

RATIONAL EXTRAPOLATION FOR THE PAGERANK VECTOR

C. BREZINSKI AND M. REDIVO-ZAGLIA

ABSTRACT. An important problem in web search is to determine the importance of each page. From the mathematical point of view, this problem consists in finding the nonnegative left eigenvector of a matrix corresponding to its dominant eigenvalue 1. Since this matrix is neither stochastic nor irreducible, the power method has convergence problems. So, the matrix is replaced by a convex combination, depending on a parameter c , with a rank one matrix. Its left principal eigenvector now depends on c , and it is the PageRank vector we are looking for. However, when c is close to 1, the problem is ill-conditioned, and the power method converges slowly. So, the idea developed in this paper consists in computing the PageRank vector for several values of c , and then to extrapolate them, by a conveniently chosen rational function, at a point near 1. The choice of this extrapolating function is based on the mathematical expression of the PageRank vector as a function of c . Numerical experiments end the paper.

1. THE PROBLEM

The mathematical problem behind web search is the computation of the nonnegative left eigenvector of a $p \times p$ matrix P corresponding to its dominant eigenvalue 1, where p is the number of pages in Google (8.06 billion at the end of March 2005). Since P is not stochastic (some rows of P may contain only zeros due to the so-called dangling nodes), it is replaced by the matrix

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

with $\mathbf{w} \in \mathbb{R}^p$ a probability vector, that is, such that $\mathbf{w} \geq 0$ and $(\mathbf{w}, \mathbf{e}) = 1$ with $\mathbf{e} = (1, \dots, 1)^T$, and $\mathbf{d} = (d_i) \in \mathbb{R}^p$ the vector with $d_i = 1$ if $\deg(i) = 0$, and 0 otherwise, where $\deg(i)$ is the outdegree of the page i , that is, the number of pages it points to.

Since the matrix \tilde{P} is not irreducible, it is replaced by the matrix

$$P_c = c\tilde{P} + (1 - c)E,$$

where c is a parameter between 0 and 1, and $E = \mathbf{e}\mathbf{v}^T$ with $\mathbf{e} = (1, \dots, 1)^T \in \mathbb{R}^p$ and \mathbf{v} is a probability vector. Such a modification of the matrix 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

Received by the editor January 23, 2007 and, in revised form, June 27, 2007.

2000 *Mathematics Subject Classification.* Primary 65B05, 65F15, 68U35.

Key words and phrases. Extrapolation, PageRank, web matrix, eigenvector computation.

The work of the second author was supported by MIUR under the PRIN grant no. 2006017542-003.

of pages. For that reason, \mathbf{v}^T is called the *personalization* vector. The matrix P_c is stochastic and irreducible, and $P_c \mathbf{e} = \mathbf{e}$.

The unique nonnegative dominant left eigenvector of P_c is denoted \mathbf{r}_c . Thus, $\mathbf{r}_c = P_c^T \mathbf{r}_c$. This vector can be computed by the power method, which consists of the iterations

$$\mathbf{r}_c^{(n+1)} = P_c^T \mathbf{r}_c^{(n)}, \quad n = 0, 1, \dots$$

with $\mathbf{r}_c^{(0)} = \mathbf{v}$. These iterations converge to \mathbf{r}_c as c^n , and originally Google chooses $c = 0.85$, which insures a good rate of convergence. Anyway, since the computation of the PageRank vector can take several days, various methods for their acceleration have been proposed [12, 5].

The vector $\tilde{\mathbf{r}} = \lim_{c \rightarrow 1} \mathbf{r}_c$ is uniquely determined as the limit, when c tends to 1, of the family of vectors \mathbf{r}_c . However, it is just one of the infinitely many solutions of $P^T \mathbf{r} = \mathbf{r}$, $\mathbf{r} \geq 0$, $(\mathbf{r}, \mathbf{e}) = 1$, which form a nontrivial convex set. Notice that the conditioning of the matrix P_c grows as $(1 - c)^{-1}$, but that the function \mathbf{r}_c is analytic in a small neighbourhood of 1 in the complex plane [10]. For a detailed analysis of the sensitivity of the vector \mathbf{r}_c , see [21]. We recommend [15] for detailed explanations about the origin, the mathematical properties, and the treatment of the PageRank problem.

An idea for obtaining approximations of $\lim_{c \rightarrow 1} \mathbf{r}_c$ is to compute the vector \mathbf{r}_c for different values of c away from 1, to interpolate them by some vector function, and finally to extrapolate this function at the point $c = 1$, or at any other point close to 1. Of course, in order to obtain good results, the interpolating function has to mimic as closely as possible the exact behavior of \mathbf{r}_c with respect to c . This behavior was analyzed in [10], where the following results are proved (see also [20]).

Theorem 1.1. *Let $\mathbf{e}, \mathbf{x}_2, \dots, \mathbf{x}_p$ be the right eigenvectors of the matrix \tilde{P} , and $\mathbf{y}, \mathbf{y}_2, \dots, \mathbf{y}_p$ its left eigenvectors corresponding to the eigenvalues $1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_p$ with $1 \geq |\tilde{\lambda}_2| \geq \dots \geq |\tilde{\lambda}_p|$.*

If \tilde{P} is diagonalizable, then

$$(1.1) \quad \mathbf{r}_c = \mathbf{y} + (1 - c) \sum_{i=2}^p \frac{\alpha_i}{1 - c\tilde{\lambda}_i} \mathbf{y}_i,$$

with $\alpha_i = \mathbf{v}^T \mathbf{x}_i$, where \mathbf{y} is one of the PAGERANK vectors (i.e. corresponding to $c = 1$).

In the general case,

$$(1.2) \quad \mathbf{r}_c = \mathbf{y} + \sum_{i=2}^p w_i(c) \mathbf{y}_i$$

with

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

and β_i equal to 0 or 1.

Furthermore, if the eigenvalue 1 has multiplicity m (1 has to be semisimple), then $\beta_i = 0$, i.e. $w_i(c) = \alpha_i$ for $i = 2, \dots, m$, and therefore

$$(1.3) \quad \mathbf{r}_c = \tilde{\mathbf{r}} + \sum_{i=m+1}^p w_i(c) \mathbf{y}_i$$

where

$$\tilde{\mathbf{r}} = \mathbf{y} + \sum_{i=2}^m \alpha_i \mathbf{y}_i = N\mathbf{v},$$

with

$$N = [\mathbf{y}, \mathbf{y}_2, \dots, \mathbf{y}_m] [\mathbf{e}, \mathbf{x}_2, \dots, \mathbf{x}_m]^T$$

a nonnegative projector, i.e. $N = N^2$ with nonnegative entries.

Notice that formula (1.3) is well defined when c tends to 1, and indeed \mathbf{r}_c can be extended in an analytic way at this point as explained above. However the vector $\tilde{\mathbf{r}}$ at $c = 1$ coincides with \mathbf{y} only if the web is made by a unique irreducible component (which is not the case). In the general case, the null space of $I - \tilde{P}$ has dimension $m \gg 1$, and $\tilde{\mathbf{r}} \geq 0$ is a linear combination of a basis of this null space; its characterization can also be given in terms of a Cesaro operator [21].

According to (1.1) or (1.2), \mathbf{r}_c is a vector rational function with a vector numerator and a scalar denominator of degree $p - 1$ in c . We also see that \mathbf{r}_c tends to $\tilde{\mathbf{r}}$ when c tends to 1.

Extrapolation methods following this analysis were given in [7]. As explained above, the idea is to compute several vectors \mathbf{r}_c for various values of c , and to interpolate them by a vector rational function of the same form as (1.1) or (1.2) but with a much smaller degree $k \leq p - 1$, and then to compute this rational function at a point outside the interval containing the values of c used before ($c = 0.85$, or $c = 1$, or any other value of c close to 1).

Although, in our extrapolation procedures, the vector \mathbf{r}_c has to be computed for different values of the parameter c , it is very important to notice that the power method does not need to be restarted for each value of c . The total number of iterations needed by our procedures is the one required for the highest value of c , and no additional iteration is needed. Indeed, as observed in [2] (see also Properties 8 and 9 of [5]), it holds

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

where $\tilde{A} = \tilde{P}^T$. This relation shows that it is possible to apply the power method simultaneously (that is, in the same loop of the program) for several values of c with only a small additional cost. Indeed, since $(\tilde{A} - I)\tilde{A}^n \mathbf{v}$ is independent of c , the vectors $\mathbf{r}_c^{(n)}$ corresponding to \tilde{c} can be directly obtained by

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

We will now discuss such extrapolation procedures in more detail, give new ones, and report some numerical experiments.

2. VECTOR RATIONAL EXTRAPOLATION

Let us describe in more detail an algorithm for vector rational extrapolation which was first given in [7] and discuss some of its properties.

We begin by interpolating the vectors $\mathbf{r}_c \in \mathbb{R}^p$ corresponding to several values of the parameter c by the vector rational function

$$(2.1) \quad \mathbf{p}(c) = \frac{\mathbf{P}_k(c)}{Q_k(c)},$$

where \mathbf{P}_k and Q_k are polynomials of degree $k \leq p-1$. The coefficients of \mathbf{P}_k are vectors, while those of Q_k are scalars. Then, an approximate value of \mathbf{r}_c , for an arbitrary value of c (in general outside the interval containing the interpolation points, thus the name of the procedure) will be given by $\mathbf{p}(c)$.

Following an idea introduced in [6], the coefficients of \mathbf{P}_k and Q_k are obtained by solving the interpolation problem

$$(2.2) \quad Q_k(c_i)\mathbf{p}_i = \mathbf{P}_k(c_i), \quad i = 0, \dots, k,$$

with $\mathbf{p}_i = \mathbf{r}_{c_i}$, and the c_i 's distinct points in $]0, 1[$.

The polynomials \mathbf{P}_k and Q_k are given by the Lagrange interpolation formula

$$(2.3) \quad \begin{aligned} \mathbf{P}_k(c) &= \sum_{i=0}^k L_i(c)\mathbf{P}_k(c_i), \\ Q_k(c) &= \sum_{i=0}^k L_i(c)Q_k(c_i) \end{aligned}$$

with

$$L_i(c) = \prod_{\substack{j=0 \\ j \neq i}}^k \frac{c - c_j}{c_i - c_j}, \quad i = 0, \dots, k.$$

Thus, from (2.2),

$$(2.4) \quad \mathbf{P}_k(c) = \sum_{i=0}^k L_i(c)Q_k(c_i)\mathbf{p}_i.$$

Let us now show how to compute $Q_k(c_0), \dots, Q_k(c_k)$. We assume that, for $c^* \neq c_i$, $i = 0, \dots, k$, the vector \mathbf{r}_{c^*} is known. Following (2.1) and (2.4), we will approximate it by

$$(2.5) \quad \mathbf{p}(c^*) = \sum_{i=0}^k L_i(c^*)a_i(c^*)\mathbf{p}_i,$$

with $a_i(c^*) = Q_k(c_i)/Q_k(c^*)$.

Let $\mathbf{s}_0, \dots, \mathbf{s}_k$ be $k+1$ linearly independent vectors. After taking their scalar products with the vector $\mathbf{p}(c^*)$, given by (2.5), and with the vector \mathbf{r}_{c^*} , we will look for $a_0(c^*), \dots, a_k(c^*)$ solutions of the system of $k+1$ linear equations

$$(2.6) \quad \sum_{i=0}^k (\mathbf{p}_i, \mathbf{s}_j) L_i(c^*) a_i(c^*) = (\mathbf{r}_{c^*}, \mathbf{s}_j), \quad j = 0, \dots, k.$$

Instead of considering (2.6) as a linear system in the unknowns $a_0(c^*), \dots, a_k(c^*)$, we can consider it as a system in the unknowns $u_i(c^*) = L_i(c^*)a_i(c^*)$, $i = 0, \dots, k$. Since the $L_i(c^*)$'s are known quantities, the $a_i(c^*)$ will be immediately deduced.

Let $M = [\mathbf{p}_0, \dots, \mathbf{p}_k]$, $S = [\mathbf{s}_0, \dots, \mathbf{s}_k]$, $\mathbf{a}(c^*) = (a_0(c^*), \dots, a_k(c^*))^T$, and

$$L(c) = \begin{pmatrix} L_0(c) & & \\ & \ddots & \\ & & L_k(c) \end{pmatrix}.$$

By setting $\mathbf{u}(c^*) = (L_0(c^*)a_0(c^*), \dots, L_k(c^*)a_k(c^*))^T = L(c^*)\mathbf{a}(c^*)$, (2.6) can be written as

$$(2.7) \quad S^T M \mathbf{u}(c^*) = S^T \mathbf{r}_{c^*},$$

and it follows that

$$(2.8) \quad \mathbf{p}(c^*) = M \mathbf{u}(c^*) = M(S^T M)^{-1} S^T \mathbf{r}_{c^*},$$

which shows that $\mathbf{p}(c^*)$ is the oblique projection of \mathbf{r}_{c^*} on $\text{span}(\mathbf{p}_0, \dots, \mathbf{p}_k)$ along $[\text{span}(\mathbf{s}_0, \dots, \mathbf{s}_k)]^\perp$ that is orthogonal to $\text{span}(\mathbf{s}_0, \dots, \mathbf{s}_k)$. This vector can be represented as a ratio of determinants; see [8, 4].

For the particular choice $\mathbf{s}_j = \mathbf{p}_j$, $j = 0, \dots, k$, the system (2.7) has a symmetric positive definite Gram matrix, and $\mathbf{p}(c^*) = \sum_{i=0}^k L_i(c^*)a_i(c^*)\mathbf{p}_i$ is the best approximation of \mathbf{r}_{c^*} in $\text{span}(\mathbf{p}_0, \dots, \mathbf{p}_k)$. In that case,

$$(2.9) \quad \mathbf{p}(c^*) = M(M^T M)^{-1} M^T \mathbf{r}_{c^*}$$

is the orthogonal projection of \mathbf{r}_{c^*} on $\text{span}(\mathbf{p}_0, \dots, \mathbf{p}_k)$. The matrix $(M^T M)^{-1} M^T$ is the pseudo-inverse M^\dagger of M , and it satisfies $M^\dagger M = I$.

Remark 2.1. Let us remark that $k' + 1$ linearly independent vectors \mathbf{s}_j could be used with $k' > k$, in which case the system (2.6) has to be solved in the least squares sense.

Once the $a_i(c^*)$'s have been obtained, the $Q_k(c_i)$'s could be computed. For that, it is necessary to know the value of $Q_k(c^*)$. Since a rational function is determined apart from a multiplying factor, it does not restrict the generality to assume that the polynomial Q_k is monic. So, from (2.3), we see that its dominant coefficient satisfies the relation

$$1 = \sum_{i=0}^k \frac{Q_k(c_i)}{\prod_{\substack{j=0 \\ j \neq i}}^k (c_i - c_j)} = Q_k(c^*) \sum_{i=0}^k \frac{a_i(c^*)}{\prod_{\substack{j=0 \\ j \neq i}}^k (c_i - c_j)},$$

which gives $Q_k(c^*)$. Then, $Q_k(c_i) = a_i(c^*)Q_k(c^*)$ for $i = 0, \dots, k$. But, as we will see now, it is even unnecessary to know the $Q_k(c_i)$'s and $Q_k(c^*)$.

Indeed, for an arbitrary value of c , we obtain an approximation of \mathbf{r}_c as

$$\mathbf{p}(c) = \frac{\mathbf{p}_k(c)}{Q_k(c)} = \frac{\sum_{i=0}^k L_i(c)Q_k(c_i)\mathbf{p}_i}{\sum_{i=0}^k L_i(c)Q_k(c_i)}.$$

Dividing the numerator and the denominator by $Q_k(c^*)$ finally leads to the extrapolation formula

$$(2.10) \quad \mathbf{p}(c) = \frac{\sum_{i=0}^k L_i(c)a_i(c^*)\mathbf{p}_i}{\sum_{i=0}^k L_i(c)a_i(c^*)}.$$

When $c = c^*$, it follows from (2.3) that the denominator in this formula is equal to 1. Indeed, for $c = c^*$, (2.3) gives

$$1 = \frac{1}{Q_k(c^*)} \sum_{i=0}^k L_i(c^*) Q_k(c_i) = \sum_{i=0}^k L_i(c^*) a_i(c^*).$$

From Formula (2.10), it is easy to see that $\mathbf{p}(c_j) = \mathbf{p}_j$ for $j = 0, \dots, k$, and that, in general, $\mathbf{p}(c^*) \neq \mathbf{r}_{c^*}$. Moreover, since for all i , $(\mathbf{p}_i, \mathbf{e}) = 1$, it follows that, for all c , $(\mathbf{p}(c), \mathbf{e}) = 1$.

When $k = p - 1$, the matrices S and M are square and nonsingular and (2.8) shows that, in this case, $\mathbf{p}(c^*) = \mathbf{r}_{c^*}$. This vector condition corresponds to p scalar ones, as well as to those of the system (2.6). Moreover, the p vector interpolation conditions (2.2) correspond to p^2 scalar ones. Thus, in total, we obtain $p^2 + p$ conditions. On the other side, when $k = p - 1$, we have p unknowns vectors of dimension p in the polynomial \mathbf{P}_k , and p unknown scalar coefficients in Q_k . Thus, by a uniqueness argument, it follows that, for all c , $\mathbf{p}(c) = \mathbf{r}_c$.

A more compact formula can be obtained as follows. Since $\mathbf{u}(c^*) = L(c^*)\mathbf{a}(c^*)$, (2.7) becomes

$$S^T M L(c^*) \mathbf{a}(c^*) = S^T \mathbf{r}_{c^*},$$

and (2.10) can be written as

$$(2.11) \quad \mathbf{p}(c) = \frac{ML(c)\mathbf{a}(c^*)}{(L(c)\mathbf{a}(c^*), \mathbf{e}')},$$

with $\mathbf{a}(c^*) = [L(c^*)]^{-1} \mathbf{u}(c^*)$ and $\mathbf{e}' = (1, \dots, 1)^T \in \mathbb{R}^{k+1}$.

We see that the computation of $\mathbf{p}(c)$ by our extrapolation method needs the knowledge of \mathbf{r}_c for $k + 2$ distinct values of c , namely c_0, \dots, c_k and c^* .

The complete vector rational extrapolation procedure is as follows.

- (1) Choose $k + 2$ distinct values of c : c_0, \dots, c_k and c^* .
- (2) Compute $\mathbf{p}_i = \mathbf{r}_{c_i}$ for $i = 0, \dots, k$, and \mathbf{r}_{c^*} .
- (3) Choose $k + 1$ linearly independent vectors $\mathbf{s}_0, \dots, \mathbf{s}_k$, or take $\mathbf{s}_i = \mathbf{p}_i$ for $i = 0, \dots, k$.
- (4) Solve the system (2.6), or the system (2.7), and compute the unknowns $a_0(c^*), \dots, a_k(c^*)$.
- (5) Compute an approximation of \mathbf{r}_c by (2.10) or (2.11).

It must be noticed that, since $(\mathbf{p}_i, \mathbf{e}) = 1$, then, $\forall c$, $(\mathbf{p}(c), \mathbf{e}) = 1$. If $L_i(c)a_i(c^*) \geq 0$ for $i = 0, \dots, k$, then the vector $\mathbf{p}(c)$ will have a unit l_1 -norm since the components of the vectors \mathbf{p}_i will be nonnegative. It is an open question to check if this condition is satisfied or not.

3. A SIMPLER VECTOR RATIONAL EXTRAPOLATION

Let us consider a vector rational extrapolation method where the extrapolating function has the same form as (1.1) or (1.2), but with only the first two terms (which corresponds to taking $p = 2$ in these formulae). Therefore, in both cases, we will consider an extrapolation function of the form

$$(3.1) \quad \mathbf{p}(c) = \tilde{\mathbf{y}} + (1 - c) \frac{1}{1 - c\lambda} \mathbf{z},$$

where $\tilde{\mathbf{y}}$ is an approximation of $\tilde{\mathbf{r}}$, \mathbf{z} is an approximation of $\alpha_2 \mathbf{y}_2$, and λ is an approximation of $\tilde{\lambda}_2$. These two unknown vectors and the unknown scalar will be computed by an interpolation procedure needing only 3 values of c .

As above, let $\mathbf{p}_i = \mathbf{r}_{c_i}$, and let the c_i 's be distinct values in $]0, 1[$. We consider the interpolation condition

$$\mathbf{p}_i = \tilde{\mathbf{y}} + \frac{1 - c_i}{1 - c_i \lambda} \mathbf{z}.$$

The difference $\mathbf{p}_i - \mathbf{p}_j$ eliminates $\tilde{\mathbf{y}}$, and we have

$$\mathbf{p}_i - \mathbf{p}_j = \frac{(c_j - c_i)(1 - \lambda)}{(1 - c_i \lambda)(1 - c_j \lambda)} \mathbf{z}.$$

We now need to compute the scalar λ and the vector \mathbf{z} . Let \mathbf{q} be a vector so that the scalar products $(\mathbf{p}_i - \mathbf{p}_j, \mathbf{q})$ and $(\mathbf{p}_k - \mathbf{p}_j, \mathbf{q})$ are different from zero. We set

$$r_{ijk} = \frac{(\mathbf{p}_i - \mathbf{p}_j, \mathbf{q})}{(\mathbf{p}_k - \mathbf{p}_j, \mathbf{q})} = \frac{c_j - c_i}{c_j - c_k} \frac{1 - c_k \lambda}{1 - c_i \lambda},$$

which gives

$$(3.2) \quad \lambda = \frac{r_{ijk}(c_j - c_k) - (c_j - c_i)}{c_i r_{ijk}(c_j - c_k) - c_k(c_j - c_i)}.$$

Then \mathbf{z} takes the form

$$(3.3) \quad \mathbf{z} = \frac{(1 - c_i \lambda)(1 - c_j \lambda)}{(c_j - c_i)(1 - \lambda)} (\mathbf{p}_i - \mathbf{p}_j).$$

Finally, \mathbf{y} is given by

$$(3.4) \quad \mathbf{y} = \mathbf{p}(1) = \mathbf{p}_i - \frac{1 - c_i}{1 - c_i \lambda} \mathbf{z}.$$

Thus, from the expressions (3.2), (3.3), and (3.4), Formula (3.1) leads to the rational vector extrapolation procedure (3.1), that is, $\mathbf{p}(c) \simeq \mathbf{r}_c$.

Since $(\mathbf{p}_i, \mathbf{e}) = 1$, then $(\mathbf{p}(c), \mathbf{e}) = 1$.

4. A MINIMIZATION PROCEDURE

Any scalar combination of different vectors $\mathbf{p}_i = \mathbf{r}_{c_i}$ can be considered as an extrapolation procedure (indeed, compare with (2.10)). So, we will now build an approximation $\mathbf{p}(c)$ of \mathbf{r}_c of the form

$$\mathbf{p}(c) = (1 - \alpha)\mathbf{p}_0 + \alpha\mathbf{p}_1 = \mathbf{p}_0 + \alpha(\mathbf{p}_1 - \mathbf{p}_0),$$

where the parameter α is chosen such that the Euclidean norm of the vector $P_c^T \mathbf{p}(c) - \mathbf{p}(c)$ is minimum. Remark that, since $P_c^T \mathbf{r}_c = \mathbf{r}_c$, this vector could be interpreted as a kind of residual.

It holds

$$P_c^T \mathbf{p}(c) - \mathbf{p}(c) = (P_c^T \mathbf{p}_0 - \mathbf{p}_0) + \alpha[P_c^T (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0)].$$

Thus

$$\begin{aligned} \|P_c^T \mathbf{p}(c) - \mathbf{p}(c)\|^2 &= \|P_c^T \mathbf{p}_0 - \mathbf{p}_0\|^2 + 2\alpha(P_c^T (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0), P_c^T \mathbf{p}_0 - \mathbf{p}_0) \\ &\quad + \alpha^2 \|P_c^T (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0)\|^2, \end{aligned}$$

and the α which minimizes $\|P_c^T \mathbf{p}(c) - \mathbf{p}(c)\|$ is

$$\alpha = -\frac{(P_c^T (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0), P_c^T \mathbf{p}_0 - \mathbf{p}_0)}{\|P_c^T (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0)\|^2}.$$

Let us mention that the products $P_c^T \mathbf{p}_i$ are cheap and easy to compute [5, 12, 15], and only two of them are required in this procedure.

Obviously this strategy could be extended to a more general form of minimization where

$$\mathbf{p}(c) = \alpha_0 \mathbf{p}_0 + \cdots + \alpha_k \mathbf{p}_k \quad \text{with} \quad \alpha_0 + \cdots + \alpha_k = 1.$$

5. NUMERICAL EXPERIMENTS

For conducting our numerical experiments, we construct several matrices $P = (p_{ij})$ in a random way as follows. First we select a random integer pm (with a uniform distribution) between 1 and $p/10$, where p is the dimension of P . Then, we generate a random integer vector \mathbf{m} of dimension p with components between 1 and pm . Each row i of the matrix P will contain, at most, $m(i)$ nonzero elements. Then, we randomly choose, for each row i , an integer vector of dimension $m(i)$, and we eliminate its identical components. The length of this vector will be the outdegree of the page i , that is, $\deg(i)$, and its components will give the indices j of the columns such that $p_{ij} = 1/\deg(i)$, all other elements being set to zero. Let us mention that it is possible to set $p_{ii} = 0$ for avoiding self-hyperlinks, an option we did not choose. Finally, among all rows, we randomly set to zero $p/5$ of them, corresponding to the dangling nodes. Thus, by construction, such matrices P (and the corresponding matrices \tilde{P} and P_c) have the same mathematical properties as those coming out from the web. Moreover, they present the advantages that their dimension, their sparsity, and their number of dangling pages can be adjusted according to the user choice.

A very important point to mention is that, in fact, we are not interested in the exact values of the components of the real and extrapolated PageRank vectors, but in their relative values, that is, in the rank of each of them compared with the other components. But, as noticed in [16], the values assigned to pages can be quite sensitive due to the stability of the PageRank algorithm. Then, the ranks could also be quite volatile with respect to changes in the matrix; this is the notion of rank-stability, first defined and studied in [3]. Moreover, stability of the PageRank algorithm does not imply rank-stability, and a change in the outlink of a page of small rank can completely change the whole ranking.

The computations were performed with a 1.73 GHz Pentium M processor, using MATLAB 7.X.

In the following tables, rows are numbered. Our methods will be denoted as VREM for the vector rational extrapolation method of Section 2, SVREM for the simpler vector rational extrapolation method given in Section 3, and VMP for the vector minimisation procedure of Section 4.

The first column of data gives the name of the extrapolation method, and the number of vectors it needs. For VREM, this number is $k+2$, and each row contains the results obtained with the first $k+1$ values in the list of the c_i 's, and with the same c^* . For SVREM and VMP the number of vectors is always 3 and 2, respectively.

The second column corresponds to the ∞ -norm of the difference between the PageRank vector \mathbf{r}_c computed by the power method with a precision of 10^{-8} , and the vector \mathbf{p} obtained by one of our procedures.

In the third column, we give the 1-norm of the same difference divided by the dimension p of the problem. This quantity represents the mean value of the absolute values of the error. We can also compute its standard deviation as

$\sigma = [(\|\mathbf{r}_c - \mathbf{p}\|_2^2 - pE^2)/(p-1)]^{1/2}$, with $E = \|\mathbf{r}_c - \mathbf{p}\|_1/p$. All our numerical experiments show that the standard deviation is quite small, thus showing that the errors are concentrated around their mean value E . Let us mention that the improvement brought by our extrapolation procedures was also checked by using the Kendall τ rank correlation coefficient [13].

The fourth column (*nch*) is the total number of changes in the ranking of the components between the PageRank vector and the vector obtained by extrapolation.

The fifth column (*ich*), after sorting \mathbf{r}_c and \mathbf{p} by descending values, is the rank where the first change occurs.

In column 6, d_{\max} is the maximum displacement of a page (in absolute value), and the rank where it occurs (before sorting) is given in column 7 (*pos*). A positive value of d_{\max} means that the *pos*-th component of \mathbf{r}_c went up in the list, and that it went down if it is negative.

The last two columns show the ranks of the page corresponding to this maximum change in the sorted PageRank vector (ix_{\max}), and in the sorted extrapolated one (iy_{\max}), respectively.

For instance, in the first row of Table 1, we test the VREM procedure with the first 3 values of c_i (0.1, 0.15 and 0.2), and $c^* = 0.5$; we have a total of 4417 pages for which the rank changes, but the first 17 ($ich - 1$) pages having the higher rank are correctly detected; the maximum displacement $d_{\max} = -47$ occurs for the component $pos = 4365$ of the vector \mathbf{r}_c , and the corresponding page has the rank $ix_{\max} = 1553$ in the sorted vector \mathbf{r}_c , while it has the rank $iy_{\max} = 1600$ in the sorted vector $\mathbf{p}(c)$. So, it went down by 47 positions.

The first 6 rows in all tables (except for Table 4) correspond to the VREM applied by using an increasing number of vectors, starting from 4. In the second part of the table, the first row (# 7) shows again the results obtained by the VREM with 4 vectors (as row #1) but using the last 3 values of c_i (that is, c_5, c_6, c_7 , instead of the first 3 values), and the same value of c^* . The row # 8 provides the results for the SVREM by using the same 3 c_i 's, and $\mathbf{q} = \mathbf{p}_7 - \mathbf{p}_5$. The VPM (row # 9) uses only \mathbf{p}_5 and \mathbf{p}_7 .

All results were conducted with $\mathbf{w} = \mathbf{v}$.

We begin by a matrix of dimension $p = 5000$ containing 942806 nonzero elements. The parameter c was fixed to 0.85, and 8 iterations were needed by the power method to achieve a precision of 10^{-8} . This vector (denoted by \mathbf{r}_c below) is taken as the PageRank vector. The highest and the smallest components of the PageRank vector were $3.84636884 \cdot 10^{-4}$ and $1.4882646 \cdot 10^{-4}$, respectively, thus meaning that, when p is very large, many components can differ only in the last digits. The values of the c_i used in the vector rational extrapolation method (denoted below as VREM) of Section 2 were 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, and $c^* = 0.5$. Only 5 iterations of the power method were needed to obtain the results with the highest value of the parameter, that is, c^* .

Let us discuss how to judge the quality of these results. It seems that the two most important parameters to consider are d_{\max} and *ich*. The parameter d_{\max} indicates the size of the largest change in the ranking. The smaller d_{\max} , the better the ranking. Thus, a criterion of good quality is to have a small value of d_{\max} . But d_{\max} can be large if *ich* is also large. The parameter *ich* indicates the location of the first change in the ranking. So, a correct ranking has been obtained for the $ich - 1$

TABLE 1. $p = 5000, c_i = 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, c^* = 0.5, c = 0.85$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	2.43e-6	2.57e-8	4417	18	-47	4365	1553	1600
2	VREM 5	5.13e-8	3.32e-9	1667	29	6	3223	2651	2645
3	VREM 6	6.03e-8	2.34e-9	1254	190	-4	2725	2358	2362
4	VREM 7	2.77e-8	1.24e-9	689	190	2	2216	890	888
5	VREM 8	3.04e-8	1.89e-9	1029	190	-4	2725	2358	2362
6	VREM 9	2.87e-8	1.74e-9	939	190	4	1378	2765	2761
7	VREM 4	6.86e-7	1.04e-8	3479	18	12	3384	2691	2679
8	SVREM 3	1.17e-5	9.68e-8	4863	1	-234	4436	3529	3763
9	VMP 2	1.20e-5	7.88e-8	4865	1	-236	4436	3529	3765

first components of the vector obtained by extrapolation. It is not important to have many changes (nch large) in the ranking if they are small, that is, if d_{\max} is small.

For example, in Table 1, it seems that the best results, according to the previous criteria, have been obtained with VREM 7 (see the values marked by a grey background), which leads to $nch = 689$ changes, with a largest change in the ranking of the pages of only $d_{\max} = 2$. Thus, our extrapolation procedures only produce small changes (1 or 2) in the ranking given by the power method.

TABLE 2. $p = 5000, c_i = 0.1, 0.15, 0.2, 0.5, 0.3, 0.35, 0.4, 0.45, c^* = 0.25, c = 0.85$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	2.43e-6	2.57e-8	4420	18	-47	4365	1553	1600
2	VREM 5	4.31e-8	2.01e-9	1102	190	-4	2725	2358	2362
3	VREM 6	3.07e-8	1.86e-9	1010	190	-4	2725	2358	2362
4	VREM 7	2.50e-8	1.54e-9	827	190	4	1378	2765	2761
5	VREM 8	3.10e-8	1.90e-9	1033	190	-4	2725	2358	2362
6	VREM 9	9.59e-7	2.71e-8	4520	10	-57	158	2710	2767
7	VREM 4	3.53e-5	2.25e-6	4808	10	127	1460	2362	2235
8	SVREM 3	1.17e-5	9.68e-8	4863	1	-234	4436	3529	3763
9	VMP 2	1.20e-5	7.88e-8	4865	1	-236	4436	3529	3765

In Table 2, we give the results obtained by using the same matrix and the same values as in Table 1, but interchanging c_3 and c^* . We remark that rows # 1 are the same in both tables. They use the same values for c_0, c_1 and c_2 , the only difference being the value of c^* . We notice that the errors of the VREM decrease when k increases, but that, for $k + 2 = 9$ (row # 6), rounding errors destroy this improvement. The best results (with respect to d_{\max} , ich , and nch) are obtained with $k + 2 = 7$ (row # 4) in both tables. But, in Table 2, if we look only at d_{\max} and ich (allowing a greater nch), VREM with only 5 vectors (row # 2) gives comparable results.

Let us now take the c_i 's closer to 0.85. With $c_i = 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65$, and $c^* = 0.25$, we obtain the results of Table 3. Comparing these results

with those of the preceding tables shows that greater values of the c_i lead to better results for the three extrapolation methods, as expected. Again, the best results are obtained with $k + 2 = 7$ (row # 4).

TABLE 3. $p = 5000$, $c_i = 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65$, $c^* = 0.25$, $c = 0.85$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	9.30e-7	1.31e-8	3788	18	-16	4139	2461	2477
2	VREM 5	2.01e-8	1.17e-9	635	207	3	1378	2765	2762
3	VREM 6	1.14e-8	6.92e-10	385	251	2	2742	936	934
4	VREM 7	2.65e-9	1.29e-10	66	272	-1	1073	272	273
5	VREM 8	3.16e-9	2.02e-10	114	272	-1	1073	272	273
6	VREM 9	2.07e-9	1.25e-10	66	272	-1	1073	272	273
7	VREM 4	3.52e-5	2.25e-6	4810	10	128	1460	2362	2234
8	SVREM 3	4.02e-6	4.48e-8	4673	14	-101	4436	3529	3630
9	VMP 2	4.15e-6	4.07e-8	4683	14	-102	4436	3529	3631

TABLE 4. $p = 5000$, $c_i = 0.55, 0.6, 0.65$, $c^* = 0.25$, $c = 0.99$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
7	VREM 4	3.52e-5	2.25e-6	4810	10	128	1460	2362	2234
8	SVREM 3	3.08e-5	2.30e-6	4349	29	-40	4436	3529	3569
9	VMP 2	3.03e-5	2.27e-6	4359	29	-40	4436	3529	3569

Computing \mathbf{r}_c for $c = 0.99$ and extrapolating at that same point, we obtain, with $c_i = 0.55, 0.6, 0.65$ and $c^* = 0.25$, the results given in Table 4 where only the second part of the results is given. For $c = 0.99$, 13 iterations of the power method are needed, while, for $c_i = 0.65$ (our highest value) only 6 are required. Of course, these results are less good than the preceding ones, not a surprise since the conditioning of the problem behaves as $(1 - c)^{-1}$. Here SVREM and VMP give comparable results with, obviously, the advantage that VMP needs only 2 vectors. With $c^* = 0.5$, the results are the same.

The computation of the dominant eigenvector of a stochastic matrix appears in various situations (stochastic processes, Markov chains, data mining, information retrieval, networks, etc.) where the dimensions of the problem could be much smaller than in the case of web matrices; see, for example, [1, 18, 17, 19, 22] and [14] where quite small matrices (around 10 or 20) are involved. In these applications, the matrix could be reducible, thus leading to the same problem as with Google's matrix, and a kind of regularization by a parameter c followed by an extrapolation procedure could be of interest. This is the reason why we now consider numerical results obtained by our extrapolation methods for a matrix of dimension 1000 having 18729 nonzero elements. With $c_i = 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45$ and $c^* = 0.5$, extrapolation at 0.85 gives the results of Table 5. The best results are obtained with VREM 9. As in Table 2, if we accept a greater value for nch , then the same $|d_{\max}|$ and ich are obtained with VREM 8. Moreover, since the maximum displacement in the positions is 2 for both cases, the value of ix_{\max} in row # 5

TABLE 5. $p = 1000, c_i = 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, c^* = 0.5, c = 0.85$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	5.14e-5	1.91e-6	913	9	-51	868	348	399
2	VREM 5	8.29e-6	5.44e-7	603	12	9	915	398	389
3	VREM 6	1.39e-6	1.15e-7	175	13	4	233	486	482
4	VREM 7	1.23e-6	1.03e-7	152	62	4	233	486	482
5	VREM 8	5.38e-7	4.60e-8	81	127	-2	796	395	397
6	VREM 9	2.66e-7	2.35e-8	46	127	2	315	145	143
7	VREM 4	5.42e-6	4.46e-7	615	15	-7	978	467	474
8	SVREM 3	1.17e-4	3.63e-6	969	8	101	117	425	324
9	VMP 2	1.35e-4	3.77e-6	966	8	101	117	425	324

TABLE 6. Stanford web: $p = 281903, c = 0.85, c^* = 0.5, c_i = 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _{1/p}$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	1.22e-3	6.26e-7	261573	4	-162408	275310	26841	189249
2	VREM 5	1.78e-3	1.40e-7	261445	4	-105526	178890	19635	125161
3	VREM 6	7.67e-4	1.02e-7	261208	4	-89744	78761	52409	142153
4	VREM 7	4.52e-4	7.50e-8	260291	4	-44139	50293	32553	76692
5	VREM 8	3.00e-4	5.25e-8	260629	4	-52413	247738	116455	168868
6	VREM 9	2.57e-4	6.93e-8	281652	11	-219944	199118	61958	281902
7	VREM 4	2.00e-3	6.74e-7	261353	4	-80743	129116	51569	132312
8	SVREM 3	1.59e-3	8.74e-7	261573	1	-160026	178890	19635	179661
9	VMP 2	1.51e-3	9.27e-7	261574	1	-160620	178890	19635	180255

means that, with VREM 8, all components before the 395th are well ranked (with a possible displacement of only 1 position), a result which is better than with VREM 9.

Finally, let us give the results obtained with the Stanford web matrix of dimension $p = 281903$ which contains 2312497 nonzero elements. On the diagonal of P , 281903 elements are nonzero. The computations with $c_i = 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45$, and $c^* = 0.5$ require at most 22 iterations of the power method, while 91 are needed for $c = 0.85$. The results are given in Table 6.

Results obtained for the Stanford matrix with greater values of the c_i and $c^* = 0.25$ are given in Table 7.

The fact that we considered only matrices of a relatively small dimension in our numerical experiments is also justified by the observation that the number of dangling nodes may exceed the number of nondangling pages. The dangling nodes could be excluded from the computation, thus reducing considerably the dimension of the problem. This can be obtained by lumping all the dangling nodes into a single node [9]. Moreover, this observation is interesting since the PageRanks of dangling nodes strongly depend on the ranks of the nondangling ones, but not vice versa [11, Thm. 4.1].

TABLE 7. Stanford web: $p = 281903, c = 0.85, c^* = 0.25, c_i = 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65$

#	Method	$\ \mathbf{r}_c - \mathbf{p}\ _\infty$	$\ \mathbf{r}_c - \mathbf{p}\ _1/p$	nch	ich	d_{\max}	pos	ix_{\max}	iy_{\max}
1	VREM 4	1.05e-3	4.15e-7	261425	4	-104574	275310	26841	131415
2	VREM 5	5.26e-4	9.61e-8	261240	4	-54793	50293	32553	87346
3	VREM 6	4.02e-4	5.95e-8	260085	7	-37547	50293	32553	70100
4	VREM 7	9.55e-5	1.45e-8	258487	14	-23600	50293	32553	56153
5	VREM 8	3.32e-5	7.33e-9	257896	29	-20639	50293	32553	53192
6	VREM 9	1.03e-5	2.98e-9	254360	14	-13364	40458	38289	51653
7	VREM 4	7.72e-4	2.76e-7	260586	4	-37167	50293	32553	69720
8	SVREM 3	1.59e-3	2.95e-7	261569	1	-79828	129116	51569	131397
9	VMP 2	1.01e-3	4.61e-7	261576	1	-81710	129116	51569	133279

As a conclusion, the best extrapolation procedure seems to be the VREM. But, obviously, more numerical experiments have to be performed to fully understand the role of each parameter, and a theoretical study of the extrapolation error has to be conducted.

ACKNOWLEDGEMENTS

We would like to thank David Enot for information on the Kendall test, and for providing us with his subroutines. We are grateful to the referees for their very careful reading of the paper, and their detailed comments which helped us to clarify some points.

REFERENCES

- [1] M. Bianchini, M. Gori, F. Scarselli, Inside PageRank, *ACM Trans. Internet Tech.*, 5 (2005) 92–128.
- [2] P. Boldi, M. Santini, S. Vigna, PageRank as a function of the damping factor, in *Proceedings of the 14th International World Wide Web Conference*, ACM Press, New York, 2005, pp. 557–566.
- [3] A. Borodin, G.O. Roberts, J.S. Rosenthal, P. Tsaparas, Finding authorities and hubs from link structures on the World Wide Web, in *Proceedings of the 11th International World Wide Web Conference*, ACM Press, New York, 2001, pp. 415–429.
- [4] C. Brezinski, *Projection Methods for Systems of Equations*, North-Holland, Amsterdam, 1997. MR1616573 (99f:65004)
- [5] C. Brezinski, M. Redivo-Zaglia, The PageRank vector: Properties, computation, approximation, and acceleration, *SIAM J. Matrix Anal. Appl.*, 28 (2006) 551–575. MR2255342 (2007h:68205)
- [6] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, S. Seatzu, Extrapolation techniques for ill-conditioned linear systems, *Numer. Math.*, 81 (1998) 1–29. MR1657714 (99j:65069)
- [7] C. Brezinski, M. Redivo-Zaglia, S. Serra-Capizzano, Extrapolation methods for PageRank computations, *C.R. Math. Acad. Sci. Paris*, 340 (2005) 393–397. MR2127117
- [8] P.J. Davis, *Interpolation and Approximation*, Reprint, Dover, New York, 1975. MR0380189 (52:1089)
- [9] N. Eiron, K.S. McCurley, J.A. Tomlin, Ranking the web frontier, in *Proceedings of the 13th International World Wide Web Conference*, ACM Press, New York, 2004, pp. 309–318.
- [10] R.A. Horn, S. Serra-Capizzano, A general setting for the parametric Google matrix, *Internet Math.*, to appear.
- [11] I.C.F. Ipsen, T.M. Selee, PageRank computation, with special attention to dangling nodes, *SIAM J. Matrix Anal. Appl.*, submitted.

- [12] S.D. Kamvar, T.H. Haveliwala, C.D. Manning, G.H. Golub, Extrapolation methods for accelerating PageRank computations, in *Proceedings of the 12th International World Wide Web Conference*, ACM Press, New York, 2003, pp. 261–270.
- [13] M. Kendall, A new measure of rank correlation, *Biometrika*, 30 (1938) 81–89.
- [14] R. Kerner, *Models of Agglomeration and Glass Transition*, Imperial College Press, London, 2006. MR2289782
- [15] A.N. Langville, C.D. Meyer, *Google's PageRank and Beyond. The Science of Search Engine Rankings*, Princeton University Press, Princeton and Oxford, 2006. MR2262054 (2007h:68002)
- [16] R. Lempel, S. Moran, Rank-stability and rank-similarity of link-based web ranking algorithms in authority-connected graphs, in *Second Workshop on Algorithms and Models for the Web-Graph (WAW 2003)*, Budapest, Hungary, May 2003.
- [17] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, U. Alon, Superfamilies of designed and evolved networks, *Science*, 303 (2004) 1538–1542.
- [18] M.E.J. Newman, The structure and function of complex networks, *SIAM Rev.*, 45 (2003) 167–256. MR2010377 (2005a:05206)
- [19] E. Ravasz, A.-L. Barabási, Hierarchical organization in complex networks, *Phys. Rev.*, E 67, 026112 (2003).
- [20] S. Serra-Capizzano, Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation, *SIAM J. Matrix Anal. Appl.*, 27 (2005) 305–312. MR2179674 (2006g:15019)
- [21] S. Serra-Capizzano, Google pageranking problem: The model and the analysis, Proc. of the Dagstuhl Seminar 7071 *Web Information Retrieval and Linear Algebra Algorithms*, March 2007, A. Frommer, M.W. Mahoney, D.B. Szyld eds., <http://drops.dagstuhl.de/opus/volltexte/2007/1069>
- [22] D.J. Watts, Networks, dynamics and small-world phenomenon, *Amer. J. Sociology*, 105 (1999) 493–527.

LABORATOIRE PAUL PAINLEVÉ, UMR CNRS 8524, UFR DE MATHÉMATIQUES PURES ET APPLIQUÉES, UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE, 59655-VILLENEUVE D'ASCQ CEDEX, FRANCE

E-mail address: `Claude.Brezinski@univ-lille1.fr`

UNIVERSITÀ DEGLI STUDI DI PADOVA, DIPARTIMENTO DI MATEMATICA PURA ED APPLICATA, VIA TRIESTE 63, 35121-PADOVA, ITALY

E-mail address: `Michela.RedivoZaglia@unipd.it`