in \mathbb{R}^p$ be the vector with $d_i = 1$ if $\deg(i) = 0$, and 0 otherwise. We set

$$\widetilde{P} = P + \mathbf{d}\mathbf{w}^T.$$

The effect of the additional matrix $\mathbf{d}\mathbf{w}^T$ is to modify the probabilities so that a surfer visiting a page without outlinks jumps to another page with the probability distribution defined by $\mathbf{w}$. This matrix $\widetilde{P}$ is stochastic, and thus it has 1 as its dominant eigenvalue, with $\mathbf{e}$ as its corresponding right eigenvector. So $I - \widetilde{P}$ is singular.

Another problem arises since $\widetilde{P}$ is reducible. In that case, $\widetilde{P}$ can have several eigenvalues on the unit circle, thus causing convergence problems to the power method. Moreover, $\widetilde{P}$ can have several left eigenvectors corresponding to its dominant eigenvalue 1 (see [3, 47, 49] for a general discussion, and [17] or [44] for the particular case of the PageRank problem). Then $\widetilde{P}$ itself is finally replaced by the matrix

$$P_c = c\widetilde{P} + (1-c)E, \qquad E = \mathbf{e}\mathbf{v}^T,$$

with $c \in [0, 1]$ and $\mathbf{v}$ a probability vector. It corresponds to adding to all pages a new set of outgoing transitions with small probabilities. The probability distribution given by the vector $\mathbf{v}$ can differ from a uniformly distributed vector, and the resultant PageRank can be biased to give preference to certain kinds of pages. For that reason, $\mathbf{v}^T$ is called the *personalization* vector. The matrix P_c is nonnegative, stochastic, and now irreducible since $\mathbf{v}$ is a positive vector. It has only one eigenvalue on the unit circle. This eigenvalue is equal to 1, and $\mathbf{e}$ is its corresponding right eigenvector [3, 40, 47, 49]. Indeed

$$P_c\mathbf{e} = c\widetilde{P}\mathbf{e} + (1-c)\mathbf{e}\mathbf{v}^T\mathbf{e} = c\mathbf{e} + (1-c)\mathbf{e} = \mathbf{e}.$$

Thus, the matrix $I - P_c$ is singular. The power iterations for the matrix P_c^T now converge to a unique vector $\mathbf{r}_c$ (obviously, depending on c), which is chosen as the PageRank vector. Let us mention that P is extremely sparse, while P_c is completely dense. However, the power method could be implemented only with sparse matrix-vector multiplications, and without even storing P_c as described in section 4. As will be seen below, the vector $\mathbf{r}_c$ can also be computed as the solution of a system of linear equations.

The PageRank problem is closely related to Markov chains [34]. For properties of stochastic matrices, we refer the interested reader to [40] and [47]. For nonnegative matrices, see [3] and [49].

We are finally faced with the following mathematical problem. We set $A_c = P_c^T$. The $p \times p$ matrix A_c has eigenvalues $|c\widetilde{\lambda}_p| \leq \cdots \leq |c\widetilde{\lambda}_2| < \widetilde{\lambda}_1 = 1$, where the $\widetilde{\lambda}_i$ are the eigenvalues of $\widetilde{P}$, and we have to compute $\mathbf{r}_c$, its unique right eigenvector corresponding to the eigenvalue $\widetilde{\lambda}_1 = 1$ [20, 34]. For that purpose, we can use the power method, which consists in the iterations

$$\mathbf{r}_c^{(n+1)} = A_c\mathbf{r}_c^{(n)}, \qquad n = 0, 1, \ldots, \tag{1}$$

with $\mathbf{r}_c^{(0)}$ given.

The vector $\mathbf{r}_c^{(0)}$ is the probability distribution over the surfer's location at step time 0, and $\mathbf{r}_c^{(n)}$ is its probability distribution at time n. The unique stationary distribution vector of the Markov chain characterized by A_c is the limit of the sequence $(\mathbf{r}_c^{(n)})$, which always exists since A_c is primitive and irreducible, and it is independent of $\mathbf{r}_c^{(0)}$. This limit is the right eigenvector $\mathbf{r}_c$ of the matrix A_c corresponding to its dominant eigenvalue 1, and it is exactly the vector that we would like to compute [40, p. 691].

The sequence $(\mathbf{r}_c^{(n)})$ given by (1) always converges to $\mathbf{r}_c$, but if $c \simeq 1$, the convergence is slow since the power method converges as c^n (see [34], and Property 12 below). So, a balance has to be found between a small value of c, which insures a fast convergence of $(\mathbf{r}_c^{(n)})$, but to a vector $\mathbf{r}_c$ which is not close to the real PageRank vector $\widetilde{\mathbf{r}} = \lim_{c\to 1}\mathbf{r}_c$, and a value of c close to 1, which leads to a better approximation $\mathbf{r}_c$ of $\widetilde{\mathbf{r}}$, but with a slow convergence. Originally, Google chose $c = 0.85$, which insures a good rate of convergence [18].

However, computing a PageRank vector can take several days, and so convergence acceleration is essential, in particular, for providing continuous updates to ranking. Moreover, some recent approaches require the computation of several PageRank vectors corresponding to different personalization vectors. Recently, several methods for accelerating the computation of the PageRank vector by the power method were proposed [31, 29]. In this paper we will provide a theoretical justification of the methods of [31], and we will put them on a firm theoretical basis. Other convergence acceleration procedures will also be proposed and discussed. In order to be able to prove that these algorithms accelerate the convergence of the power method, they have to be strongly supported by theoretical results. This is what will be achieved in this paper. It is not our purpose here to test these algorithms numerically, nor to compare them with other possible procedures for obtaining the PageRank vector.

For a detailed exposition of the PageRank problem, see [34] and [36]. Other reviews are [26] and [2].

3. The PageRank vector.

Since P_c is stochastic and irreducible, $\mathbf{r}_c$ is the unique right eigenvector of $A_c = P_c^T$ corresponding to the simple eigenvalue 1, that is, $A_c\mathbf{r}_c = \mathbf{r}_c$. By the Perron–Frobenius theorem (see, for example, [49, p. 35]), $\mathbf{r}_c \geq 0$. It is normalized so that $\mathbf{e}^T\mathbf{r}_c = 1$, and, thus, it is a probability vector.

In this section, we will study the properties of this vector, and, in particular, we will give implicit and explicit expressions for it. Then we will discuss its computation by the power method. This discussion will lead us, in the next two sections, to various procedures for its approximation, and to processes for accelerating the convergence of the power method.

3.1. Implicit expressions for the PageRank vector.

Let us give implicit expressions for $\mathbf{r}_c$.

Setting $\widetilde{A} = \widetilde{P}^T$, we have

$$\begin{aligned} A_c\mathbf{r}_c &= c\widetilde{A}\mathbf{r}_c + (1-c)\mathbf{v}\mathbf{e}^T\mathbf{r}_c \\ &= c\widetilde{A}\mathbf{r}_c + (1-c)\mathbf{v} \\ &= \mathbf{r}_c. \end{aligned}$$

Thus, $(I - c\widetilde{A})\mathbf{r}_c = (1-c)\mathbf{v}$, that is, we have the following.

PROPERTY 1.

$$\begin{aligned} \mathbf{r}_c &= (1-c)(I - c\widetilde{A})^{-1}\mathbf{v} \\ &= \mathbf{v} + c(\widetilde{A} - I)(I - c\widetilde{A})^{-1}\mathbf{v}. \end{aligned}$$

The second expression is deduced from the first one by noticing that $(I - c\widetilde{A})^{-1} = I + c\widetilde{A}(I - c\widetilde{A})^{-1}$.

Following Property 1, $\mathbf{r}_c$ can be obtained as the solution of the dense system of linear equations $(I - c\widetilde{A})\mathbf{r}_c = (1-c)\mathbf{v}$. Replacing A_c by its expression leads to $(I - cP^T - c\mathbf{w}\mathbf{d}^T)\mathbf{r}_c = (1-c)\mathbf{v}$. But $\mathbf{e}^T\mathbf{w}\mathbf{d}^T = \mathbf{e}^T\widetilde{A} - \mathbf{e}^TP^T$. Thus, since $\mathbf{e}^T\mathbf{w} = 1$ and $\mathbf{e}^T = \mathbf{e}^T\widetilde{A}$, we have $\mathbf{d}^T = \mathbf{e}^T - \mathbf{e}^TP^T$, and, when $\mathbf{w} = \mathbf{v}$, we finally obtain the sparse system $(I - cP^T)\mathbf{r}_c = \gamma\mathbf{v}$, where $\gamma = \|\mathbf{r}_c\|_1 - c\|P^T\mathbf{r}_c\|_1$ [22]. A particular choice of γ only results in a rescaling of the solution of this system, and it can always be chosen so that $\mathbf{r}_c$ is a probability vector. Various iterative methods for the solution of this system are discussed in many papers, including [1, 4, 19].

From Property 1, we immediately obtain the following.

PROPERTY 2.

$$\begin{aligned} \mathbf{r}_c &= (1-c)\sum_{i=0}^{\infty} c^i\widetilde{A}^i\mathbf{v} \\ &= \mathbf{v} + c(\widetilde{A} - I)\sum_{i=0}^{\infty} c^i\widetilde{A}^i\mathbf{v}. \end{aligned}$$

These results were proved in [5]. These series are convergent since $\rho(\widetilde{A}) = 1$ and $0 \leq c < 1$. Since $\mathbf{r}_c$ can be expressed as a power series, it will allow us to construct rational approximations of it; see section 5.

Remark 1. It is easy to check from the result of Property 2 that $\mathbf{e}^T\mathbf{r}_c = 1$. Indeed $\mathbf{e}^T\widetilde{A}^i = \mathbf{e}^T$, and thus

$$\mathbf{e}^T\mathbf{r}_c = (1-c)\sum_{i=0}^{\infty} c^i\mathbf{e}^T\widetilde{A}^i\mathbf{v} = (1-c)\sum_{i=0}^{\infty} c^i\mathbf{e}^T\mathbf{v} = (1-c)\sum_{i=0}^{\infty} c^i,$$

since $\mathbf{e}^T\mathbf{v} = 1$. But $\sum_{i=0}^{\infty} c^i = (1-c)^{-1}$, which shows the result.

3.2. Explicit expressions for the PageRank vector.

Let us now give explicit forms for $\mathbf{r}_c$. We will first express it as a rational function, and then propose a polynomial form.

3.2.1. Rational expressions.

We assume that $\widetilde{P}$ is diagonalizable. Thus, $\widetilde{P} = XDX^{-1}$, where $D = \mathrm{diag}(1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_p)$, and where $X = [\mathbf{e}, \mathbf{x}_2, \ldots, \mathbf{x}_p]$ is the matrix whose columns are the right eigenvectors of $\widetilde{P}$. Also, let $Y = [\widetilde{\mathbf{r}}, \mathbf{y}_2, \ldots, \mathbf{y}_p]$ be the matrix whose columns are the right eigenvectors of $\widetilde{P}^T$, that is, $\widetilde{A}$. We have $X^{-T} = Y$ and

$$(I - c\widetilde{A})^{-1} = X^{-T}(I - cD)^{-1}X^T.$$

But

$$(I-cD)^{-1} = \begin{pmatrix} (1-c)^{-1} & & & \\ & (1-c\widetilde{\lambda}_2)^{-1} & & \\ & & \ddots & \\ & & & (1-c\widetilde{\lambda}_p)^{-1} \end{pmatrix}, \quad X^T\mathbf{v} = \begin{pmatrix} 1 \\ \mathbf{x}_2^T\mathbf{v} \\ \vdots \\ \mathbf{x}_p^T\mathbf{v} \end{pmatrix},$$

and it follows that

$$\mathbf{u} = (I-cD)^{-1}X^T\mathbf{v} = \begin{pmatrix} 1/(1-c) \\ \mathbf{x}_2^T\mathbf{v}/(1-c\widetilde{\lambda}_2) \\ \vdots \\ \mathbf{x}_p^T\mathbf{v}/(1-c\widetilde{\lambda}_p) \end{pmatrix}.$$

So, we finally obtain $\mathbf{r}_c = (1-c)X^{-T}\mathbf{u} = (1-c)Y\mathbf{u}$. This result was given in [44], where a similar proof when $\widetilde{P}$ is not diagonalizable could also be found, thus leading to the following result.

PROPERTY 3. *If $\widetilde{P}$ is diagonalizable,*

$$\mathbf{r}_c = \widetilde{\mathbf{r}} + (1-c)\sum_{i=2}^{p} \frac{\alpha_i}{1-c\widetilde{\lambda}_i}\mathbf{y}_i,$$

with $\alpha_i = \mathbf{x}_i^T\mathbf{v}$.

In the general case

$$\mathbf{r}_c = \widetilde{\mathbf{r}} + \sum_{i=2}^{p} w_i(c)\,\mathbf{y}_i,$$

with

$$\begin{aligned} w_2(c) &= (1-c)\alpha_2/(1-c\widetilde{\lambda}_2), \\ w_i(c) &= [(1-c)\alpha_i + c\beta_i w_{i-1}(c)]/(1-c\widetilde{\lambda}_i), \quad i = 3, \ldots, p, \end{aligned}$$

and β_i equal to 0 or 1.

It follows from this result that since $\mathbf{r}_c$ is a rational function without poles at $c = 1$, there exists a unique vector which is the limit, when c tends to 1, of $\mathbf{r}_c$. This vector is only one of the nonnegative normalized dominant left eigenvectors of $\widetilde{P}$, and it will be chosen as the real PageRank vector. This vector depends on $\mathbf{v}$, a natural property since $\widetilde{P}$ depends on the personalization vector, and also on the multiplicity of the eigenvalue 1 as explained in [17]. Let us mention that, as proved in [20, 34], the eigenvalues of the matrix P_c are $c\widetilde{\lambda}_i$ for $i \geq 2$; see also [24], where it is stated that $\widetilde{\lambda}_2 = 1$ in the special case of the Google matrix.

Let us now give another expression for $\mathbf{r}_c$. It comes from the well-known expression for the resolvent of a matrix (see, for example, [9, pp. 19–20]). We have

$$(I - c\widetilde{A})^{-1} = \frac{1}{\det(I - c\widetilde{A})}(I + cB_1 + \cdots + c^{p-1}B_{p-1}),$$

where the matrices B_i are given by the Le Verrier–Faddeev–Souriau algorithm

$$\begin{aligned} B_1 &= \widetilde{A} + \gamma_1 I, & \gamma_1 &= -\mathrm{tr}(\widetilde{A}), \\ B_i &= \widetilde{A}B_{i-1} + \gamma_i I, & \gamma_i &= -\frac{1}{i}\,\mathrm{tr}(\widetilde{A}B_{i-1}), \qquad i = 2, \ldots, p, \end{aligned}$$

where "tr" designates the trace of a matrix. Moreover

$$\begin{aligned} \det(I - c\widetilde{A}) &= 1 + \gamma_1 c + \cdots + \gamma_p c^p, \\ B_p &= 0, \\ \widetilde{A}^{-1} &= -(1/\gamma_p)B_{p-1}. \end{aligned}$$

Thus, it follows from Property 1 that

$$\mathbf{r}_c = \frac{1 - c}{1 + \gamma_1 c + \cdots + \gamma_p c^p}(I + cB_1 + \cdots + c^{p-1}B_{p-1})\mathbf{v}.$$

This result shows that $\mathbf{r}_c$ is a rational function with a vector numerator of degree p at most, and a scalar denominator of degree p, while in Property 3 both degrees were at most $p - 1$. Let us conciliate these two results.

Since $\widetilde{A}$ has an eigenvalue equal to 1, then $1 + \gamma_1 + \cdots + \gamma_p = 0$, and it follows that

$$\begin{aligned} 1 + \gamma_1 c + \cdots + \gamma_p c^p &= -\gamma_1 - \cdots - \gamma_p + \gamma_1 c + \cdots + \gamma_p c^p \\ &= \gamma_1(c - 1) + \gamma_2(c^2 - 1) + \cdots + \gamma_p(c^p - 1). \end{aligned}$$

Thus, after cancellation of $c - 1$ in the numerator and in the denominator, we obtain the following.

PROPERTY 4.

$$\mathbf{r}_c = -\frac{(I + cB_1 + \cdots + c^{p-1}B_{p-1})\mathbf{v}}{\gamma_1 + \gamma_2(1 + c)\cdots + \gamma_p(1 + \cdots + c^{p-1})}.$$

Remark 2. In this expression of $\mathbf{r}_c$, the denominator can also be written as $\beta_0 + \cdots + \beta_{p-1}c^{p-1}$ with $\beta_i = \gamma_{i+1} + \cdots + \gamma_p$ for $i = 0, \ldots, p - 1$.

Notice that if the minimal polynomial of A_c for the vector $\mathbf{v}$ has degree $m < p$, then cancellation occurs between the scalar denominator polynomial and the matrix numerator polynomial, thus reducing $\mathbf{r}_c$ to a rational function of type $(m - 1, m - 1)$ [21, pp. 87–94].

Since, by Properties 3 and 4, $\mathbf{r}_c$ is a rational function in the variable c, it is justifiable to approximate it by a rational function with lower degrees, as proposed in section 5.

3.2.2. Polynomial form. We will now give a polynomial expression for $\mathbf{r}_c$. Let $\Pi_m(\lambda) = a_0 + a_1\lambda + \cdots + a_m\lambda^m$ be the minimal polynomial of A_c for the vector $\mathbf{v}$ with $m \leq p$. Since A_c has a unique eigenvalue equal to 1, Π_m can be written as $\Pi_m(\lambda) = (\lambda - 1)Q_{m-1}(\lambda)$. So

$$\Pi_m(A_c)\mathbf{v} = (A_c - I)Q_{m-1}(A_c)\mathbf{v} = A_c Q_{m-1}(A_c)\mathbf{v} - Q_{m-1}(A_c)\mathbf{v} = 0.$$

Thus, $Q_{m-1}(A_c)\mathbf{v}$ is the eigenvector of A_c corresponding to the eigenvalue 1, that is, we have the following.

PROPERTY 5.

$$\mathbf{r}_c = Q_{m-1}(A_c)\mathbf{v}.$$

If we set $Q_{m-1}(\lambda) = b_0 + \cdots + b_{m-1}\lambda^{m-1}$, then $b_i = -(a_0 + \cdots + a_i) = a_{i+1} + \cdots + a_m$ for $i = 0, \ldots, m-1$ (compare with Remark 2).

Property 5, shows that approximating Q_{m-1} in some sense will lead to approximations of the vector $\mathbf{r}_c$. Such procedures will be described in section 6.

4. Computation of the PageRank vector.

The PageRank vector $\mathbf{r}_c$ can be computed by the power method starting from any nonzero vector such that $\mathbf{e}^T\mathbf{r}_c^{(0)} = 1$. We will start it from $\mathbf{v}$, a choice justified by Property 5, and by Property 7 given below:

$$\begin{aligned}\mathbf{r}_c^{(0)} &= \mathbf{v},\\ \mathbf{r}_c^{(n+1)} &= A_c\mathbf{r}_c^{(n)}, \quad n = 0, 1, \ldots.\end{aligned}$$

Obviously, for all $n, \mathbf{r}_c^{(n)} \geq 0$. Moreover, $\mathbf{e}^T\mathbf{r}_c^{(0)} = 1$. So, by induction, $\mathbf{e}^T\mathbf{r}_c^{(n+1)} = \mathbf{e}^T A_c\mathbf{r}_c^{(n)} = (P_c\mathbf{e})^T\mathbf{r}_c^{(n)} = \mathbf{e}^T\mathbf{r}_c^{(n)}$. Thus, we have the following.

PROPERTY 6.

$$\mathbf{r}_c^{(n)} = A_c^n\mathbf{v} \geq 0, \quad \textit{and} \quad \|\mathbf{r}_c^{(n)}\|_1 = \mathbf{e}^T\mathbf{r}_c^{(n)} = 1, \quad n = 0, 1, \ldots.$$

Substituting A_c by its expression, an iterate of the power method can be written as

$$\mathbf{r}_c^{(n+1)} = cP^T\mathbf{r}_c^{(n)} + c(\mathbf{d}^T\mathbf{r}_c^{(n)})\mathbf{w} + (1-c)\mathbf{v}.$$

So, an iteration costs only one matrix–vector product by the very sparse matrix P^T. Moreover, neither A_c nor $\widetilde{A}$ has to be stored. In addition, the vector $\mathbf{d}$ can be eliminated, thus making the power method easy and cheap to implement. Since, as seen above, $\mathbf{d}^T = \mathbf{e}^T - \mathbf{e}^T P^T$, then, for any vector $\mathbf{x}$, it holds, after replacing A_c, $\widetilde{A}$, and $\mathbf{d}^T$ by their expressions, that

$$A_c\mathbf{x} = cP^T\mathbf{x} + (c\|\mathbf{x}\|_1 - \|cP^T\mathbf{x}\|_1)\mathbf{w} + (1-c)\|\mathbf{x}\|_1\mathbf{v}.$$

If $\mathbf{w} = \mathbf{v}$, the formula given in [31, Alg. 1] for computing such matrix–vector products is recovered. In the particular case of the power method, $\mathbf{x} = \mathbf{r}_c^{(n)}$, $\|\mathbf{r}_c^{(n)}\|_1 = 1$, and the above formula simplifies to

$$\mathbf{r}_c^{(n+1)} = A_c\mathbf{r}_c^{(n)} = cP^T\mathbf{r}_c^{(n)} + (c - \|cP^T\mathbf{r}_c^{(n)}\|_1)\mathbf{w} + (1-c)\mathbf{v}.$$

Only one vector has to be stored by iteration. See [34] for details about the operational count.

As will be seen in Property 14, it follows from Property 6 that the vectors $\mathbf{r}_c^{(n)} - \mathbf{r}_c$ satisfy a difference equation, a result that will be used for proving that the ϵ-algorithms accelerate the convergence of the power method (see section 6.3).

As proved in [5], an important property is that the vectors $\mathbf{r}_c^{(n)}$ computed by the power method are the partial sums of the second series for $\mathbf{r}_c$ given in Property 2. Let us give a simpler proof of this result.

Property 7.

$$\begin{aligned}\mathbf{r}_c^{(n)} &= (1-c)\sum_{i=0}^{n-1} c^i \widetilde{A}^i \mathbf{v} + c^n \widetilde{A}^n \mathbf{v}, \quad n \geq 0,\\ &= \mathbf{v} + c(\widetilde{A} - I)\sum_{i=0}^{n-1} c^i \widetilde{A}^i \mathbf{v}.\end{aligned}$$

Proof. Let us prove the second identity. For $n = 0$, the sum is zero and the result is true. For $n = 1$, we have

$$\mathbf{r}_c^{(1)} = c\widetilde{A}\mathbf{r}_c^{(0)} + (1-c)\mathbf{v}\mathbf{e}^T\mathbf{r}_c^{(0)} = \mathbf{v} + (\widetilde{A} - I)c\mathbf{v}.$$

Assuming that the result holds for n, we have

$$\begin{aligned}\mathbf{r}_c^{(n+1)} &= [c\widetilde{A} + (1-c)\mathbf{v}\mathbf{e}^T]\mathbf{r}_c^{(n)}\\ &= c\widetilde{A}\mathbf{r}_c^{(n)} + (1-c)\mathbf{v} \quad \text{by Property 6}\\ &= c\widetilde{A}\mathbf{v} + c^2\widetilde{A}(\widetilde{A} - I)\sum_{i=0}^{n-1} c^i \widetilde{A}^i \mathbf{v} + (1-c)\mathbf{v}\\ &= c\widetilde{A}\mathbf{v} + c(\widetilde{A} - I)\sum_{i=1}^{n} c^i \widetilde{A}^i \mathbf{v} + (1-c)\mathbf{v}\\ &= \mathbf{v} + c(\widetilde{A} - I)\sum_{i=0}^{n} c^i \widetilde{A}^i \mathbf{v}.\end{aligned}$$

The first result can be easily obtained from the second one. □

Since the power method furnishes the partial sums of the power series for $\mathbf{r}_c$, its iterates will be directly used for constructing Padé-type approximants of this vector; see section 5.

Property 8 immediately follows from Property 7.

Property 8.

$$\begin{aligned}\mathbf{r}_c^{(0)} &= \mathbf{v},\\ \mathbf{r}_c^{(n+1)} &= \mathbf{r}_c^{(n)} + c^{n+1}(\widetilde{A} - I)\widetilde{A}^n \mathbf{v}, \quad n = 0, 1, \ldots .\end{aligned}$$

Moreover, the following holds.

Property 9.

$$(\widetilde{A} - I)\widetilde{A}^n \mathbf{v} = \frac{1}{c^{n+1}}(\mathbf{r}_c^{(n+1)} - \mathbf{r}_c^{(n)}), \quad n = 0, 1, \ldots .$$

This property, proved in [5], shows that it is possible to apply the power method simultaneously for several values of c with only a small additional cost. Indeed, by Property 9, one only has to compute the vectors $(\widetilde{A} - I)\widetilde{A}^n\mathbf{v}$ once, and then use

Property 8 for computing the partial sums $\mathbf{r}_{\widetilde{c}}^{(n)}$ of the series $\mathbf{r}_{\widetilde{c}}$ for a different value $\widetilde{c}$ of c. So, we have

$$\begin{aligned}\mathbf{r}_{\widetilde{c}}^{(0)} &= \mathbf{v},\\ \mathbf{r}_{\widetilde{c}}^{(n+1)} &= \mathbf{r}_{\widetilde{c}}^{(n)} + \widetilde{c}^{n+1}\frac{1}{c^{n+1}}(\mathbf{r}_c^{(n+1)} - \mathbf{r}_c^{(n)}), \quad n = 0, 1, \ldots.\end{aligned}$$

Since $I + c\widetilde{A} + \cdots + c^{n-1}\widetilde{A}^{n-1} = (I - c\widetilde{A})^{-1}(I - c^n\widetilde{A}^n)$, the results of Property 7 can also be written as follows for comparison with Property 1.

Property 10.

$$\begin{aligned}\mathbf{r}_c^{(n)} &= (1-c)(I - c\widetilde{A})^{-1}(I - c^n\widetilde{A}^n)\mathbf{v} + c^n\widetilde{A}^n\mathbf{v}\\ &= \mathbf{v} + c(\widetilde{A} - I)(I - c\widetilde{A})^{-1}(I - c^n\widetilde{A}^n)\mathbf{v}.\end{aligned}$$

Let us now give expressions for the error. From Properties 1, 6, and 10, it is easy to prove the following.

Property 11.

$$\begin{aligned}\mathbf{r}_c - \mathbf{r}_c^{(n)} &= A_c^n(\mathbf{r}_c - \mathbf{v})\\ &= c^n\widetilde{A}^n(\mathbf{r}_c - \mathbf{v})\\ &= (I - c\widetilde{A})^{-1}(\mathbf{r}_c^{(n+1)} - \mathbf{r}_c^{(n)}).\end{aligned}$$

Since $\widetilde{A}$ is a column stochastic matrix $\|\widetilde{A}\|_1 = 1$, and since, in our case, it is also reducible, then $|\widetilde{\lambda}_2| = 1$, and we obtain the following.

Property 12.

$$\begin{aligned}\|\mathbf{r}_c - \mathbf{r}_c^{(n)}\|_1 &\leq c^n \ \|\mathbf{r}_c - \mathbf{v}\|_1\\ &\leq \frac{1}{1-c}\ \|\mathbf{r}_c^{(n+1)} - \mathbf{r}_c^{(n)}\|_1.\end{aligned}$$

Let us note that $1/(1-c)$ is the 1-norm of the matrix $(I - c\widetilde{A})^{-1}$ and that the condition number of the PageRank problem is $(1+c)/(1-c)$ [28].

Let us now explain how rational and polynomial approximations of $\mathbf{r}_c$ could be obtained from the iterates of the power method. In both cases, increasing the degree of the approximation produces a sequence of approximations of $\mathbf{r}_c$ of increasing order which, under certain assumptions, converge to $\mathbf{r}_c$ faster than the iterates of the power method.

5. Padé approximation of the PageRank vector.

As proved in Properties 3 and 4, $\mathbf{r}_c$ is a vector rational function of type $(p-1, p-1)$ (or $(m-1, m-1)$, where m is the degree of the minimal polynomial of A_c for the vector $\mathbf{v}$) in the variable c, that is, a rational function with a numerator of degree $p-1$ (or $m-1$) with vector coefficients, and a common scalar denominator of degree $p-1$ (or $m-1$). Moreover, by Property 2, the vector Taylor series expansion of $\mathbf{r}_c$ is known. So, the partial sums of this series could be used for constructing rational approximations of $\mathbf{r}_c$ of type $(k-1, k-1)$ with $k < p$ (or $k < m$). The coefficients of these rational functions will be chosen so that their power series expansion agrees with that of $\mathbf{r}_c$ as far as possible. Such types of rational functions are called Padé approximants.

The first possibility is to construct the scalar Padé approximants $[k-1/k-1]$ separately for each component of $\mathbf{r}_c$. In that case, each component will be matched up to the term of degree $2k-2$ inclusively. However, each scalar Padé approximant could have a different denominator for each component. For more on Padé approximation, see [6, 9].

A solution that seems preferable is to use vector Padé approximants since the components of $\mathbf{r}_c$ are rational functions with a common denominator, which is exactly the characterizing property of vector Padé approximants. These approximants, introduced by Van Iseghem [48], are defined as follows.

Let $\mathbf{f}$ be a vector formal power series

$$\mathbf{f}(\xi) = \sum_{i=0}^{\infty} \boldsymbol{\sigma}_i \xi^i, \quad \boldsymbol{\sigma}_i \in \mathbb{R}^p. \tag{2}$$

We look for a vector rational function whose series expansion in ascending powers of ξ agrees with $\mathbf{f}$ as far as possible. By vector rational function, we mean a function with vector coefficients in the numerator and with a scalar denominator. More precisely, we look for $\mathbf{a}_0, \ldots, \mathbf{a}_{k-1} \in \mathbb{R}^p$, and $b_0, \ldots, b_{k-1} \in \mathbb{R}$, with $k \leq p$ (or $k \leq m$), such that

$$(b_0 + \cdots + b_{k-1}\xi^{k-1})(\boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_1 \xi + \cdots) - (\mathbf{a}_0 + \cdots + \mathbf{a}_{k-1}\xi^{k-1}) = \mathcal{O}(\xi^s), \tag{3}$$

with s, the order of approximation, as high as possible. If $s = k$, this vector rational function is called a *Padé-type approximant* of $\mathbf{f}$, while it is called a *Padé approximant* if $s = 2k - 1$.

Identifying to zero the vector coefficients of the terms of degree 0 to $k - 1$ in the left-hand side of (3), we obtain

$$\begin{aligned} \mathbf{a}_0 &= b_0 \boldsymbol{\sigma}_0, \\ \mathbf{a}_1 &= b_0 \boldsymbol{\sigma}_1 + b_1 \boldsymbol{\sigma}_0, \\ &\;\vdots \\ \mathbf{a}_{k-1} &= b_0 \boldsymbol{\sigma}_{k-1} + \cdots + b_{k-1} \boldsymbol{\sigma}_0. \end{aligned} \tag{4}$$

For any choice of the coefficients b_i of the denominator with $b_0 \neq 0$, the rational function $(\mathbf{a}_0 + \cdots + \mathbf{a}_{k-1}\xi^{k-1})/(b_0 + \cdots + b_{k-1}\xi^{k-1})$ obtained by this procedure is a vector Padé-type approximant, and it is denoted by $(k-1/k-1)_{\mathbf{f}}(\xi)$. Its order of approximation is $s = k$, that is, $(k-1/k-1)_{\mathbf{f}}(\xi) - \mathbf{f}(\xi) = \mathcal{O}(\xi^k)$. The computation of the approximant $(k-1/k-1)_{\mathbf{f}}$ needs the knowledge of $\boldsymbol{\sigma}_0, \ldots, \boldsymbol{\sigma}_{k-1}$. Thus, in practice, only small values of k could be used depending on the number of vectors one could store.

Let us now try to improve the order of approximation, that is, to construct vector Padé approximants. However, as we will see now, this order could not be improved simultaneously for all components of the approximants since k has to be smaller than p. Indeed, for eliminating the term of degree k in (3), it is necessary and sufficient that $0 = b_0 \boldsymbol{\sigma}_k + \cdots + b_{k-1}\boldsymbol{\sigma}_1$. Since a rational function is defined apart from a multiplying factor, we can set $b_0 = 1$, and we get

$$b_1 \boldsymbol{\sigma}_{k-1} + \cdots + b_{k-1}\boldsymbol{\sigma}_1 = -\boldsymbol{\sigma}_k.$$

This is a system of p equations with $k - 1 \leq p$ unknowns. It has to be solved in the least squares sense. Setting $C = [\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{k-1}] \in \mathbb{R}^{p \times (k-1)}$ and $\mathbf{b} = (b_{k-1}, \ldots, b_1)^T$, this system can be rewritten as $C\mathbf{b} = -\boldsymbol{\sigma}_k$. Let $C^\dagger$ be a left inverse of C, that is, a $(k-1) \times p$ matrix such that $C^\dagger C = I \in \mathbb{R}^{(k-1)\times(k-1)}$. Thus, $\mathbf{b} = -C^\dagger \boldsymbol{\sigma}_k$. Once the b_i's have been obtained, the $\mathbf{a}_i$'s can be directly computed by the relations (4). The matrix $C^\dagger$ has the form $C^\dagger = (Z^T C)^{-1} Z^T$, where $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_{k-1}] \in \mathbb{R}^{p \times (k-1)}$ is any matrix such that $Z^T C \in \mathbb{R}^{(k-1)\times(k-1)}$ is nonsingular. The rational approximant constructed in that way is called a vector Padé approximant of $\mathbf{f}$, although its order of approximation s is only equal to k, and it is denoted by $[k-1/k-1]_{\mathbf{f}}(\xi)$. The reason for this abuse of language is that we now have $\mathbf{z}_i^T[k-1/k-1]_{\mathbf{f}}(\xi) - \mathbf{z}_i^T\mathbf{f}(\xi) = \mathcal{O}(\xi^{2k-1})$ for $i = 1, \ldots, k-1$. Obviously, since a left inverse is not unique, different vector Padé approximants could be constructed. Of course, the simplest choice is $Z = C$. In that case, $C^\dagger$ is the pseudoinverse of C, and the corresponding Padé approximants can be computed by the Recursive Projection Algorithm (RPA) [13, sect. 4.4].

Another way of proceeding for obtaining the coefficients b_i is to consider the $k-1$ scalar equations

$$\begin{aligned} b_1(\mathbf{z}, \boldsymbol{\sigma}_{k-1}) + \cdots + b_{k-1}(\mathbf{z}, \boldsymbol{\sigma}_1) &= -(\mathbf{z}, \boldsymbol{\sigma}_k), \\ &\vdots \\ b_1(\mathbf{z}, \boldsymbol{\sigma}_{2k-3}) + \cdots + b_{k-1}(\mathbf{z}, \boldsymbol{\sigma}_{k-1}) &= -(\mathbf{z}, \boldsymbol{\sigma}_{2k-2}), \end{aligned}$$

where $\mathbf{z}$ is any vector such that the matrix of this system is nonsingular. For these approximants, we again have $[k-1/k-1]_{\mathbf{f}}(\xi) - \mathbf{f}(\xi) = \mathcal{O}(\xi^k)$, but now $\mathbf{z}^T[k-1/k-1]_{\mathbf{f}}(\xi) - \mathbf{z}^T\mathbf{f}(\xi) = \mathcal{O}(\xi^{2k-1})$. These approximants are more costly than the previous ones since more iterates of the power method are needed. They can be recursively computed by applying the topological ϵ-algorithm [6] to the iterates of the power method.

Intermediate strategies between using one vector equation and $k-1$ vectors $\mathbf{z}_i$ or using $k-1$ vector equations and only one vector $\mathbf{z}$ could also be employed as those described in [11]. Other sequence transformations of the same type are given in [15].

By Property 2, $\mathbf{r}_c$ is also the product of the vector $\mathbf{v}$ by a matrix power series. Thus, $\mathbf{r}_c$ can be approximated by matrix Padé approximants. However, this solution involves high-dimensional matrix inversion, which is not possible in our case.

Let us now apply these general procedures for constructing Padé-type and Padé approximants of $\mathbf{r}_c$. The second series of Property 2 for $\mathbf{r}_c$ corresponds to $\mathbf{f}$ with $\xi = c$, $\boldsymbol{\sigma}_0 = \mathbf{v}$, and $\boldsymbol{\sigma}_i = (\widetilde{A} - I)\widetilde{A}^{i-1}\mathbf{v}$ for $i \geq 1$. Following Property 9, these vector coefficients are obtained directly from the power method since $\boldsymbol{\sigma}_i = (\mathbf{r}_c^{(i+1)} - \mathbf{r}_c^{(i)})/c^{i+1}$. The vector Padé-type approximants satisfy the same property as $\mathbf{r}_c$ and the iterates of the power method, namely, we have the following.

PROPERTY 13.

$$\begin{aligned} \mathbf{e}^T(k-1/k-1)_{\mathbf{f}}(c) &= 1 \quad \forall c \in [0,1], \\ (k-1/k-1)_{\mathbf{f}}(c) &\geq 0 \quad \forall c \in [0,\delta], \quad \delta \in [0,1]. \end{aligned}$$

Proof. We have $\mathbf{e}^T\boldsymbol{\sigma}_0 = 1$, and $\mathbf{e}^T\boldsymbol{\sigma}_i = 0$ for $i \geq 1$. Thus, multiplying the relations (4) by $\mathbf{e}^T$ gives $\mathbf{e}^T\mathbf{a}_i = b_i$ for $i = 0, \ldots, k-1$, which shows the first result.

Since $(k-1/k-1)_{\mathbf{f}}(c)$ approximates $\mathbf{r}_c$ near zero, and for all $c \in [0,1), \mathbf{r}_c \geq 0$, then $\exists\delta \in [0,1]$ such that $(k-1/k-1)_{\mathbf{f}}(c) \geq 0$ for $c \in [0,\delta]$. Let us mention that the value of δ is unknown but that is should be 1. □

Let us also mention that the value of c in Padé-style approximants can be complex.

6. Acceleration of the power method. Let us begin by recalling some general issues about sequence transformations for accelerating the convergence.

Let $(\mathbf{x}^{(n)})$ be a sequence converging to a limit $\mathbf{x}$. The idea behind a convergence acceleration method is extrapolation. It is assumed that the sequence $(\mathbf{x}^{(n)})$ behaves in a certain way or, in other terms, as a certain function of n depending on some unknown parameters including its limit $\mathbf{x}$. These parameters (and so also the limit $\mathbf{x}$) are determined by interpolation starting from an index n, and then the function is extrapolated at infinity. If the sequence $(\mathbf{x}^{(n)})$ behaves exactly as the extrapolation function, then the estimated limit obtained by the extrapolation process gives its exact limit $\mathbf{x}$. If $(\mathbf{x}^{(n)})$ does not behave exactly as the extrapolation function, then the estimated limit given by the extrapolation process is only an approximation of $\mathbf{x}$, denoted by $\mathbf{y}^{(n)}$ since it depends on n. So, an extrapolation process transforms the sequence $(\mathbf{x}^{(n)})$ into a new sequence $(\mathbf{y}^{(n)})$ which, under certain assumptions (that is, if the extrapolation function closely follows the exact behavior of $(\mathbf{x}^{(n)})$), converges to $\mathbf{x}$ faster than $(\mathbf{x}^{(n)})$, that is, $\lim_{n\to\infty} \|\mathbf{y}^{(n)} - \mathbf{x}\|/\|\mathbf{x}^{(n)} - \mathbf{x}\| = 0$. A sequence transformation for accelerating the convergence can always be considered as an extrapolation procedure, and conversely an extrapolation procedure always leads to an acceleration method for some classes of sequences. A sequence transformation does not modify the sequence to accelerate, and the way it is generated is taken into account only for obtaining acceleration results.

The set $\mathcal{K}_T$ of sequences such that, for all n, $\mathbf{y}^{(n)} = \mathbf{x}$ is called the *kernel* of the transformation $T : (\mathbf{x}^{(n)}) \longmapsto (\mathbf{y}^{(n)})$. So, when a sequence belongs to the kernel of a transformation, its limit is exactly obtained. An important conjecture about sequence transformations is that, if a sequence is *close*, in a sense to be specified, to the kernel of a transformation, then its convergence will be accelerated by this transformation. Many numerical results point out that this conjecture is true. However, theoretical results in this direction are very partial, and no general ones exist.

Thus, for constructing an efficient acceleration process for a given sequence, one must first study its behavior with respect to n, and then construct an extrapolation process based on an extrapolation function as close as possible (in some sense) to the exact one. This extrapolation function will define the kernel of the transformation.

On sequence transformations for accelerating the convergence by extrapolation, and, in particular, for the ϵ-algorithm and the other algorithms that will be used below, see [13].

For defining such an acceleration process for the iterates of the power method for the computation of the PageRank vector, we use the idea behind Krylov's method. As proved in Property 5, $\mathbf{r}_c = Q_{m-1}(A_c)\mathbf{v}$, where $\Pi_m(\lambda) = (\lambda - 1)Q_{m-1}(\lambda)$ is the minimal polynomial of A_c for the vector $\mathbf{v}$. We also have

$$\mathbf{r}_c = A_c^n \mathbf{r}_c = A_c^n Q_{m-1}(A_c)\mathbf{v} = Q_{m-1}(A_c)A_c^n\mathbf{v} = Q_{m-1}(A_c)\mathbf{r}_c^{(n)}. \tag{5}$$

Thus, replacing, in this relation, Q_{m-1} by an approximating polynomial Q_{k-1} of degree $k-1 \leq m-1$ leads to polynomial approximations of $\mathbf{r}_c$ of the form

$$\mathbf{r}_c^{(k,n)} = Q_{k-1}(A_c)\mathbf{r}_c^{(n)}. \tag{6}$$

As will be seen below, the polynomials Q_{k-1} will be constructed from the vectors $\mathbf{r}_c^{(i)}$ for $i \geq n$. Under certain assumptions, the new sequences $(\mathbf{r}_c^{(k,n)})$ will converge to $\mathbf{r}_c$ faster than the sequence $(\mathbf{r}_c^{(n+k)})$ produced by the power method, that is, such that the sequence $(\|\mathbf{r}_c^{(k,n)} - \mathbf{r}_c\|/\|\mathbf{r}_c^{(n)} - \mathbf{r}_c\|)$ tends to zero, for k fixed and n tending to infinity. When n is fixed and k increases, then for $k = m$, the degree of the minimal polynomial of A_c for the vector $\mathbf{v}$, the exact result $\mathbf{r}_c$ is obtained. Obviously, since m is very large this is not a procedure that could be used in practice. However, when k grows, the $(\mathbf{r}_c^{(k,n)})$'s become more accurate approximations of $\mathbf{r}_c$ since we are getting closer to the kernel of the transformation as explained above.

Another procedure, called *cycling*, consists of computing $\mathbf{r}_c^{(k,0)}$ as above, and then starting again the iterations (1) of the power method from $\mathbf{r}_c^{(0)} = \mathbf{r}_c^{(k,0)}$. This was the strategy used in [31].

In both cases, it must be noted that the higher k, the greater the number of vectors $\mathbf{r}_c^{(i)}$ obtained by the power method to store, thus limiting the value of k to be used in practice. This value depends on the numbers of vectors one could store. This is an important point to take into consideration since, if many vectors have to be stored for accelerating the power method, one might as well use a more powerful eigenvalue algorithm, like the restarted Arnoldi's method [23]. Even if the PageRank vector is computed only for a subset of the pages, again Arnoldi's method may be more interesting than the accelerated power method. This is a remark to take into account when considering convergence acceleration procedures.

In section 6.1, we will explain in a different way, simplifying, unifying, and generalizing the Quadratic Extrapolation presented in [31]. This generalization will be related to Krylov subspace methods, and some properties will be given. Then, in section 6.2, this generalization will also be included in the framework of the method of moments, where a polynomial P_k approximating the minimal polynomial Π_m will be constructed. In section 6.3, we will discuss the various ϵ-algorithms and recover the Aitken Extrapolation, as well as the Epsilon Extrapolation given in [31]. These algorithms are generalizations of the well-known Aitken's Δ^2 process. The section closes by reviewing some other possible acceleration methods.

6.1. Vector least squares extrapolation. Let us give a first procedure for computing the coefficients of the polynomial P_k which approximates the minimal polynomial Π_m. We set $P_k(\lambda) = a_0 + \cdots + a_k\lambda^k$, where the a_i's depend on k and another index denoted by n, as we will see, and $P_k(1) = a_0 + \cdots + a_k = 0$. Considering the iterates of the power method, we set

$$R_n = [\mathbf{r}_c^{(n)}, \ldots, \mathbf{r}_c^{(n+k-1)}].$$

Since, for all $n, \mathbf{r}_c^{(n)} = A_c^n\mathbf{v}$, it holds that

$$A_c^n P_k(A_c)\mathbf{v} = P_k(A_c)\mathbf{r}_c^{(n)} = a_0\mathbf{r}_c^{(n)} + \cdots + a_k\mathbf{r}_c^{(n+k)} \simeq 0. \tag{7}$$

Since the coefficients a_i are defined apart from a multiplying factor, and since P_k has exact degree k, we can assume that $a_k = 1$ without restricting the generality. Thus, (7) can be rewritten as

$$R_n\mathbf{a} \simeq -\mathbf{r}_c^{(n+k)},$$

with $\mathbf{a} = (a_0, \ldots, a_{k-1})^T$. Solving this system in the least squares sense gives

$$\mathbf{a} = -(R_n^T R_n)^{-1} R_n^T \mathbf{r}_c^{(n+k)}. \tag{8}$$

Let us remark, in connection with [31], that $(R_n^T R_n)^{-1} R_n^T$ is the pseudoinverse of R_n.

By taking into account that $P_k(1) = 0$, the computation can be simplified as in [31]. We have $a_0 = -a_1 - \cdots - a_{k-1} - 1$. Replacing a_0 by this expression in (7) gives

$$R'_n\mathbf{a}' = -(\mathbf{r}_c^{(n+k)} - \mathbf{r}_c^{(n)}) \tag{9}$$

with $R'_n = [\mathbf{r}_c^{(n+1)} - \mathbf{r}_c^{(n)}, \ldots, \mathbf{r}_c^{(n+k-1)} - \mathbf{r}_c^{(n)}]$ and $\mathbf{a}' = (a_1, \ldots, a_{k-1})^T$. This system is then solved in the least squares sense, that is, $\mathbf{a}' = -(R_n^{'T} R'_n)^{-1} R_n^{'T}(\mathbf{r}_c^{(n+k)} - \mathbf{r}_c^{(n)})$.

Remark 3. Instead of formula (8), any other left inverse of R_n could be used, thus leading to

$$\mathbf{a} = -(Z_n^T R_n)^{-1} Z_n^T \mathbf{r}_c^{(n+k)},$$

where $Z_n = [\mathbf{z}_n, \ldots, \mathbf{z}_{n+k-1}]$ is a $p \times k$ matrix such that $Z_n^T R_n$ is nonsingular. The system (9) can be solved in a similar way.

We now have to compute $\mathbf{r}_c^{(k,n)}$ by (6). We set

$$Q_{k-1}(\lambda) = b_0 + b_1\lambda + \cdots + b_{k-1}\lambda^{k-1}.$$

Since $P_k(\lambda) = (\lambda - 1)Q_{k-1}(\lambda)$, it follows that

$$b_i = -(a_0 + \cdots + a_i) = a_{i+1} + \cdots + a_k, \qquad i = 0, \ldots, k-1. \tag{10}$$

Note that since $a_0 + \cdots + a_k = 0$ and $a_k = 1$, we also have $b_0 = -a_0$ and $b_{k-1} = a_k = 1$. Let $\mathbf{b} = (b_0, \ldots, b_{k-1})^T$. Thus, $\mathbf{r}_c^{(k,n)} = R_n\mathbf{b}$. Denoting by L the $k \times k$ lower triangular matrix whose elements are equal to 1, then, from (10), $\mathbf{b} = -L\mathbf{a}$, and it follows that

$$\mathbf{r}_c^{(k,n)} = R_n\mathbf{b} = -R_nL\mathbf{a} = R_nL(R_n^TR_n)^{-1}R_n^Tr_c^{(n+k)}.$$

We also have $A_c\mathbf{r}_c^{(k,n)} = R_{n+1}\mathbf{b}$.

Thus, from what precedes, we obtain

$$\mathbf{r}_c^{(k,n)} = Q_{k-1}(A_c)\mathbf{r}_c^{(n)} = b_0\mathbf{r}_c^{(n)} + b_1\mathbf{r}_c^{(n+1)} + \cdots + b_{k-1}\mathbf{r}_c^{(n+k-1)}. \tag{11}$$

Since $\mathbf{r}_c^{(n+i)} = A_c^i\mathbf{r}_c^{(n)}$, this relation shows that $\mathbf{r}_c^{(k,n)} \in K_k(A_c, \mathbf{r}_c^{(n)})$, the Krylov subspace of dimension k spanned by the vectors $\mathbf{r}_c^{(n)}, \ldots, A_c^{k-1}\mathbf{r}_c^{(n)}$. More precisely, since $b_{k-1} = 1$, $\mathbf{r}_c^{(k,n)} \in \mathbf{r}_c^{(n+k-1)} + K_{k-1}(A_c, \mathbf{r}_c^{(n)})$. Moreover, the vector

$$\mathbf{e}^{(k,n)} = P_k(A_c)\mathbf{r}_c^{(n)} = (A_c - I)Q_{k-1}(A_c)\mathbf{r}_c^{(n)} = A_c\mathbf{r}_c^{(k,n)} - \mathbf{r}_c^{(k,n)}$$

belongs to $K_{k+1}(A_c, \mathbf{r}_c^{(n)})$, more precisely, since $b_{k-1} = 1$, $\mathbf{e}^{(k,n)} \in \mathbf{r}_c^{(n+k)} + K_k(A_c, \mathbf{r}_c^{(n)})$. From (11), we also see that $\mathbf{e}^{(k,n)} = \Delta R_n\mathbf{b} \in K_k(A_c, \Delta\mathbf{r}_c^{(n)})$; more precisely, it belongs to $\Delta\mathbf{r}_c^{(n+k-1)} + K_{k-1}(A_c, \Delta\mathbf{r}_c^{(n)})$ (Δ is the usual forward difference operator). Since $\mathbf{r}_c^{(k,n)}$ approximates the eigenvector $\mathbf{r}_c$ of A_c, the vector $\mathbf{e}^{(k,n)}$ plays the role of a residual. We have

$$\begin{aligned} R_n^T\mathbf{e}^{(k,n)} &= R_n^TR_n\mathbf{a} + R_n^T\mathbf{r}_c^{(n+k)} \\ &= -R_n^TR_n(R_n^TR_n)^{-1}R_n^T\mathbf{r}_c^{(n+k)} + R_n^T\mathbf{r}_c^{(n+k)} \\ &= 0. \end{aligned}$$

Thus, $\mathbf{e}^{(k,n)}$ is orthogonal to the columns of R_n, and we have the following.

THEOREM 1.

$$\begin{aligned} \mathbf{r}_c^{(k,n)} &\in \mathbf{r}_c^{(n+k-1)} + K_{k-1}(A_c, \mathbf{r}_c^{(n)}), \\ A_c\mathbf{r}_c^{(k,n)} - \mathbf{r}_c^{(k,n)} &\perp K_k(A_c, \mathbf{r}_c^{(n)}). \end{aligned}$$

This result shows that vector least squares extrapolation can be considered as a Krylov subspace method for computing $\mathbf{r}_c$.

Moreover, since $K_k(A_c, \mathbf{r}_c^{(n)}) \subseteq K_{k+1}(A_c, \mathbf{r}_c^{(n)})$, we have the following.

Corollary 1.

$$\|\mathbf{e}^{(k+1,n)}\| \leq \|\mathbf{e}^{(k,n)}\|.$$

Obviously, when $k = m$, $\mathbf{e}^{(m,n)} = 0$.

Writing down the conditions of Theorem 1, we immediately obtain several determinantal expressions. Such expressions have no direct practical use, but they could be of interest in proving theoretical results about our vector least squares extrapolation, and in obtaining recursive algorithms for the computation of the vectors $\mathbf{r}_c^{(k,n)}$.

Corollary 2.

$$\mathbf{e}^{(k,n)} = (-1)^{k-1} \frac{\begin{vmatrix} \mathbf{r}_c^{(n)} & \cdots & \mathbf{r}_c^{(n+k)} \\ (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n+k)}) \\ \vdots & & \vdots \\ (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n+k)}) \end{vmatrix}}{\begin{vmatrix} (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n+k-1)}) \\ \vdots & & \vdots \\ (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n+k-1)}) \end{vmatrix}}.$$

The determinant in the numerator denotes the vector obtained by expanding it with respect to its first row by the classical rules for expanding a determinant.

Let $D_k^{(n)}$ be the determinant in the denominator of $\mathbf{e}^{(k,n)}$. Comparing this result with (11) shows, since $\mathbf{r}_c^{(n+i)} = A_c^i \mathbf{r}_c^{(n)}$, that we have the following.

Corollary 3. *It holds that* $\mathbf{e}^{(k,n)} = \widetilde{Q}_{k-1}(A_c)\mathbf{r}_c^{(n)}/D_k^{(n)}$, *with*

$$\widetilde{Q}_{k-1}(\lambda) = (-1)^{k-1} \begin{vmatrix} 1 & \cdots & \lambda^k \\ (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n+k)}) \\ \vdots & & \vdots \\ (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n+k)}) \end{vmatrix}.$$

Note that the polynomial $\widetilde{Q}_{k-1}(\lambda)/D_k^{(n)}$ is monic. Moreover, the ratio of determinants given in Corollary 2 shows that $\mathbf{e}^{(k,n)}$ can also be expressed as a Schur complement (see [9, p. 150] or [52]), thus leading to the following.

Corollary 4.

$$\mathbf{e}^{(k,n)} = \mathbf{r}_c^{(n+k)} - R_n \begin{pmatrix} (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n+k-1)}) \\ \vdots & & \vdots \\ (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n)}) & \cdots & (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n+k-1)}) \end{pmatrix}^{-1} \begin{pmatrix} (\mathbf{r}_c^{(n)}, \mathbf{r}_c^{(n+k)}) \\ \vdots \\ (\mathbf{r}_c^{(n+k-1)}, \mathbf{r}_c^{(n+k)}) \end{pmatrix}.$$

Let us now express the vectors $\mathbf{r}_c^{(k,n)}$ as a ratio of determinants. We have the following theorem.

Theorem 2.

$$\mathbf{r}_c^{(k,n)} = (-1)^{k-1} \frac{\begin{vmatrix} \mathbf{r}_c^{(n)} & \cdots & \mathbf{r}_c^{(n+k-1)} \\ (\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n)}) & \cdots & (\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n+k-1)}) \\ \vdots & & \vdots \\ (\Delta\mathbf{r}_c^{(n+k-2)}, \Delta\mathbf{r}_c^{(n)}) & \cdots & (\Delta\mathbf{r}_c^{(n+k-2)}, \Delta\mathbf{r}_c^{(n+k-1)}) \end{vmatrix}}{\begin{vmatrix} (\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n)}) & \cdots & (\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n+k-2)}) \\ \vdots & & \vdots \\ (\Delta\mathbf{r}_c^{(n+k-2)}, \Delta\mathbf{r}_c^{(n)}) & \cdots & (\Delta\mathbf{r}_c^{(n+k-2)}, \Delta\mathbf{r}_c^{(n+k-2)}) \end{vmatrix}}.$$

Proof. We have $\mathbf{e}^{(k,n)} = \Delta R_n \mathbf{b}$. Taking into account that $b_{k-1} = 1$, this relation can also be written as $\mathbf{e}^{(k,n)} = \Delta\widetilde{R}_n\widetilde{\mathbf{b}} + \Delta\mathbf{r}_c^{(n+k-1)}$, with $\widetilde{R}_n = [\mathbf{r}_c^{(n)}, \ldots, \mathbf{r}_c^{(n+k-2)}]$ and $\widetilde{\mathbf{b}} = (b_0, \ldots, b_{k-2})^T$. Solving, as above, the system $\mathbf{e}^{(k,n)} = 0$ in the least squares sense gives $\widetilde{\mathbf{b}} = -(\Delta\widetilde{R}_n^T\Delta\widetilde{R}_n)^{-1}\Delta\widetilde{R}_n^T\Delta\mathbf{r}_c^{(n+k-1)}$. Thus, since $\mathbf{r}_c^{(k,n)} = \widetilde{R}_n\widetilde{\mathbf{b}} + \mathbf{r}_c^{(n+k-1)}$, we get

$$\mathbf{r}_c^{(k,n)} = \mathbf{r}_c^{(n+k-1)} - \widetilde{R}_n(\Delta\widetilde{R}_n^T\Delta\widetilde{R}_n)^{-1}\Delta\widetilde{R}_n^T\Delta\mathbf{r}_c^{(n+k-1)}.$$

This relation shows that $\mathbf{r}_c^{(k,n)}$ is a Schur complement, and the result follows from Schur's determinantal formula. □

Since $A_c\mathbf{r}_c^{(k,n)} = R_{n+1}\mathbf{b}$, we immediately have the following.

Corollary 5. $\mathbf{e}^{(k,n)}$ *is given by a formula similar to the formula of Theorem* 2 *after replacing the first row of the numerator by* $\Delta\mathbf{r}_c^{(n)}, \ldots, \Delta\mathbf{r}_c^{(n+k-1)}$. *Moreover* $(\Delta\mathbf{r}_c^{(n+i)}, \mathbf{e}^{(k,n)}) = 0$ *for* $i = 0, \ldots, k-2$, *that is,* $\Delta\widetilde{R}_n^T\mathbf{e}^{(k,n)} = 0$.

Note that $R_n = [\widetilde{R}_n, \mathbf{r}_c^{(n+k-1)}]$, and $\mathbf{b} = (\widetilde{\mathbf{b}}^T, b_{k-1})^T$. Polynomial expressions for $\mathbf{r}_c^{(k,n)}$ and $\mathbf{e}^{(k,n)}$ similar to that of Corollary 3 can easily be deduced from Theorem 2 and Corollary 5. The preceding results can be easily modified if R_n is replaced by Z_n.

Thus, in this section, we have generalized to an arbitrary value of k the Quadratic Extrapolation presented in [31] which corresponds to $k = 3$. Moreover, it has been related to Krylov subspace methods.

In practice, the value of k is limited by the dimension p of the problem and by the number of vectors to store for computing the vector $\mathbf{r}_c^{(k,n)}$. For $k = 2$, we obtain the new vector sequence transformation

$$\mathbf{r}_c^{(2,n)} = (A_c - \alpha_n I)\mathbf{r}_c^{(n)} = \mathbf{r}_c^{(n+1)} - \alpha_n\mathbf{r}_c^{(n)} \quad \text{with} \quad \alpha_n = \frac{(\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n+1)})}{(\Delta\mathbf{r}_c^{(n)}, \Delta\mathbf{r}_c^{(n)})}.$$

This relation corresponds to the ratio of determinants given in Theorem 2.

These vector least squares extrapolation procedures follow an idea similar to that used in the least squares extrapolation discussed in [13, sect. 3.10] for scalar sequences and in the vector transformations proposed in [15].

6.2. The method of moments. The generalization of the Quadratic Extrapolation [31] discussed in the previous section could be interpreted as a special case of the method of moments of Vorobyev [50, pp. 14–16] (see also [8, pp. 154–157]). Thus, we will have a different point of view on this generalization, which is always helpful for obtaining theoretical results, such as acceleration properties.

Let $\mathbf{u}_0, \ldots, \mathbf{u}_k$ be linearly independent vectors in $\mathbb{R}^p$ and $\mathbf{z}_0, \ldots, \mathbf{z}_{k-1}$ also, where $k+1 \leq p$. The method of moments consists of constructing the linear mapping A_k on $E_k = \operatorname{span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1})$ such that

$$\begin{aligned}
&\mathbf{u}_1 = A_k \mathbf{u}_0,\\
&\mathbf{u}_2 = A_k \mathbf{u}_1 = A_k^2 \mathbf{u}_0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&\mathbf{u}_{k-1} = A_k \mathbf{u}_{k-2} = A_k^{k-1} \mathbf{u}_0,\\
&\mathbf{P}_k \mathbf{u}_k = A_k \mathbf{u}_{k-1} = A_k^k \mathbf{u}_0,
\end{aligned}$$

where $\mathbf{P}_k$ is the projection on E_k orthogonal to $F_k = \operatorname{span}(\mathbf{z}_0, \ldots, \mathbf{z}_{k-1})$.

These relations completely determine the mapping A_k. Indeed, for any $\mathbf{u} \in E_k$, there exist numbers $b_0, \ldots, b_{k-1}$ such that

$$\mathbf{u} = b_0 \mathbf{u}_0 + \cdots + b_{k-1} \mathbf{u}_{k-1}. \tag{12}$$

Thus,

$$\begin{aligned}
A_k \mathbf{u} &= b_0 A_k \mathbf{u}_0 + \cdots + b_{k-2} A_k \mathbf{u}_{k-2} + b_{k-1} A_k \mathbf{u}_{k-1}\\
&= b_0 \mathbf{u}_1 + \cdots + b_{k-2} \mathbf{u}_{k-1} + b_{k-1} \mathbf{P}_k \mathbf{u}_k \in E_k.
\end{aligned} \tag{13}$$

Since $\mathbf{P}_k \mathbf{u}_k \in E_k$, there exist numbers $a_0, \ldots, a_{k-1}$ such that

$$\mathbf{P}_k \mathbf{u}_k = -a_0 \mathbf{u}_0 - \cdots - a_{k-1} \mathbf{u}_{k-1}, \tag{14}$$

that is,

$$a_0 \mathbf{u}_0 + \cdots + a_{k-1} \mathbf{u}_{k-1} + \mathbf{P}_k \mathbf{u}_k = (a_0 + a_1 A_k + \cdots + a_{k-1} A_k^{k-1} + A_k^k) \mathbf{u}_0 = 0.$$

But

$$(\mathbf{z}_i, \mathbf{u}_k - \mathbf{P}_k \mathbf{u}_k) = 0 \quad \text{for } i = 0, \ldots, k-1,$$

that is, for $i = 0, \ldots, k-1$,

$$a_0 (\mathbf{z}_i, \mathbf{u}_0) + \cdots + a_{k-1} (\mathbf{z}_i, \mathbf{u}_{k-1}) + (\mathbf{z}_i, \mathbf{u}_k) = 0.$$

Solving this system gives the a_i's and, thus, A_k is completely determined.

Now, if we set

$$P_k(\xi) = a_0 + \cdots + a_{k-1} \xi^{k-1} + \xi^k,$$

then

$$P_k(A_k) \mathbf{u}_0 = 0,$$

which shows that P_k is an annihilating polynomial of A_k for the vector $\mathbf{u}_0$.

We will be looking for the eigenvectors of A_k belonging to E_k. Let $\mathbf{u} \in E_k$. From (13) and (14), we have

$$\begin{aligned}
A_k \mathbf{u} &= b_0 \mathbf{u}_1 + \cdots + b_{k-2} \mathbf{u}_{k-1} + b_{k-1} (-a_0 \mathbf{u}_0 - \cdots - a_{k-1} \mathbf{u}_{k-1})\\
&= -a_0 b_{k-1} \mathbf{u}_0 + (b_0 - a_1 b_{k-1}) \mathbf{u}_1 + \cdots + (b_{k-2} - a_{k-1} b_{k-1}) \mathbf{u}_{k-1}.
\end{aligned} \tag{15}$$

If λ is an eigenvalue of A_k and $\mathbf{u}$ is the corresponding eigenvector, then

$$A_k\mathbf{u} = \lambda(b_0\mathbf{u}_0 + \cdots + b_{k-1}\mathbf{u}_{k-1}),$$

and, since $\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}$ are linearly independent in E_k, we see from (15) that we must have

$$\begin{aligned} -a_0 b_{k-1} &= b_0\lambda, \\ b_i - a_{i+1}b_{k-1} &= b_{i+1}\lambda, \qquad i = 0, \ldots, k-2, \end{aligned} \tag{16}$$

that is, in matrix form,

$$\begin{pmatrix} -\lambda & 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & -\lambda & \ddots & & 0 & -a_1 \\ 0 & 1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & -\lambda & 0 & -a_{k-3} \\ \vdots & & \ddots & 1 & -\lambda & -a_{k-2} \\ 0 & \cdots & \cdots & 0 & 1 & -a_{k-1}-\lambda \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-3} \\ b_{k-2} \\ b_{k-1} \end{pmatrix} = 0.$$

Since this system has a nonzero solution, its determinant must be zero, that is,

$$P_k(\lambda) = 0.$$

Moreover, we must have $b_{k-1} \neq 0$, since otherwise all the b_i's would be zero. Since an eigenvector is defined up to a multiplying factor, then b_{k-1} can be arbitrarily set to 1 and, from (16), we have

$$b_i = a_{i+1} + b_{i+1}\lambda, \qquad i = k-2, \ldots, 0.$$

We see that, for $\lambda = 1$, these relations are the same as (10).

As seen above, for $\mathbf{u}$ as in (12), $A_k\mathbf{u}$ is given by (15), and the transformation mapping the coordinates $b_0, \ldots, b_{k-1}$ of $\mathbf{u}$ in the basis formed by the elements $\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}$ into the coordinates of $A_k\mathbf{u}$ in the same basis is given by the matrix $\widetilde{A}_k$ of the system

$$\begin{pmatrix} 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & \ddots & & \vdots & -a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & -a_{k-2} \\ 0 & \cdots & 0 & 1 & -a_{k-1} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-2} \\ b_{k-1} \end{pmatrix} = \begin{pmatrix} -a_0 b_{k-1} \\ b_0 - b_{k-1}a_1 \\ \vdots \\ b_{k-3} - b_{k-2}a_{k-2} \\ b_{k-2} - b_{k-1}a_{k-1} \end{pmatrix}.$$

Thus, the polynomial P_k is the characteristic polynomial of the $k \times k$ matrix $\widetilde{A}_k$ which represents the mapping A_k in E_k. Consequently, $\widetilde{A}_k$ is regular if and only if $a_0 \neq 0$, and the rank of A_k is equal to the rank of $\widetilde{A}_k$.

In the particular case where $\mathbf{u}_i = A_c^i\mathbf{u}_0$, $i = 0, 1, \ldots,$ which is the case we treated, it is possible to obtain an expression for A_k. Let $\mathbf{u}$ be as in (12). Then

$$\begin{aligned} A_c\mathbf{u} &= b_0A_c\mathbf{u}_0 + \cdots + b_{k-2}A_c\mathbf{u}_{k-2} + b_{k-1}A_c\mathbf{u}_{k-1} \\ &= b_0A_c\mathbf{u}_0 + \cdots + b_{k-2}A_c^{k-1}\mathbf{u}_0 + b_{k-1}A_c^k\mathbf{u}_0 \\ &= b_0A_k\mathbf{u}_0 + \cdots + b_{k-2}A_k^{k-1}\mathbf{u}_0 + b_{k-1}A_c^k\mathbf{u}_0, \end{aligned}$$

and it follows that

$$\begin{aligned}\mathbf{P}_k A_c \mathbf{u} &= b_0 A_k \mathbf{u}_0 + \cdots + b_{k-2} A_k^{k-1} \mathbf{u}_0 + b_{k-1} \mathbf{P}_k \mathbf{u}_k \\ &= b_0 A_k \mathbf{u}_0 + \cdots + b_{k-2} A_k^{k-1} \mathbf{u}_0 + b_{k-1} A_k^{k} \mathbf{u}_0 \\ &= A_k (b_0 \mathbf{u}_0 + \cdots + b_{k-1} \mathbf{u}_{k-1}) = A_k \mathbf{u},\end{aligned}$$

which shows that $A_k = \mathbf{P}_k A_c$ on E_k. Since, if $\mathbf{u} \in E_k$, $\mathbf{P}_k \mathbf{u} \in E_k$, then the domain of A_k can be extended to the whole space $\mathbb{R}^p$ by setting

$$A_k = \mathbf{P}_k A_c \mathbf{P}_k.$$

Now let $\mathbf{u} \in \mathbb{R}^p$. Setting $U_k = [\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}]$, $Z_k = [\mathbf{z}_0, \ldots, \mathbf{z}_{k-1}]$, and $\mathbf{a} = (a_0, \ldots, a_{k-1})^T$, the conditions $(\mathbf{P}_k \mathbf{u} - \mathbf{u}, \mathbf{z}_i) = 0$ for $i = 0, \ldots, k-1$ can be written as $Z_k^T U_k \mathbf{a} = -Z_k^T \mathbf{u}$ and it follows that $\mathbf{P}_k \mathbf{u} = -U_k \mathbf{a} = U_k (Z_k^T U_k)^{-1} Z_k^T \mathbf{u}$, which gives

$$\mathbf{P}_k = U_k (Z_k^T U_k)^{-1} Z_k^T.$$

It must be noted that A_k is not an injection since $\mathbf{P}_k$ is not.

6.3. The $\boldsymbol{\epsilon}$-algorithms. The ϵ-algorithms are sequence transformations which map a given sequence into new ones which, under certain assumptions, converge faster to the same limit. Let us now discuss the various ϵ-algorithms for vector sequences.

As above, let $(\mathbf{x}^{(n)})$ be a vector sequence converging to $\mathbf{x}$. The vector ϵ-algorithm consists in the recursive rule

$$\begin{aligned}\boldsymbol{\epsilon}_{-1}^{(n)} &= 0, \\ \boldsymbol{\epsilon}_{0}^{(n)} &= \mathbf{x}^{(n)}, \\ \boldsymbol{\epsilon}_{j+1}^{(n)} &= \boldsymbol{\epsilon}_{j-1}^{(n+1)} + \left[\boldsymbol{\epsilon}_{j}^{(n+1)} - \boldsymbol{\epsilon}_{j}^{(n)}\right]^{-1}\end{aligned}$$

for $j = 0, 1, \ldots$ and $n = 0, 1, \ldots$, where the inverse of a vector $\mathbf{y}$ is defined by $\mathbf{y}^{-1} = \mathbf{y}/(\mathbf{y}, \mathbf{y})$. The vectors with an odd lower index are intermediate computations without any interesting meaning, while those with an even lower index approximate $\mathbf{x}$. These rules are also valid for the scalar ϵ-algorithm (in which case the ϵ's are not in bold in what follows) with $\epsilon_0^{(n)} = (\mathbf{x}^{(n)})_i$, the ith component of $\mathbf{x}^{(n)}$. The rules of the topological ϵ-algorithm are slightly different, and can be found, for example, in [13, sect. 4.2]. The computation of $\boldsymbol{\epsilon}_{2k}^{(n)}$ needs the knowledge of $\mathbf{x}^{(n)}, \ldots, \mathbf{x}^{(n+2k)}$ and the storage of $2k+1$ vectors, thus restricting k to small values in our case.

The kernels of the scalar ϵ-algorithm (applied separately on each components), of the vector ϵ-algorithm, and of the topological ϵ-algorithm contain the set of sequences satisfying the characteristic relation

$$b_0(\mathbf{x}^{(n)} - \mathbf{x}) + \cdots + b_{k-1}(\mathbf{x}^{(n+k-1)} - \mathbf{x}) = 0, \quad n = 0, 1, \ldots, \tag{17}$$

where the b_i's are any numbers satisfying $b_0 b_{k-1} \neq 0$. Thus, if one of these ϵ-algorithms is applied to a sequence $(\mathbf{x}^{(n)})$ satisfying (17), then, by construction, $\boldsymbol{\epsilon}_{2k-2}^{(n)} = \mathbf{x}$ for $n = 0, 1, \ldots$.

Let us now study our particular case. From (5), we have $\mathbf{r}_c = Q_{m-1}(A_c)\mathbf{r}_c^{(n)}$. Moreover, since $\mathbf{r}_c = A_c^i \mathbf{r}_c$ for all i, $\sum_{i=0}^{m-1} b_i \mathbf{r}_c = \sum_{i=0}^{m-1} b_i A_c^i \mathbf{r}_c = Q_{m-1}(A_c)\mathbf{r}_c = \mathbf{r}_c$, assuming that $\sum_{i=0}^{m-1} b_i = 1$, which does not restrict the generality. Thus, subtracting the second relation from the first one, we get the following.

PROPERTY 14.

$$Q_{m-1}(A_c)(\mathbf{r}_c^{(n)} - \mathbf{r}_c) = b_0(\mathbf{r}_c^{(n)} - \mathbf{r}_c) + \cdots + b_{m-1}(\mathbf{r}_c^{(n+m-1)} - \mathbf{r}_c) = 0, \quad n = 0, 1, \ldots.$$

Thus, applying one of the ϵ-algorithms to the vector sequence $(\mathbf{r}_c^{(n)})$ gives $\boldsymbol{\epsilon}_{2m-2}^{(n)} = \mathbf{r}_c$ for $n = 0, 1, \ldots$ and produces approximations $\boldsymbol{\epsilon}_{2k-2}^{(n)}$ of $\mathbf{r}_c$ for $k < m$. Since, by the theory of the ϵ-algorithms, there exist numbers $b_0', \ldots, b_{k-1}'$ such that

$$b_0'(\mathbf{r}_c^{(i)} - \boldsymbol{\epsilon}_{2k-2}^{(n)}) + \cdots + b_{k-1}'(\mathbf{r}_c^{(i+k-1)} - \boldsymbol{\epsilon}_{2k-2}^{(n)}) = 0, \quad i = 0, 1, \ldots,$$

then

$$\boldsymbol{\epsilon}_{2k-2}^{(n)} = Q_{k-1}(A_c)\mathbf{r}_c^{(n)}, \qquad n = 0, 1, \ldots,$$

with $Q_{k-1}(\lambda) = b_0' + \cdots + b_{k-1}'\lambda^{k-1}$. These vectors $\boldsymbol{\epsilon}_{2k-2}^{(n)}$ are rational approximations of $\mathbf{r}_c$ in the Padé style. In the case of the topological ϵ-algorithm, it is well known that the vectors it computes can be represented as a ratio of determinants, and we have (see, for example, [13, p. 221]) the following.

THEOREM 3. *For the topological ϵ-algorithm, $\boldsymbol{\epsilon}_{2k-2}^{(n)} = \widetilde{Q}_{k-1}(A_c)\mathbf{r}_c^{(n)}/\widetilde{Q}_{k-1}(1)$, with*

$$\widetilde{Q}_{k-1}(\lambda) = \begin{vmatrix} 1 & \cdots & \lambda^k \\ (\mathbf{y}, \Delta\mathbf{r}_c^{(n)}) & \cdots & (\mathbf{y}, \Delta\mathbf{r}_c^{(n+k-1)}) \\ \vdots & & \vdots \\ (\mathbf{y}, \Delta\mathbf{r}_c^{(n+k-2)}) & \cdots & (\mathbf{y}, \Delta\mathbf{r}_c^{(n+2k-3)}) \end{vmatrix},$$

where $\mathbf{y}$ is such that $\widetilde{Q}_{k-1}(1) \neq 0$.

Let us now analyze the behavior of the vectors $\boldsymbol{\epsilon}_{2k-2}^{(n)}$ when k is fixed and n tends to infinity. The relation of Property 14 shows that the vectors $\mathbf{r}_c^{(n)} - \mathbf{r}_c$ satisfy a linear homogeneous difference equation of order $m-1$ with constant coefficients. In the particular case where the zeros $c\widetilde{\lambda}_2, \ldots, c\widetilde{\lambda}_m$ of Q_{m-1} (which are the eigenvalues of A_c) are real and simple, and all the eigenvectors of A_c are present in the spectral decomposition of $\mathbf{v}$, the solution of this difference equation is

$$\mathbf{r}_c^{(n)} = \mathbf{r}_c + \sum_{i=2}^{m}(c\widetilde{\lambda}_i)^n \mathbf{v}_i, \quad n = 0, 1, \ldots, \tag{18}$$

where the vectors $\mathbf{v}_i \in \mathbb{R}^p$ depend on the eigenvectors of A_c. The solution of the relation of Property 14 was studied in its full generality in [12] (see also [13, Thm. 2.18]), but it will not be reproduced here for length reasons. Let us mention only that if an eigenvalue $\widetilde{\lambda}_i$ has multiplicity k_i, then $\mathbf{v}_i$ is replaced in (18) by a polynomial of degree $k_i - 1$ with vector coefficients.

Using (18), we have the following convergence and acceleration results which support, in particular, the numerical results given in [31] for $k = 1$. They follow directly from the acceleration theorems proved by Wynn [51] for the scalar ϵ-algorithm and by Matos for the vector ϵ-algorithm [39]

THEOREM 4. *If A_c is diagonalizable, and if all the eigenvectors of A_c are present in the spectral decomposition of $\mathbf{v}$, then, for $1 \leq k \leq m-1$,*

$$\|\boldsymbol{\epsilon}_{2k}^{(n)} - \mathbf{r}_c\|_2 = \mathcal{O}((c\widetilde{\lambda}_{k+2})^n),$$

$$\lim_{n\to\infty} \frac{\|\boldsymbol{\epsilon}_{2k}^{(n)} - \mathbf{r}_c\|_2}{\|\boldsymbol{\epsilon}_{2k-2}^{(n)} - \mathbf{r}_c\|_2} = 0.$$

If some of the eigenvalues of A_c are multiple, they have to be counted according to their multiplicity, and the polynomial factors in the solution of the relation of Property 14 come into the discussion. However, the theory and the results remain essentially the same (in particular Theorem 4), but they become more complicated to write down (see, for example, Theorem 5 of [39]). These results are also valid for the topological ϵ-algorithm.

Other approximations of the Padé style are the vectors $\mathbf{E}_{k-1}^{(n)}$ computed by the scalar E-algorithm (applied componentwise) or the vector E-algorithm [7] (see also [13, pp. 55–72, 228–232]). Applying the convergence and acceleration results proved in [7, 45, 38], conclusions similar to those of Theorem 4 can be obtained. Let us also mention that the ϵ-algorithm and E-algorithm are related to Schur complements [52, pp. 233–238].

For the scalar ϵ-algorithm, when $k = 2$, the well-known Aitken's Δ^2 process is recovered. The kernel of Aitken's process is the set of scalar sequences $(x^{(n)})$ satisfying

$$b_0(x^{(n)} - x) + b_1(x^{(n+1)} - x) = 0, \qquad n = 0, 1, \ldots,$$

with $b_0 + b_1 \neq 0$, or, equivalently,

$$x^{(n)} = x + \alpha\mu^n, \qquad n = 0, 1, \ldots,$$

with $\mu \neq 1$. Note that the form of the first relation is the same as (17) when $k = 2$.

Aitken's Δ^2 process can be written in different ways. For example, we have the three following equivalent formulae:

$$\epsilon_2^{(n)} = x^{(n)} - \frac{(x^{(n+1)} - x^{(n)})^2}{x^{(n+2)} - 2x^{(n+1)} + x^{(n)}} \tag{19}$$

$$= x^{(n+1)} - \frac{(x^{(n+2)} - x^{(n+1)})(x^{(n+1)} - x^{(n)})}{x^{(n+2)} - 2x^{(n+1)} + x^{(n)}} \tag{20}$$

$$= x^{(n+2)} - \frac{(x^{(n+2)} - x^{(n+1)})^2}{x^{(n+2)} - 2x^{(n+1)} + x^{(n)}}. \tag{21}$$

If each component of the vectors $\mathbf{r}_c^{(n)}$ successively plays the role of $x^{(n)}$, then (19) is exactly the Aitken Extrapolation given by formula (15) of [31], while (20) corresponds to the Epsilon Extrapolation of [31]. Another implementation of the same extrapolation method can be obtained by using (21). However, let us mention that, although these are completely equivalent from the mathematical point of view, the numerical stability of these formulae can be quite different. It is well known that Aitken's process accelerates the convergence of sequences such that $\exists\delta \neq 1$, $\lim_{n\to\infty}(x^{(n+1)}-x)/(x^{(n)}-x) = \delta$, which, by Property 11, is exactly our case with $\delta = c\widetilde{\lambda}_2$. Thus, the effectiveness of the methods proposed in [31] is justified by the preceding discussion and by Theorem 4.

Each of the scalars $\epsilon_2^{(n)}$ produced by Aitken's process applied separately on each component has a different denominator. On the contrary, using the vector or the topological ϵ-algorithm for transforming the vectors $\mathbf{r}_c^{(n)}$ will lead to vectors $\boldsymbol{\epsilon}_2^{(n)}$ with the same denominator for each component, and thus will be more similar to the exact form of $\mathbf{r}_c$.

Let us recall that the ϵ-algorithms are related to various Padé-style approximants [6, 9]. If the scalar ϵ-algorithm is applied to the partial sums of a formal power series f with scalar coefficients, then the quantities $\epsilon_{2k}^{(n)}$ it computes are the Padé approximants $[n+k/k]_f$ of f. Reciprocally, the quantities $\epsilon_{2k}^{(n)}$ given by this algorithm with $\epsilon_0^{(n)} = x^{(n)}$ are the $[n+k/k]_f$ Padé approximants of the series $f(\xi) = x^{(0)} + (x^{(1)} - x^{(0)})\xi + (x^{(2)} - x^{(1)})\xi^2 + \cdots$. In particular, the $\epsilon_2^{(n)}$'s computed by Aitken's process are identical to its Padé approximants $[n/1]_f(1)$. Thus, applying the scalar ϵ-algorithm separately on each component of a series with vector coefficients, as in [31], produces Padé approximants with, in general, a different denominator for each component, while, in our case, all denominators should be identical by Properties 3 and 4. On the contrary, the vector and the topological ϵ-algorithms provide rational approximations of the series (2), but with the same denominator for all components. Thus, they seem to be better adapted to the acceleration of the power method. Moreover, it would also be interesting to consider the approximants $[k-1/k-1]$ for increasing values of k instead of the approximants $[n/1]$. It is possible to use the vector E-algorithm, which also leads to rational vector approximations of $\mathbf{r}_c$ with a unique denominator for all components, and allows more flexibility by an arbitrary choice of auxiliary vector sequences; see [13, sect. 4.3].

6.4. Other algorithms. Of course, the acceleration procedures studied above are not the only possible ones. Among them, there exist several other acceleration methods whose kernel is the set of sequences satisfying (17), where the vectors $\mathbf{x}^{(n)}$ are those obtained by the power method, and $\mathbf{x}$ is the vector $\mathbf{r}_c$ we are looking for. In this section, we will briefly review some of them, since they probably are those having the best acceleration properties, as explained at the beginning of section 6.

The relation of Property 14 can also be written as

$$\mathbf{r}_c^{(n)} = \mathbf{r}_c^{(k,n)} + \alpha_0 \Delta \mathbf{r}_c^{(n)} + \cdots + \alpha_{k-2} \Delta \mathbf{r}_c^{(n+k-2)}, \quad n = 0, 1, \ldots,$$

that is,

$$\mathbf{r}_c^{(n)} = \mathbf{r}_c^{(k,n)} + \Delta \widetilde{R}_n \boldsymbol{\alpha}, \tag{22}$$

with $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{k-2})^T$ and $\widetilde{R}_n = [\mathbf{r}_c^{(n)}, \ldots, \mathbf{r}_c^{(n+k-2)}]$, as in section 6.1.

There are several ways to compute this vector $\boldsymbol{\alpha}$. One of them leads to a sequence transformation named the *multilevel vector theta-type* (VTT) defined in [14] by

$$\mathbf{r}_c^{(k,n)} = \mathbf{r}_c^{(n)} - \Delta \widetilde{R}_n (Z_n^T \Delta^2 \widetilde{R}_n)^{-1} Z_n^T \Delta^2 \mathbf{r}_c^{(n)}, \quad n = 0, 1, \ldots, \quad k > 2,$$

where $Z_n \in \mathbb{R}^{p \times (k-1)}$.

The *multilevel biorthogonal vector theta-type* (BVTT) transformation is a particular case of the VTT, with the same kernel [14].

A general methodology, based on various strategies, for constructing sequence transformations whose kernel contains sequences of the form (22) is described in [10]. These transformations can be implemented either by one of the ϵ-algorithms given in the preceding section, by the RPA [13, sect. 4.4], or by the $S\beta$-algorithm [27]. The case where the matrix to be inverted is singular is treated similarly in [15] by using pseudoinverses and pseudo-Schur complements, whose properties are studied in [43]. Another vector sequence transformation related to the method of moments is the *modified minimal polynomial extrapolation* (MMPE) of Pugachev [42]. Application of other vector extrapolation methods, such as the RRE and the MPE, to PageRank computations are discussed in [46], but numerical experiments have yet to be carried out.

7. Conclusions.

In this paper, we analyzed the PageRank problem and its solution by the power method. Several procedures for accelerating the convergence of its iterates were proposed, and some theoretical results were given. However, no results for comparing these algorithms exist so far. When the parameter k in these acceleration methods increases, in general their efficiency increases, but the number of vectors to store also increases, thus putting a restriction on their practical use due to the huge dimension of the problem. Moreover, the behaviors of these algorithms are quite similar, and the choice between them is, more or less, a matter of taste. Thus, extensive numerical experiments have to be carried out, and perhaps they could help in making this choice.

Let us mention another problem related to PageRank computations. When c approaches 1 (which corresponds to the real PageRank vector), Property 12 shows that the speed of convergence of the power method reduces, and, moreover, the matrix A_c becomes more and more ill conditioned (as proved in [28], its condition number behaves as $(1-c)^{-1}$), the conditioning of the eigenproblem becomes poor, and $\mathbf{r}_c$ cannot be computed accurately. So, to avoid these drawbacks, $\mathbf{r}_c$ can be computed for several values of c far away from 1 by any procedure, and then these vectors can be extrapolated at the point $c = 1$ (or at any other point). In order for an extrapolation procedure to work well, the extrapolating function has to mimic as closely as possible the behavior of $\mathbf{r}_c$ with respect to the parameter c. Extrapolation algorithms based on the analysis of this dependence, given in [44], are described in [17]; see [16] for more developments.

Acknowledgments. We are grateful to Stefano Serra-Capizzano for pointing out the problem of accelerating PageRank computations and the discussions we had with him. We would like to thank Gene Golub for reading a first draft of the manuscript and suggesting additional references. We appreciated several valuable and constructive comments by Daniel Szyld and are grateful to the referees for their comments, which greatly helped us look deeper into the problem. Finally, we would like to thank Valeria Simoncini, who asked us to expand the first version of this paper for making it also a survey.

REFERENCES

[1] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin, *PageRank computation and the structure of the Web: Experiments and algorithms*, in Proceedings of the Eleventh International World Wide Web Conference, ACM Press, New York, 2002; available online from http://www2002.org/CDROM/poster/173.pdf.

[2] P. Berkhin, *A survey on PageRank computing*, Internet Math., 2 (2005), pp. 73–120.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.

[4] M. Bianchini, M. Gori, and F. Scarselli, *Inside Pagerank*, ACM Trans. Internet Technol., 5 (2005), pp. 92–128.

[5] P. Boldi, M. Santini, and S. Vigna, *PageRank as a function of the damping factor*, in Proceedings of the Fourteenth International World Wide Web Conference, ACM Press, New York, 2005, pp. 557–566.

[6] C. Brezinski, *Padé-Type Approximation and General Orthogonal Polynomials*, Internat. Ser. Numer. Math. 50, Birkhäuser Verlag, Basel, 1980.

[7] C. Brezinski, *A general extrapolation algorithm*, Numer. Math., 35 (1980), pp. 175–187.

[8] C. Brezinski, *Projections Methods for Systems of Equations*, North-Holland, Amsterdam, 1997.

[9] C. Brezinski, *Computational Aspects of Linear Control*, Kluwer, Dordrecht, 2002.

[10] C. Brezinski, *Vector sequence transformations: Methodology and applications to linear systems*, J. Comput. Appl. Math., 98 (1998), pp. 149–175.

[11] C. Brezinski, *Biorthogonal vector sequence transformations and Padé approximation of vector series*, Appl. Numer. Math., 41 (2002), pp. 437–442.

[12] C. Brezinski and M. Crouzeix, *Remarques sur le procédé Δ^2 d'Aitken*, C. R. Acad. Sci. Paris Sér. A-B, 270 (1970), pp. 896–898.

[13] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods. Theory and Practice*, North-Holland, Amsterdam, 1991.

[14] C. Brezinski and M. Redivo-Zaglia, *Vector and matrix sequence transformations based on biorthogonality*, Appl. Numer. Math., 21 (1996), pp. 353–373.

[15] C. Brezinski and M. Redivo-Zaglia, *New vector sequence transformations*, Linear Algebra Appl., 389 (2004), pp. 189–213.

[16] C. Brezinski and M. Redivo-Zaglia, *Rational Extrapolation of the PageRank Vectors*, in progress.

[17] C. Brezinski, M. Redivo-Zaglia, and S. Serra-Capizzano, *Extrapolation methods for PageRank computations*, C. R. Math. Acad. Sci. Paris, 340 (2005), pp. 393–397.

[18] S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks ISDN Syst., 30 (1998), pp. 107–117.

[19] G. M. Del Corso, A. Gulli, and F. Romani, *Fast PageRank computation via a sparse linear system*, in Algorithms and Models for the Web-Graph, S. Leonardi, ed., Lecture Notes in Comput. Sci. 3243, Springer-Verlag, New York, 2004, pp. 118–130.

[20] L. Eldén, *The Eigenvalues of the Google Matrix*, Report LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, Sweden, 2003.

[21] F. R. Gantmacher, *The Theory of Matrices*, Vol. 1, Chelsea, New York, 1959.

[22] D. Gleich, L. Zhukov, and P. Berkhin, *Fast Parallel PageRank: A Linear System Approach*, Technical report, Yahoo! Inc., 2004.

[23] G. H. Golub and C. Greif, *Arnoldi-Type Algorithms for Computing Stationary Distribution Vectors, with Application to PageRank*, Technical report SCCM-04-15, Stanford University, Stanford, CA, 2004.

[24] T. H. Haveliwala and S. D. Kamvar, *The Second Eigenvalue of the Google Matrix*, Technical report, Stanford University, Stanford, CA, 2003; available online from http://dbpubs.stanford.edu/pub/2003-20.

[25] I. C. F. Ipsen and S. Kirkland, *Convergence analysis of a PageRank updating algorithm by Langville and Meyer*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 952–967.

[26] 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.

[27] K. Jbilou, *A general projection algorithm for solving systems of linear equations*, Numer. Algorithms, 4 (1993), pp. 361–377.

[28] S. D. Kamvar and T. H. Haveliwala, *The Condition Number of the PageRank Problem*, Technical report, Stanford University, Stanford, CA, 2003; available online from http://dbpubs.stanford.edu/pub/2003-36.

[29] S. Kamvar, T. Haveliwala, and G. Golub, *Adaptive methods for the computation of PageRank*, Linear Algebra Appl., 386 (2004), pp. 51–65.

[30] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Exploiting the Block Structure of the Web for Computing PageRank*, Technical report, Stanford University, Stanford, CA, 2003; available online from http://dbpubs.stanford.edu/pub/2003-17.

[31] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolations methods for accelerating PageRank computations*, in Proceedings of the Twelfth International World Wide Web Conference, ACM Press, New York, 2003, pp. 261–270.

[32] G. Kollias, E. Gallopoulos, and D. B. Szyld, *Asynchronous iterative computations with Web information retrieval structures: The PageRank case*, in Proceedings of the Conference on Parallel Computing 2005, Malaga, Spain, 2005.

[33] A. N. Langville and C. D. Meyer, *The use of linear algebra by web search engines*, IMAGE Newsletter, 33 (2004), pp. 2–6.

[34] A. N. Langville and C. D. Meyer, *Deeper inside PageRank*, Internet Math., 1 (2005), pp. 335–400.

[35] A. N. Langville and C. D. Meyer, *Updating PageRank with iterative aggregation*, in Proceedings of the Thirteenth World Wide Web Conference, ACM Press, New York, 2004, pp. 392–393.

[36] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.

[37] C. P.-C. Lee, G. H. Golub, and S. A. Zenios, *A Fast Two-Stage Algorithm for Computing PageRank and Its Extensions*, Technical report SCCM-2003-15, Scientific Computation and Computational Mathematics, Stanford University, Stanford, CA, 2003; available online from http://www-sccm.stanford.edu/nf-publications-tech.html.

[38] A. C. Matos, *Acceleration results for the vector E-algorithm*, Numer. Algorithms, 1 (1991), pp. 237–260.

[39] A. C. Matos, *Convergence and acceleration properties for the vector ϵ-algorithm*, Numer. Algorithms, 3 (1992), pp. 313–320.

[40] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[41] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical report, Stanford University, Stanford, CA, 1999; available online from http://dbpubs.stanford.edu/pub/1999-66.

[42] B. P. Pugachev, *Acceleration of convergence of iterative processes and a method of solving systems of nonlinear equations*, USSR Comput. Maths. Maths. Phys., 17 (1978), pp. 199–207.

[43] M. Redivo-Zaglia, *Pseudo-Schur complements and their properties*, Appl. Numer. Math., 50 (2004), pp. 511–519.

[44] 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.

[45] A. Sidi, *On a generalization of the Richardson extrapolation process*, Numer. Math., 57 (1990), pp. 365–377.

[46] A. Sidi, *Approximation of Largest Eigenpairs of Matrices and Applications to PageRank Computation*, Technical report CS-2004-16, Computer Science Department, Technion, Haifa, Israel, 2004.

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

[48] J. Van Iseghem, *Vector Padé approximants*, in Numerical Mathematics and Applications, R. Vichnevetsky and J. Vignes, eds., North-Holland, Amsterdam, 1986, pp. 73–77.

[49] R. S. Varga, *Matrix Iterative Analysis*, 2nd ed., Springer, New York, 2000.

[50] Yu. V. Vorobyev, *Method of Moments in Applied Mathematics*, Gordon and Breach, New York, 1965.

[51] P. Wynn, *On the convergence and stability of the epsilon algorithm*, SIAM J. Numer. Anal., 3 (1966), pp. 91–122.

[52] F.-Z. Zhang, ed., *The Schur Complement and Its Applications*, Springer, New York, 2005.