SIAM J. MATRIX ANAL. APPL.
Vol. 38, No. 2, pp. 343–360

BLOCK MATRIX FORMULATIONS FOR EVOLVING NETWORKS*

CATERINA FENU† AND DESMOND J. HIGHAM‡

Abstract. Many types of pairwise interactions take the form of a fixed set of nodes with edges that appear and disappear over time. In the case of discrete-time evolution, the resulting evolving network may be represented by a time-ordered sequence of adjacency matrices. We consider here the issue of representing the system as a single, higher-dimensional block matrix, built from the individual time slices. We focus on the task of computing network centrality measures. From a modeling perspective, we show that there is a suitable block formulation that allows us to recover dynamic centrality measures respecting time's arrow. From a computational perspective, we show that the new block formulation leads to the design of more effective numerical algorithms. In particular, we describe matrix-vector product based algorithms that exploit sparsity. Results are given on realistic data sets.

Key words. centrality, complex network, evolving network, graph, tensor

AMS subject classifications. 05C50, 15A69

DOI. 10.1137/16M1076988

1. Introduction.

A *multilayer network*, also known as a *network of networks* [5, 25], is a graph where connections are formed within and between well-defined slices, each of which is itself a network. In this case it is natural to regard the connectivity structure as a three-dimensional tensor. We focus here on a specific type of multilayering where each slice represents a time point. More precisely, let $\{G^{[k]}\}_{k=1}^{M} = \big(V, \{E^{[k]}\}_{k=1}^{M}\big)$ be a sequence of unweighted graphs evolving in discrete time. Here, the set of nodes V with $|V| = n$ is fixed and the evolution in time is given by the change in the set of edges, $E^{[k]}$. With this notation, given the ordered sequence of time points $\{t_k\}_{k=1}^{M}$, the network at time t_k is represented by its $n \times n$ adjacency matrix $A^{[k]}$. As is usual for unweighted networks, the (i, j)th entry of $A^{[k]}$ equals 1 if there is an edge from node i to node j at time t_k, and 0 otherwise. This type of connectivity structure arises naturally in many types of human interaction. For example, within a given population, we may record physical interactions, phone calls, text messages, e-mails, co-authorships, social media contacts, or correlations between behavior such as energy usage or online shopping; see [22] for an overview.

Although we may regard $\{A^{[k]}\}_{k=1}^{M}$ as a three-dimensional tensor, we emphasize that, in this context, the third-dimension is very different from the first two. Typical quantities of interest are invariant to the ordering of the nodes—we may consistently permute the rows and columns of each $A^{[k]}$, or equivalently, we may relabel the nodes, without affecting our conclusions. However, for most purposes, it is not appropriate to reorder the time points. This raises a question that motivates the work presented here: to what extent can we rely on ideas from the generic multilayer/tensor viewpoint when studying evolving networks? More specifically, how do we express an evolving

*Received by the editors May 25, 2016; accepted for publication (in revised form) by D. B. Szyld January 4, 2017; published electronically May 2, 2017.
http://www.siam.org/journals/simax/38-2/M107698.html
Funding: The work of the second author was supported by EPSRC/RCUK Established Career fellowship EP/M00158X/1 and a Royal Society/Wolfson Research Merit Award.

†Department of Computer Science, University of Pisa, 56127 Pisa, Italy (caterina.fenu@for.unipi.it).
‡Department of Mathematics and Statistics, University of Strathclyde, G1 1XQ Glasgow, U.K. (d.j.higham@strath.ac.uk).

network as a single, large block matrix? In this paper we address this question in the context of computing node centrality.

In recent years, node centrality indices based on the notion of dynamic walks have been introduced and discussed [1, 17, 20]. We review their definitions, introduce a new connection between them and then propose a block matrix representation that allows us to recover centrality indices by applying standard matrix functions. In more detail, given the sequence of adjacency matrices $\{A^{[k]}\}_{k=1}^{M}$, we will describe how the block matrix

$$B := \begin{bmatrix} \alpha A^{[1]} & \beta_2 I & & & \\ & \alpha A^{[2]} & \beta_3 I & & \\ & & \ddots & \ddots & \\ & & & \alpha A^{[M-1]} & \beta_M I \\ & & & & \alpha A^{[M]} \end{bmatrix} \tag{1.1}$$

can be used to capture dynamic centrality. This formulation takes inspiration from the idea of *flattening* a tensor, also known as *reshaping*, *unfolding*, or *matricizing* [15, 26, 30], in the sense that it represents a tensor as a single, larger, two-dimensional array. (However, we note that the matrix B in (1.1) is not a proper unfolding.) Expressing node centrality as entries of standard matrix functions allows us to consider customized algorithms that have improved efficiency.

The material is organized as follows. In section 2, we review a class of centrality measures based on the concept of dynamic walks. In section 3 we then present a block representation from which these centrality measures can be recovered using standard matrix functions. Methods for the computation of the centrality measures using the new block approach are presented in 4. Computational experiments on synthetic and voice call data are described in section 5. Within these tests, we also study the supracentrality matrix formulation from [34]. Final conclusions are given in section 6.

2. Centrality.

In this section, we review the concepts of time-dependent centrality measures from [1, 17, 20] and introduce a new connection between them. Centrality measures are widely used for identifying influential players in a network. Many such measures arose within the field of social network analysis, motivated either explicitly or implicitly from the idea that the network nodes communicate, or pass information, along the edges; see, for example, [6, 13]. In this way, centrality quantifies a sense in which a node takes part in traversals. Quoting from [7], "all measures of centrality assess a node's involvement in the walk structure of a network."

For the time-dependent links that we consider here, it has been pointed out by several authors that any type of message-passing (or disease-passing) basis for centrality should account for the time-ordering of the interactions; see, for example, [9, 22, 31]. If X meets Y today and Y meets Z tomorrow, then the path $X \to Y \to Z$ makes sense from a message-passing point of view, but $Z \to Y \to X$ does not. Traversals must respect the arrow of time.

In [20], as a means to develop a time-dependent centrality measure, the authors introduced the notion of a *dynamic walk* as follows.

DEFINITION 2.1. *A* dynamic walk *of length w from node i_1 to node i_{w+1} consists of a sequence of edges $i_1 \to i_2$, $i_2 \to i_3$, ..., $i_w \to i_{w+1}$ and a nondecreasing sequence of times $t_{r_1} \le t_{r_2} \le \cdots \le t_{r_w}$ such that $A^{[r_m]}_{i_m, i_{m+1}} \neq 0$.*

This definition was used to define the *dynamic communicability matrix*

$$\mathcal{Q}^{[j]} = \left(I - aA^{[1]}\right)^{-1} \left(I - aA^{[2]}\right)^{-1} \cdots \left(I - aA^{[j]}\right)^{-1} = \prod_{s=1}^{j} \left(I - aA^{[s]}\right)^{-1}. \tag{2.1}$$

We assume henceforth that the parameter a satisfies $a < 1/\max_s \rho(A^{[s]})$ with $\rho(A^{[s]})$ denoting the spectral radius of the matrix $A^{[s]}$. Each resolvent in (2.1) may then be expanded as

$$\left(I - aA^{[s]}\right)^{-1} = \sum_{k=0}^{\infty} a^k \left(A^{[s]}\right)^k.$$

In view of this, $(\mathcal{Q}^{[j]})_{ij}$ can be seen as a weighted sum of the number of dynamic walks from i to j using the ordered sequence $\{A^{[k]}\}_{k=1}^{M}$, in which the count for walks of length w is scaled by a^w. The overall ability of nodes to broadcast or receive information in this sense is given by the row and column sums

$$C^{\text{broadcast}} = \mathcal{Q}^{[j]}\mathbf{1} \qquad \text{and} \qquad C^{\text{receive}} = {\mathcal{Q}^{[j]}}^T\mathbf{1}, \tag{2.2}$$

respectively, where $\mathbf{1}$ is the vector of all ones. See [20] for a more detailed explanation. Numerical tests in [20] showed that these broadcast and receive centralities are generally very different from the measures that arise when we ignore time-dependency and consider only the aggregate adjacency matrix $\sum_{k=1}^{M} A^{[k]}$, and subsequent work in [27] showed that they were better able to match the views of social media experts when applied to Twitter data. Using real interaction data concerning patients and staff in a large hospital, the study in [9] showed that broadcast centrality adds value in the search for significant spreaders of disease.

Motivated by the treatment of static networks in [11], the authors in [1] used the dynamic communicability matrix idea to introduce two kinds of dynamic betweenness: the *nodal betweenness* of a node and the *temporal betweenness* of a time point.

Let $\bar{A}_r^{[k]}$ denote the matrix obtained from $A^{[k]}$ by removing all the edges involving node r, that is, $\bar{A}_r^{[k]} = A^{[k]} - E_r^{[k]}$, where $E_r^{[k]}$ has nonzero elements only in row and column r, which coincide with those of $A^{[k]}$. Then, the matrix

$$\bar{\mathcal{Q}}_r^{[M]} = \prod_{s=1}^{M} \left(I - a\bar{A}_r^{[s]}\right)^{-1}$$

quantifies the ability of nodes to communicate without using node r. The *nodal betweenness of node* r [1] is defined as

$$\text{NB}_r := \frac{1}{(n-1)^2 - (n-1)} \sum_{i \neq j \neq r}\sum \frac{(\mathcal{Q}^{[M]})_{ij} - (\bar{\mathcal{Q}}_r^{[M]})_{ij}}{(\mathcal{Q}^{[M]})_{ij}}. \tag{2.3}$$

This measure quantifies the relative decrease in information exchange when node r is removed from the network.

Let $\{\widehat{A}^{[k,q]}\}_{k=1}^{M}$ denote the adjacency matrix sequence obtained by replacing $A^{[q]}$ with 0, that is,

$$\widehat{A}^{[k,q]} = (1 - \delta_{kq})A^{[k]},$$

where δ_{kq} is the Kronecker delta. Then, the matrix

$$\widehat{\mathcal{Q}}^{[M,q]} = \prod_{s=1}^{M} \left(I - a\widehat{A}^{[s,q]}\right)^{-1} = \prod_{\substack{s=1\\ s\neq q}}^{M} \left(I - aA^{[s]}\right)^{-1}$$

describes how well nodes interchange information without using the connections at time q. The *temporal betweenness* [1] of time point q is defined as

$$\mathrm{TB}^{[M,q]} := \frac{1}{(n-1)^2-(n-1)}\sum\sum_{i\neq j}\frac{(\mathcal{Q}^{[M]})_{ij}-(\widehat{\mathcal{Q}}^{[M,q]})_{ij}}{(\mathcal{Q}^{[M]})_{ij}}. \tag{2.4}$$

We refer to [1] for further details and illustrative examples involving these measures, and we note that in practical use the matrices $\mathcal{Q}^{[M]}$, $\bar{\mathcal{Q}}_r^{[M]}$, and $\widehat{\mathcal{Q}}^{[M,q]}$ should be properly scaled in order to prevent over/underflows.

For future convenience, we extend the notation to allow for walks that start and finish at arbitrary time points. Let us denote by $\mathcal{Q}^{[i,j]}$ the dynamic communicability matrix obtained by multiplying the resolvents corresponding to the ordered sequence $\{A^{[s]}\}_{s=i}^{j}$, $1\le i\le j\le M$, that is,

$$\mathcal{Q}^{[i,j]} = \prod_{s=i}^{j}\left(I-aA^{[s]}\right)^{-1} = \left(I-aA^{[i]}\right)^{-1}\cdots\left(I-aA^{[j]}\right)^{-1}. \tag{2.5}$$

With this notation we can quantify broadcast and receive centralities over any subinterval. In general, we may use

$$C_{\text{broadcast}}^{[i,j]} = \mathcal{Q}^{[i,j]}\mathbf{1} \qquad \text{and} \qquad C_{\text{receive}}^{[i,j]} = {\mathcal{Q}^{[i,j]}}^T\mathbf{1} \tag{2.6}$$

to quantify the ability of a node to spread or receive information, respectively, taking into account the evolution of the network between t_i and t_j.

In many applications, such as the spread of rumors or disease, recent walks are more important than those that started a long time ago. For this reason, the authors in [17] introduced the *running dynamic communicability matrix*, $\mathcal{S}^{[j]}$, obtained recursively, starting from $\mathcal{S}^{[0]}=0$, as

$$\mathcal{S}^{[j]} = \left(I+e^{-b\Delta t_j}\mathcal{S}^{[j-1]}\right)\left(I-aA^{[j]}\right)^{-1}-I, \qquad j=1,\dots,M, \tag{2.7}$$

where $\Delta t_j = t_j - t_{j-1}$. In this recurrence, the parameter a is used to penalize long walks and the parameter b is used to filter out old activity. Overall, $\mathcal{S}^{[j]}$ maintains walk counts that are scaled in terms of length w by a^w and chronological age t by e^{-bt}. Running versions of the broadcast and receive communicabilities are then given by the row/column sums of the matrix $\mathcal{S}^{[j]}$, that is,

$$\mathcal{S}^{[j]}\mathbf{1} \qquad \text{and} \qquad {\mathcal{S}^{[j]}}^T\mathbf{1}. \tag{2.8}$$

For use in the next section, the following lemma points out a connection between the running dynamic communicability matrix $\mathcal{S}^{[j]}$ in (2.7) and the dynamic communicability matrices $\mathcal{Q}^{[i,j]}$ in (2.5).

LEMMA 2.2. *For the running dynamic communicability matrix $\mathcal{S}^{[j]}$ in* (2.7)*, we have*

$$\mathcal{S}^{[j]} = \sum_{i=1}^{j}\left(1-e^{-b\Delta t_i}\right)e^{-b\sum_{\ell=i+1}^{j}\Delta t_\ell}\mathcal{Q}^{[i,j]} - I,$$

where $\Delta t_1=\infty$.

Proof. The proof uses induction. For $j = 1$, we have

$$\mathcal{S}^{[1]} = \left(I + e^{-b\Delta t_1}\mathcal{S}^{[0]}\right)\left(I - aA^{[1]}\right)^{-1} - I = \mathcal{Q}^{[1,1]} - I.$$

Suppose that the identity is valid for $j = k - 1$. We will then show that it is valid also for $j = k$. We have

$$\begin{aligned}
\mathcal{S}^{[k]} &= \left(I + e^{-b\Delta t_k}\mathcal{S}^{[k-1]}\right)\mathcal{Q}^{[k,k]} - I \\
&= \mathcal{Q}^{[k,k]} + e^{-b\Delta t_k}\left[\sum_{i=1}^{k-1}\left(1 - e^{-b\Delta t_i}\right)e^{-b\sum_{\ell=i+1}^{k-1}\Delta t_\ell}\mathcal{Q}^{[i,k-1]} - I\right]\mathcal{Q}^{[k,k]} - I \\
&= \sum_{i=1}^{k-1}\left(1 - e^{-b\Delta t_i}\right)e^{-b\sum_{\ell=i+1}^{k-1}\Delta t_\ell}e^{-b\Delta t_k}\mathcal{Q}^{[i,k-1]}\mathcal{Q}^{[k,k]} + \left(1 - e^{-b\Delta t_k}\right)\mathcal{Q}^{[k,k]} - I \\
&= \sum_{i=1}^{k-1}\left(1 - e^{-b\Delta t_i}\right)e^{-b\sum_{\ell=i+1}^{k}\Delta t_\ell}\mathcal{Q}^{[i,k]} + \left(1 - e^{-b\Delta t_k}\right)\mathcal{Q}^{[k,k]} - I \\
&= \sum_{i=1}^{k}\left(1 - e^{-b\Delta t_i}\right)e^{-b\sum_{\ell=i+1}^{k}\Delta t_\ell}\mathcal{Q}^{[i,k]} - I,
\end{aligned}$$

where we used the fact that $\mathcal{Q}^{[i,k-1]}\mathcal{Q}^{[k,k]} = \mathcal{Q}^{[i,k]}$. ◻

3. Block matrix formulations.

Our aim now is to study block matrix representations of the data $\{A^{[k]}\}_{k=1}^{M}$ that transform the network sequence into an "equivalent" large, static network with adjacency matrix of dimension Mn. We have two main requirements for such a representation.

- We would like to be able to interpret this static network in terms of the interactions represented by the original data.
- We would like to be able to recover the dynamic centrality measures discussed in the previous section by applying standard matrix functions to this larger network.

In the case of general multilayer networks, the authors in [32] introduce an *influence matrix* $W \in \mathbb{R}^{M\times M}$ such that $w_{ij} \geq 0$ measures the influence of layer j on layer i. They then study node centrality via the Mn by Mn matrix

$$\begin{bmatrix} w_{11}A^{[1]} & w_{12}A^{[2]} & \dots & w_{1M}A^{[M]} \\ w_{21}A^{[1]} & w_{22}A^{[2]} & \dots & w_{2M}A^{[M]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1}A^{[1]} & w_{M2}A^{[2]} & \dots & w_{MM}A^{[M]} \end{bmatrix}.$$

In our specific context, where (a) the layers represent time slices that have a natural ordering, and (b) centrality concepts are motivated from traversals around the network, this formulation appears to add little value. If each $M \times M$ block represents a time slice, then the existence of an edge at one time slice should not influence the propensity for traversal *within* some other time slice. Hence, only the simple block-diagonal version ($w_{ij} = 0$ for all $i \neq j$) makes intuitive sense in our context.

Returning to Definition 2.1, we note that a dynamic walk may use any number of edges within a time slice and may then *wait* until a later time slice and continue the traversal. For dynamic communicability defined via (2.1), within each time slice, use of an edge penalizes the walk count by a and moving from one time slice to the next carries no penalty. For the more general running measure based on (2.7), waiting until the next time slice costs a factor $e^{-b\Delta t_j}$. We may capture this type of weighted count by introducing a link *from a node at one time slice to the equivalent node at the next time slice*; that is, by adding identity matrices along the first superdiagonal to obtain $B \in \mathbb{R}^{Mn\times Mn}$ defined in (1.1), where $\{\beta_\ell\}_{\ell=2,M}$ and α are parameters. The matrix B represents the interactions among the nodes of the evolving network as a larger static network with Mn nodes. More precisely, each node in the original data is copied M times and edges within each time point are represented as connections among the same group of nodes. Moreover, the information can flow from a time slice to another by using links that connect each node to its equivalent in the next time step.

The next theorem confirms that this structure captures the required communicabilities when $\alpha = a$ and $\beta_\ell \equiv 1$ or $\beta_\ell = e^{-b\Delta t_\ell}$ for the two cases, that is, it satisfies also the second requirement we listed at the beginning of this section.

THEOREM 3.1. *The dynamic communicability matrices* $\mathcal{Q}^{[i,j]}$ *in* (2.5) *and the running dynamic communicability matrices* $\mathcal{S}^{[j]}$ *in* (2.7) *can be computed by applying the function* $f(x) = (1-x)^{-1}$ *to the matrix* B *in* (1.1).

Proof. It is straightforward to show that the kth power of the matrix B has the form

$$B^k = \begin{bmatrix} \alpha^k h_k\left(A^{[1]}\right) & \beta_2\alpha^{k-1}h_{k-1}(A^{[1]},A^{[2]}) & \cdots & \prod_{\ell=2}^{M}\beta_\ell\alpha^{k-r}h_{k-r}(A^{[1]},\dots,A^{[M]}) \\ & \alpha^k h_k\left(A^{[2]}\right) & \ddots & \vdots \\ & & \ddots & \beta_M\alpha^{k-1}h_{k-1}(A^{[M-1]},A^{[M]}) \\ & & & \alpha^k h_k\left(A^{[M]}\right) \end{bmatrix},$$

where

$$h_k(x_1,x_2,\dots,x_n) = \sum_{l_1+l_2+\cdots l_n = k} {x_1}^{l_1}{x_2}^{l_2}\cdots {x_n}^{l_n}$$

is the *complete homogeneous symmetric polynomial* of degree k, $r = M-1$, and $h_k(\cdot,\cdot,\dots,\cdot) = 0$ if $k < 0$.

In general, denoting block (i,j), $(i,j = 1,\dots,M)$ of B^k by $[B^k]_{ij}$, we have that

$$[B^k]_{ij} = \beta^{[i+1,j]}\alpha^{k+i-j}h_{k+i-j}(A^{[i]},\dots,A^{[j]}), \qquad i \le j,$$

where $\beta^{[i+1,j]}$ denotes the scalar $\prod_{\ell=i+1}^{j}\beta_\ell$.

The matrix-valued function $f(B) = \sum_{k=0}^{\infty} B^k$ has blocks

$$[f(B)]_{ij} = \beta^{[i+1,j]}\sum_{k=0}^{\infty}\alpha^{k+i-j}h_{k+i-j}(A^{[i]},\dots,A^{[j]}) = \beta^{[i+1,j]}\prod_{\ell=i}^{j}(I-\alpha A^{[\ell]})^{-1}.$$

Hence, the dynamic communicability matrices $\mathcal{Q}^{[i,j]}$ can be obtained from the block $[f(B)]_{ij}$ setting $\beta_\ell = 1$ for $\ell = 2,\dots,M$ and $\alpha = a$.

The running dynamic communicability matrices $\mathcal{S}^{[j]}$ are obtained starting from the blocks on the jth block-column. In particular, setting $\beta_\ell = e^{-b\Delta t_\ell}$, $\alpha = a$, and

$D = (1-e^{-b\Delta t_1}, \dots, 1-e^{-b\Delta t_M}) \otimes I_n$, we have

$$\left[\sum_{i=1}^{j} [Df(B)]_{ij}\right] - I = \sum_{i=1}^{j} \left(1-e^{-b\Delta t_i}\right) \beta^{[i+1,j]} \mathcal{Q}^{[i,j]} - I$$
$$= \sum_{i=1}^{j} \left(1-e^{-b\Delta t_i}\right) e^{-b\sum_{\ell=i+1}^{j} \Delta t_\ell} \mathcal{Q}^{[i,j]} - I.$$

The statement now follows from Lemma 2.2. ∎

Similar statements apply to the betweenness measures in (2.3) and (2.4).

Theorem 3.2. *Let $\bar{A}_r^{[k]}$ denote the matrix obtained from $A^{[k]}$ by removing all the edges involving node r and let $\{\widehat{A}^{[k,q]}\}_{k=1}^M$ be the adjacency matrix sequence obtained by replacing $A^{[q]}$ with 0. Then, for $f(x) = (1-x)^{-1}$,*

$$\mathrm{NB}_r = \frac{1}{(n-1)^2-(n-1)} \sum_{i\neq j\neq r}\sum \frac{[f(B)]_{1M}^{ij} - \left[f(\bar{B}_r)\right]_{1M}^{ij}}{[f(B)]_{1M}^{ij}},$$

$$\mathrm{TB}^{[M,q]} = \frac{1}{(n-1)^2-(n-1)} \sum_{i\neq j}\sum \frac{[(f(B)]_{1M}^{ij} - [(f(\widehat{B}^{[q]})]_{1,M-1}^{ij}}{[f(B)]_{1M}^{ij}},$$

where $\bar{B}_r$ and $\widehat{B}^{[q]}$ are given by

$$\bar{B}_r = \begin{bmatrix} \alpha\bar{A}_r^{[1]} & I & & & \\ & \alpha\bar{A}_r^{[2]} & I & & \\ & & \ddots & \ddots & \\ & & & \alpha\bar{A}_r^{[M-1]} & I \\ & & & & \alpha\bar{A}_r^{[M]} \end{bmatrix},$$

$$\widehat{B}^{[q]} = \begin{bmatrix} \alpha A^{[1]} & I & & & & & & \\ & \alpha A^{[2]} & I & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \alpha A^{[q-1]} & I & & & \\ & & & & \alpha A^{[q+1]} & I & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \alpha A^{[M-1]} & I \\ & & & & & & & \alpha A^{[M]} \end{bmatrix}$$

and $[(f(B)]_{\ell,k}^{ij}$ denotes the (i,j)th element of the (ℓ,k)th block of the matrix $f(B)$.

Proof. A proof follows along the same lines as that of Theorem 3.1. ∎

3.1. Another formulation. An alternative block matrix formulation that was specifically designed for evolving networks appears in [34]. Those authors define the *supracentrality matrix* to have the general form

$$\mathbb{M} = \begin{bmatrix} \epsilon M^{[1]} & I & & & \\ I & \epsilon M^{[2]} & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & \epsilon M^{[M-1]} & I \\ & & & I & \epsilon M^{[M]} \end{bmatrix}. \tag{3.1}$$

Here, $M^{[k]}$ is an n by n centrality matrix based on $A^{[k]}$. The authors use the simple choice $M^{[k]} \equiv A^{[k]}$ to illustrate the idea, but mention that other static centrality functions could be used, such as the Katz [24] resolvent-based version $M^{[k]} \equiv (I - \alpha A^{[k]})^{-1}$ (which is the single time-point case of (2.1)). It is then proposed in [34] to apply a standard static network centrality algorithm to the supracentrality matrix $\mathbb{M}$. The parameter ϵ in (3.1) is included to account for the fact that the identity matrices represent "between layer" connections that are inherently different from the "within layer" weights arising from the network data. A key element of (3.1) is the appearance of identity matrices in the super- *and sub*-block-diagonal positions. If the overall centrality measure applied to $\mathbb{M}$ is motivated by monitoring traversals around the large, static Mn by Mn network, then, because of the identity matrices on the subdiagonal blocks, some of these traversals will be traveling *backwards in time* with respect to the original time-stamped data. Similarly, suppose that each $A^{[k]}$ is symmetric, so that edges are undirected in each time slice. Then, with $M^{[k]} \equiv A^{[k]}$ or $M^{[k]} \equiv (I - \alpha A^{[k]})^{-1}$, we see that $\mathbb{M}$ is symmetric. However, from the simple example mentioned in section 2, where X meets Y today and Y meets Z tomorrow, we can see that time's arrow introduces asymmetry, even when the individual interactions are symmetric. In section 5, we will perform some illustrative tests that compare results from (1.1) and (3.1).

4. Computational tasks.

Including a temporal dimension in network data can pose a considerable computational challenge. For example, a recent study [31] monitored physical proximity for a cohort of around 1000 university students. Alluding to the large volume of time-stamped data that arose, the authors argue that "there is currently no coherent theoretical framework for summarizing the tens of thousands of interactions per day in this complex network." Focusing on the computation of running broadcast and receive communicabilities given by (2.8), our aim in this section is to consider algorithms that can deal with tens of thousands of nodes over multiple time points.

From Theorem 3.1, setting $\alpha = a$, $\beta_\ell = e^{-b\Delta t_\ell}$, and $f(x) = (1-x)^{-1}$, we obtain

$$\begin{aligned} \mathcal{S}^{[j]}\mathbf{1}_{\mathrm{n}} &= (\mathbf{d} \otimes I_{\mathrm{n}})^T f(B)(\mathbf{e}_j \otimes \mathbf{1}_{\mathrm{M}}) - \mathbf{1}_{\mathrm{n}}, \\ {\mathcal{S}^{[j]}}^T\mathbf{1}_{\mathrm{n}} &= (\mathbf{e}_j \otimes I_{\mathrm{n}})^T f(B)^T(\mathbf{d} \otimes \mathbf{1}_{\mathrm{n}}) - \mathbf{1}_{\mathrm{n}}, \end{aligned} \tag{4.1}$$

where $\mathbf{d} = [1, 1-\beta_2, \ldots, 1-\beta_M]^T$, and $\mathbf{1}_{\mathrm{M}}$ and $\mathbf{1}_{\mathrm{n}}$ are vectors of all ones in $\mathbb{R}^M$ and $\mathbb{R}^n$, respectively.

The computation of the running broadcast and receive communicabilities (4.1) can be dealt with by using two different approaches: the first one involves the computation of quantities of the kind $\mathbf{u}^T f(B)\mathbf{v}$, with $\mathbf{u}, \mathbf{v}$ vectors of length Mn, and the second one focuses on the solution of a sparse linear system. In the following, we will compare these methods in terms of execution time and accuracy.

4.1. Use of quadrature formulas.

We are interested in computation of quantities of the form

$$\mathbf{u}^T f(B)\mathbf{v}, \qquad \mathbf{u}, \mathbf{v} \in \mathbb{R}^{Mn},$$

with $\mathbf{u}, \mathbf{v}$ unit vectors and $f(B) = (I-B)^{-1}$ nonsymmetric.

In particular, $\mathbf{u} = \mathbf{d} \otimes \mathbf{e}_i, \mathbf{v} = \mathbf{e}_j \otimes \mathbf{1}_{\mathrm{M}}$ and $\mathbf{u} = \mathbf{e}_j \otimes \mathbf{e}_i, \mathbf{v} = \mathbf{d} \otimes \mathbf{1}_{\mathrm{n}}$, $i = 1, \ldots, n$, for the broadcast and receive running communicabilities of node i, respectively.

In this case, since both the vectors $\mathbf{u}$ and $\mathbf{v}$ and the matrix B are very sparse, the probability of breakdown during the computation is high. For this reason, it is convenient to add a dense vector to each initial vector (see [2]) and resort to a block algorithm. In particular, we use the nonsymmetric block Lanczos algorithm [14] and pairs of block Gauss and anti-Gauss quadrature rules [8, 12, 28].

If $U = [\mathbf{u}\ \mathbf{1}]$ and $V = [\mathbf{v}\ \mathbf{1}]$, then we want to approximate the quantities

$$U^T f(B)V, \qquad U, V \in \mathbb{R}^{Mn\times 2}.$$

The nonsymmetric block Lanczos algorithm applied to the matrix B with initial blocks $U_1 = U$ and $V_1 = V$ yields, after ℓ steps, the decompositions

$$\begin{aligned} B\,[U_1,\dots,U_\ell] &= [U_1,\dots,U_\ell]\,J_\ell + U_{\ell+1}\Gamma_\ell \mathbf{E}_\ell^T,\\ B^T\,[V_1,\dots,V_\ell] &= [V_1,\dots,V_\ell]\,J_\ell^T + V_{\ell+1}\Delta_\ell \mathbf{E}_\ell^T, \end{aligned}$$

where J_ℓ is the matrix

$$J_\ell = \begin{pmatrix} \Omega_1 & \Delta_1^T & & & \\ \Gamma_1 & \Omega_2 & \Delta_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{\ell-2} & \Omega_{\ell-1} & \Delta_{\ell-1}^T \\ & & & \Gamma_{\ell-1} & \Omega_\ell \end{pmatrix} \in \mathbb{R}^{2\ell\times 2\ell}, \tag{4.2}$$

and Ω_i, Δ_i, and Γ_i are computed by using the three-term recurrence in the nonsymmetric block Lanczos algorithm; see [14, 12] for details. Moreover, $\mathbf{E}_k = \mathbf{e}_k^T \otimes I_2$ for $k = 1, 2, \dots, \ell$ are $2 \times (2\ell)$ block matrices which contain 2×2 zero blocks everywhere, except for the kth block, which coincides with the identity matrix I_2. The ℓ-block nonsymmetric Gauss quadrature rule $\mathcal{G}_\ell$ can then be expressed as

$$\mathcal{G}_\ell = \mathbf{E}_1^T g(J_\ell)\mathbf{E}_1.$$

As shown in [12], the $(\ell+1)$-block nonsymmetric anti-Gauss rule can be computed in terms of the matrix $\widetilde{J}_{\ell+1}$ as

$$\mathcal{H}_{\ell+1} = \mathbf{E}_1^T g(\widetilde{J}_{\ell+1})\mathbf{E}_1,$$

where

$$\widetilde{J}_{\ell+1} = \left(\begin{array}{c|c} J_\ell & \begin{matrix} \\ \sqrt{2}\Delta_\ell^T \end{matrix} \\ \hline \begin{matrix} & \sqrt{2}\Gamma_\ell \end{matrix} & \Omega_{\ell+1} \end{array}\right) \in \mathbb{R}^{2(\ell+1)\times 2(\ell+1)}.$$

Since $f(x) = (1-x)^{-1}$ with $|x| < 1$ is analytic in a simply connected domain whose boundary encloses the spectrum of B but is not close to it (see [12] for details), the arithmetic mean

$$F_\ell = \frac{1}{2}\left(\mathcal{G}_\ell + \mathcal{H}_{\ell+1}\right) \tag{4.3}$$

between Gauss and anti-Gauss quadrature rules can be used as an approximation of the matrix-valued expression $U^T f(B)V$.

4.2. Resolution of a sparse linear system. Using the same notation as in subsection 4.1, we need to compute the quantities

$$\mathbf{u}^T(I-B)^{-1}\mathbf{v}, \qquad \mathbf{u},\mathbf{v}\in\mathbb{R}^{Mn}.$$

This can be done by solving the sparse linear system $(I-B)\mathbf{x}=\mathbf{v}$ and then computing the scalar product $\mathbf{u}^T\mathbf{x}$.

The linear system can be solved either directly or iteratively. The peculiarity of the block formulation allows us to have at hand a regular matrix splitting. In fact, we have $I\geq 0$, $B\geq 0$, and $\rho(B)<1$. Therefore, the iterative method

$$\mathbf{x}^{(k+1)}=B\mathbf{x}^{(k)}+\mathbf{v},$$

with given starting vector $\mathbf{x}^{(0)}$, converges to the solution $\mathbf{x}$.

Another classical approach to solving linear systems is the *LSQR method* that makes use of the Golub–Kahan algorithm. After ℓ steps of this method with starting vector $q_1=\mathbf{v}$, the solution $\mathbf{x}^{(\ell)}\in\mathbb{R}^{Mn}$ is defined as

$$\mathbf{x}^{(\ell)}=P_\ell\mathbf{y}^{(\ell)}=\beta_1P_\ell C^{\dagger}_{\ell+1,\ell}\mathbf{e}_1,$$

where P_ℓ and $C_{\ell+1,\ell}$ are computed via the Golub–Kahan algorithm. Then, $\mathbf{y}^{(\ell)}\in\mathbb{R}^\ell$ is the solution of the least squares problem

$$\min\|C_{\ell+1,\ell}\mathbf{y}^{(\ell)}-\beta_1\mathbf{e}_1\|_2,$$

where $\beta_1=\|\mathbf{v}\|$ and $\mathbf{e}_1$ is the first vector of the canonical base of size $\ell+1$.

5. Computational tests. In this section, we perform some numerical tests in order to judge the effectiveness of the new block matrix formulation, from both the computational and modeling points of view. First, we compare the methods described in the previous section against the original approach presented in [17], that is, by using the recursive formula (2.7). Then we show the relevance of the upper triangular block formulation compared with the supracentrality matrix approach given in [34].

5.1. Computation using the new block approach. As a first set of experiments, we compare different ways of dealing with the computation of the quantities defined by (2.8). In particular, we focus on the computation of the running broadcast communicabilities $\mathcal{S}^{[j]}\mathbf{1}$. We recall that the computation of these communicabilities at a given time step can not be recovered using the same information at a previous time step, that is, the update of the running broadcast communicabilities $\mathcal{S}^{[j]}\mathbf{1}$ needs the computation of the whole matrix $\mathcal{S}^{[j]}$.

We analyze the following methods.

original is the original approach presented in [17]. In particular, we use the mldivide MATLAB function to compute the inverse of the matrix $\left(I-aA^{[j]}\right)$ in recursion (2.7).

quadrules is the approach based on the Gauss and anti-Gauss quadrature rules described in subsection 4.1. More precisely, we perform as many steps of the block nonsymmetric Lanczos algorithm as necessary to obtain a relative distance

$$\frac{\|\mathcal{G}_\ell-\mathcal{H}_{\ell+1}\|_{\max}}{\|F_\ell\|_{\max}} \qquad \text{with} \quad \|X\|_{\max}:=\max_{1\leq i,j\leq k}|X_{ij}|$$

less than 10^{-3}.

linsolv is the first procedure described in subsection 4.2, in which we solve the big linear system $(I - B)\mathbf{x} = \mathbf{v}$ using the mldivide MATLAB function.

iterative is the second method proposed in subsection 4.2, namely, the iterative approach based on the regular matrix splitting $I - B$. We perform as many iterations as necessary to reach a relative accuracy of 10^{-3} on the difference between two consecutive approximations.

lsqr is the method based on the solution of the linear system obtained by using the lsqr MATLAB function. In particular, we set the tolerance to 10^{-3}.

We want to test the performance of the methods when the size of the matrix B in (1.1) increases. This can be done in various ways. As a first approach, in order to simulate the evolution on a given set of nodes, we independently sample M times from the same static network model with a fixed number of nodes n. This is done using the random network package CONTEST [33]. As a second approach, we generate the M matrices by using the evolving network model proposed and analyzed in [19]. Here, the network sequence corresponds to the sample path of a discrete time Markov chain, and hence the adjacency matrices are correlated over time. All computations were carried out with MATLAB version 9.0 (R2016a) 64-bit for Linux, in double precision arithmetic, on an Intel Xeon computer with 32 Gb RAM.

Table 1 shows the results obtained for the scale-free random graph model generated using the pref function of the CONTEST toolbox, which implements a preferential attachment model. We set a fixed number of nodes $n = 10^3$ and let M range from 20 to 100 in order to test the performance of the methods when the number of time steps is increasing. The table displays the time required to compute the running broadcast communicabilities of the n nodes of the evolving network and the absolute error $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty = \max_i |x_i - \tilde{x}_i|$, where $\tilde{\mathbf{x}}$ is the approximation and $\mathbf{x}$ is the vector computed with the original approach.

The results clearly show that the quadrature rules based on the block Lanczos algorithm do not improve the performance of the original method, while both methods based on the resolution of the big linear system work well. In particular, the difference between the original solution $\mathbf{x}$ and the one computed via the mldivide MATLAB function is at the level of machine precision, as is the one obtained using the lsqr function; for this reason, the error values are not reported for those two methods. It is also worth noting that the iterative method gives good results and is very fast, keeping in mind that we are interested in the rank of the nodes rather than the value of the index.

To investigate the behavior of the methods proposed with respect to the network model, we performed the same computation as in Table 1 using a range-dependent random graph generated using the renga function of the CONTEST toolbox. Table 2 shows the results obtained setting $n = 10^3$ and varying M from 20 to 100. The results

TABLE 1

Execution time and absolute error for the computation of the running broadcast communicabilities with $n = 1000$ and $M = 20, \ldots, 100$. The network model is obtained from the **pref** *function of the CONTEST toolbox.*

	original	**quadrules**		**linsolv**	**iterative**		**lsqr**
M	time	time	err.	time	time	err.	time
20	3.46e+01	4.36e+01	3.34e-03	4.05e+00	1.17e-02	1.06e-03	3.87e-02
40	7.06e+01	1.06e+02	6.23e-03	2.77e+01	1.12e-02	1.05e-03	6.56e-02
60	1.16e+02	1.52e+02	2.52e-03	6.08e+01	1.40e-02	1.01e-03	6.76e-02
80	1.45e+02	2.44e+02	3.30e-03	1.04e+02	1.62e-02	9.22e-04	9.85e-02
100	1.94e+02	2.48e+02	3.94e-03	1.53e+02	1.94e-02	1.06e-03	1.49e-01

Table 2

Execution time and absolute error for the computation of the running broadcast communicabilities with $n = 1000$ and $M = 20, \dots, 100$. The network model is obtained from the **renga** *function of the CONTEST toolbox.*

	original	**quadrules**		**linsolv**	**iterative**		**lsqr**
M	time	time	err.	time	time	err.	time
20	3.34e+01	1.27e+02	2.18e-02	8.90e-01	2.48e-02	8.89e-03	1.71e-01
40	6.87e+01	2.23e+02	3.93e-02	2.34e+00	2.58e-02	1.04e-02	3.14e-01
60	1.07e+02	3.40e+02	2.18e-02	3.60e+00	2.61e-02	1.20e-02	4.38e-01
80	1.39e+02	4.55e+02	8.73e-02	4.94e+00	3.02e-02	1.02e-02	5.90e-01
100	1.74e+02	7.09e+02	1.66e-02	6.51e+00	3.31e-02	8.91e-03	7.09e-01

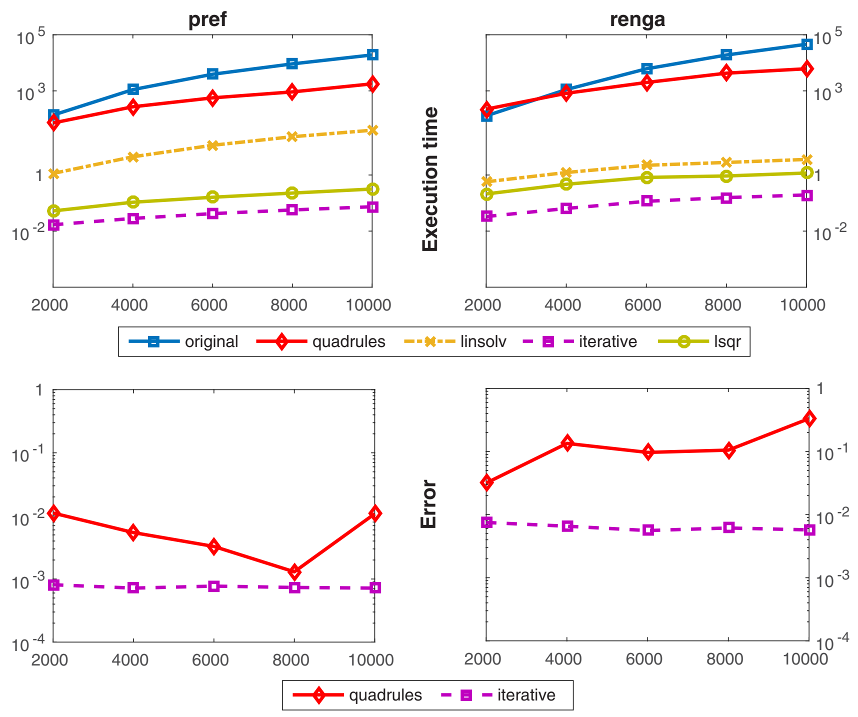

Fig. 1. *Log-scale plot of the execution time (upper graphs) and the absolute error (lower graphs) for the computation of the running broadcast communicabilities with $M = 10$ and $n = 2000, \dots, 10000$. The network model is obtained from the* **pref** *function (left graphs) and the* **renga** *function (right graphs) of the CONTEST toolbox.*

show that the block quadrature rule method and the iterative resolution of the big linear system are slower and the error is greater than that obtained from pref. However, the performance of the direct solution of the linear system is faster and gives a small error. Again, the LSQR method is the best among those proposed.

We now investigate the behavior of the methods when the number of time steps is fixed and the size of the network increases. Figure 1 shows the results obtained when $M = 10$ and n goes from 2000 to 10000 with respect to the pref function (left graphs) and the renga function (right graphs). It is clear that the computation is strongly affected by the complexity of the calculation when the number of nodes increases rather than when the number of time steps is large. Indeed, the original method needs to invert M matrices of size n whose level of sparsity decreases at each time step. On the contrary, none of the methods proposed here need the explicit inversion of the whole matrix. They are based on the approximation of the solution of the linear system by computing matrix-vector multiplications. The resulting computations are efficient because the time-slice matrices and the right-hand sides are very sparse. It is worth noting that we need to wait more than one hour to obtain the value of the index for a network with 6000 nodes or more by using the original approach. We see that the method based on the iterative solution of the linear system is not only the fastest among the five, but tolerates well the change of dimension, making this approach a very good method for dealing with large networks.

As a second set of numerical experiments, we make use of the *triadic closure model* developed in [19]. Starting from an Erdős–Rényi network model [10] with a given edge density, we generate a sequence of M matrices in which the network at time point $k+1$ is built starting from the network at the previous time point. In particular, the expected value of $A^{[k+1]}$ given $A^{[k]}$ is

$$\mathcal{F}(A^{[k]}) = (1-\tilde{\omega})A^{[k]} + (\mathbf{1}^T\mathbf{1} - A^{[k]}) \circ (\delta\mathbf{1}^T\mathbf{1} + \epsilon(A^{[k]})^2),$$

where $\tilde{\omega} \in (0,1)$ is the death rate, $\delta\mathbf{1} + \epsilon(A^{[k]})^2$ is the birth rate with $0 < \delta \ll 1$ and $0 < \epsilon(n-2) < 1-\delta$, and $\mathbf{1}$ is the vector of all ones. This model is based on the social science hypothesis that "friends of friends" tend to become friends; that is, new edges are more likely between pairs of nodes that are separated by many paths of length 2.

Tables 3 and 4 display the results obtained by setting $\tilde{\omega} = \delta = 20/n^2$ and $\epsilon = 5/n^2$, where $A^{[1]}$ is an Erdős–Rényi network model with an edge density of 0.1 and 0.3, respectively. We report the execution time and the relative error between the approximate solution and the solution obtained by using the original approach. The results show that the computation based on the quadrature rules is ineffective in this case—NaN indicates that convergence was not attained. This can be explained by

TABLE 3

Execution time and relative error for the computation of the running broadcast communicabilities with $M = 10$ and $n = 2000, \ldots, 10000$. The network sequence is obtained from the triadic closure model.

	original	**quadrules**		**linsolv**	**iterative**		**lsqr**
n	time	time	err.	time	time	err.	time
2000	1.70e+02	1.08e+03	3.95e-01	8.67e+00	8.69e-01	2.32e-03	3.53e+00
4000	1.40e+03	7.13e+03	1.76e-01	4.27e+01	5.43e+00	3.92e-03	1.58e+01
6000	5.15e+03	2.35e+04	8.78e-01	1.10e+02	1.30e+01	3.87e-03	3.98e+01
8000	1.27e+04	5.91e+04	9.94e-01	2.22e+02	2.72e+01	5.02e-03	7.57e+01
10000	2.31e+04	1.05e+05	1.00e+00	1.38e+03	4.64e+01	4.90e-03	1.88e+02

TABLE 4

Execution time and relative error for the computation of the running broadcast communicabilities with $n = 1000$ and $M = 20, \ldots, 100$. The network sequence is obtained from the triadic closure model.

	original	**quadrules**		**linsolv**	**iterative**		**lsqr**
M	time	time	err.	time	time	err.	time
20	3.95e+01	1.60e+03	2.73e-01	2.54e+00	5.32e+00	5.53e-03	3.46e+00
40	8.27e+01	1.09e+04	NaN	8.82e+00	1.08e+01	7.53e-03	6.28e+00
60	1.31e+02	2.23e+04	NaN	1.90e+01	1.57e+01	9.75e-03	9.25e+00
80	1.63e+02	3.68e+04	NaN	3.29e+01	2.21e+01	1.17e-02	1.27e+01
100	2.17e+02	6.31e+04	NaN	4.59e+01	2.72e+01	6.69e-02	1.68e+01

taking into account the sparsity level of the matrices involved in the computation, which does not allow us to gain advantage from the use of matrix-vector products. On the contrary, the behavior of the methods based on the solution of the linear system is satisfactory, but again the sparsity level influences the performance of the iterative method based on the matrix splitting. This fact is more evident in Table 4, where we obtain comparable results from the methods based on the solution of the linear system.

The computed examples point out the effectiveness of the new block formulation relative to the original approach, especially when the dimension of the individual matrices is high. It is clear that *inverse-free* algorithms based on matrix-vector products are efficient for very sparse networks. Moreover, these kinds of algorithms work well on modern machines, since the computation can be fully parallelized.

5.2. New block formulation vs. supracentrality matrix. Having used the new block formulation (1.1) to develop efficient computational strategies, we now compare the relevance of the associated centrality measures with that of the supracentrality version (3.1). We first conduct a numerical test based on a synthetic time-dependent network. We generate the network in such a way that one node has a temporal connectivity pattern that allows it to initiate a disproportionate number of traversals. We note that this type of hierarchical pattern of interactions has been found, either explicitly or implicitly, in empirical studies of online behavior. For example, in the context of online forums, Graham and Wright [16] singled out *agenda-setters*, who are responsible for new thread creation, and thereby influence *subsequent* interactions, writing that "the inclusion of agenda-setting reflects our view that influence is not limited to the volume of posts alone." Huffaker et al. [23] discovered hierarchy within the use of chat features in a massively multiplayer online (MMO) role-playing game, and found that in general "players send messages to higher-level experts." It is therefore useful to have centrality tools that can discover and quantify this type of influence in the time-dependent setting.

To build a simple data set, we use $n = 200$ nodes and $M = 4$ time levels. We begin by setting each $A^{[k]}$ to be an independent, directed random graph where the probability of an edge from node i to node j at time k is given by $4/n$, independently of i, j, and k. In this way, each node has an expected out-degree of 4 at each time level and there is no structure to the interactions. We then remove all edges that emanate from node 1. Finally, we repeat the following procedure 16 times:

- at time level $k = 1$ connect node 1 to a uniformly chosen node, n_2;
- at time level $k = 2$ connect node n_2 to a uniformly chosen node, n_3;
- at time level $k = 3$ connect node n_3 to a uniformly chosen node, n_4;
- at time level $k = 4$ connect node n_4 to a uniformly chosen node, n_5.

In this way, node 1 is given 16 edges that are guaranteed to have a follow-on effect in terms of dynamic walks around the network. In the above construction, target nodes $n_1, n_2, \ldots$ are chosen uniformly and independently across $1, 2, \ldots, n$, and in a final processing step, repeated edges within a time level and self-loops are deleted.

The upper-left scatter plot in Figure 2 shows, for each node, the aggregate out-degree on the horizontal axis against the dynamic broadcast communicability, as given by the row sums of a normalized version, $\mathcal{S}^{[M]}/\|\mathcal{S}^{[M]}\|_2$, of the dynamic communicability matrix from (2.7). Here we used $\alpha = 0.9/\rho^\star$, where $\rho^\star = 4.2$ is the maximum spectral radius over the time levels, and $b = 1$ with $\Delta t = 1$. In this diagram, node 1 is highlighted with a star symbol. We see that, despite having only a typical aggregate out-degree, node 1 produces by far the highest communicability score, which

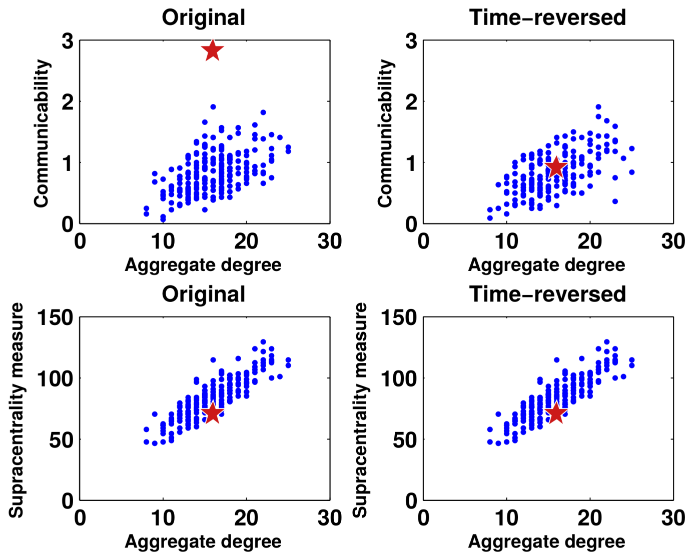

FIG. 2. *Node centrality scatter plots for a synthetic network. The special node,* 1*, is marked with a star. In each case the horizontal axis shows aggregate out-degree. The upper-left and lower-left diagrams use dynamic communicability and supracentrality-based marginal node centrality, respectively, for the vertical axis. The upper-right and lower-right diagrams repeat the experiment with the data in reverse time order. By construction, the supracentrality results are invariant to time reversal.*

reflects the fact that its edges have a knock-on effect through time. In the upper-right diagram in Figure 2, we repeat the test with the time levels taken in reverse order. In this case, the built-in dynamic walks *finish* at node 1, rather than starting there, and the benefit of these walks is now shared more evenly among the randomly chosen initial and intermediary nodes. The performance of node 1 is now more compatible with its aggregate out-degree and hence the dynamic communicability measure does not highlight any special structure.

In the same way, the lower-left and lower-right scatter plots in Figure 2 plot the aggregate out-degree against a nodal centrality measure based on the supracentrality matrix (3.1) for the original and time-reversed data, respectively. Here, we used static Katz centrality matrices along the diagonal, so $M^{[k]} = (I - \alpha A^{[k]})^{-1}$, with α chosen as in the first two experiments. To maintain compatibility we also used $\epsilon = e$. For our overall centrality measure, we again used the Katz resolvent, that is $(I - \widehat{\alpha}\mathbb{M})^{-1}$, with $\widehat{\alpha}$ chosen to be a factor 0.9 times the reciprocal of the spectral radius of $\mathbb{M}$. To obtain a single measure for each node, we used the *marginal node centrality* measure defined in [34]. We see that this type of centrality calculation, which does not maintain the time ordering, fails to highlight the role of node 1.

Next we use a set of simulated voice call data from the IEEE VAST 2008 Challenge [21]. This dynamic data set is designed to represent interactions between a controversial socio-political movement, and it incorporates some unusual temporal activity. The data involves $n = 400$ cell phone users, giving a complete set of time-stamped pairwise calls between them. Each call is logged via the send and receive nodes, a start time, and a duration in seconds. Among the extra information supplied by the competition designers was the strong suggestion that one node acts as

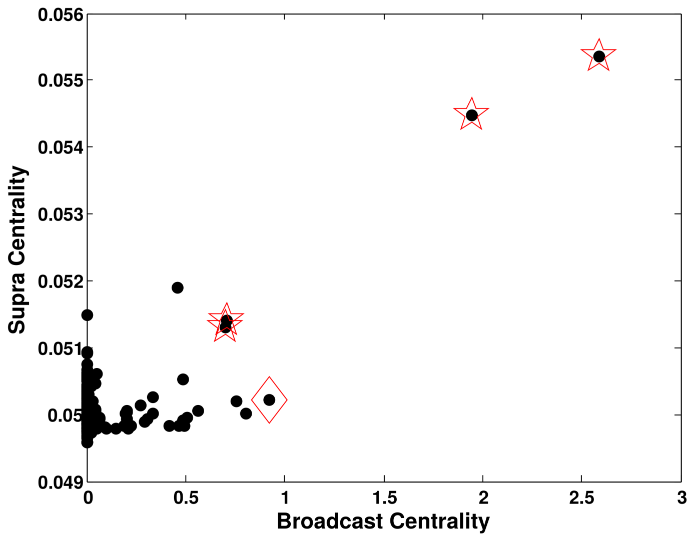

FIG. 3. *Scatter plot of broadcast and supracentrality-based centrality for a* 400 *node voice call network. Here, five particular nodes are known to be influential. The ringleader node is marked with a (red) diamond and four other inner circle nodes are marked with (red) stars.*

the "ringleader" within a key inner circle. Based on analyses submitted by challenge teams, we believe that this ringleader has ID 200, and the rest of the inner circle consists of four nodes with IDs 1, 2, 3, and 5. Further details can currently be found at http://www.cs.umd.edu/hcil/VASTchallenge08/index.htm.

This data was studied in terms of temporal centrality in [18], where it was shown that a continuous-time version of broadcast centrality can identify the key players, even though they are not the dominant users in terms of aggregate call time.

For our discrete-time experiment, we used 30-minute time windows over days 1 to 6. The symmetric adjacency matrix $A^{[k]}$ recorded whether nodes i and j spent any time interacting in the kth 30-minute time window. To compute the broadcast centrality (2.8), we took α to be a factor of 0.9 times the reciprocal of the maximum spectral radius of the $\rho(A^{[k]})$ over k, and $b = 0.1$ with $\Delta t = 1$. As in the previous experiment, we chose comparable parameters for the supracentrality matrix (3.1). Here, we used static Katz centrality matrices along the diagonal, so $M^{[k]} = (I - \alpha A^{[k]})^{-1}$ with the same α and with $\epsilon = e^b$. The overall centrality measure was then based on row sums of the Katz resolvent, $(I - \widehat{\alpha}\mathbb{M})^{-1}$, with $\widehat{\alpha}$ chosen to be a factor 0.9 times the reciprocal of the spectral radius of $\mathbb{M}$.

Figure 3 plots the broadcast centrality against the supracentrality-based measure. Here the ringleader node is marked with a red diamond and the other four inner circle nodes are marked with red five-pointed stars. We see that both centrality measures highlight two particular inner circle members and all four inner circle members appear in the top seven of both centrality rankings. However, for the ringleader, marked with a diamond, broadcast centrality ranks the node 3rd, whereas the supracentrality measure places this node 48th (out of 400 nodes). We conclude that, in this experiment, the time-respecting measure is better able to discover the importance of the ringleader node.

We note that these conclusions are consistent with the results in [29], where an algorithm was proposed to quantify the asymmetry caused by the arrow of time. Our computations also make it clear that, in the Katz setting, use of the supracentrality matrix also requires a third parameter to be chosen, for the resolvent system involving $\mathbb{M}$.

6. Conclusions.

This work focused on a context where a time-dependent sequence of networks is provided for a given set of nodes. Equivalently, we have an ordered sequence of adjacency matrices, or a three-dimensional tensor. Expressing the tensor as a large block matrix corresponds to representing the system as a single, static network in which nodes make multiple appearances. Such a representation has the advantage that a variety of computational approaches can be designed and tested. In particular, we found that an iterative method based on a regular matrix splitting was particularly effective. However, construction of the block matrix representation must be undertaken with care. In the case of extracting resolvent-based network centrality measures that are motivated through the concept of traversals through the network, we highlighted a block matrix structure that makes available time-respecting centralities and illustrated its practical advantages over an alternative formulation.

Interesting avenues for future work in this context include

- developing strategies for choosing algorithm parameters, a key example being the length of the time windows, where there is a trade-off between dimensionality and sparsity;
- considering matrix functions other than the resolvent;
- computing other walk-based centrality measures, such as total communicability [4] or hub-authority communicability [3];
- studying block matrix formulations of more general multilayer networks where time is one dimension of many.

Acknowledgment. The first author would like to thank the Department of Mathematics and Statistics at the University of Strathclyde for their hospitality during the time in which this work was initiated.

REFERENCES

[1] A. Alsayed and D. J. Higham, *Betweenness in time dependent networks*, Chaos Solitons Fractals, 72 (2015), pp. 35–48.
[2] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082.
[3] M. Benzi, E. Estrada, and C. Klymko, *Ranking hubs and authorities using matrix functions*, Linear Algebra Appl., 438 (2013), pp. 2447–2474.
[4] M. Benzi and C. Klymko, *Total communicability as a centrality measure*, J. Complex Netw., 1 (2013), pp. 124–149.
[5] S. Boccaletti, G. Bianconi, R. Criado, C. I. D. Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, *The structure and dynamics of multilayer networks*, Phys. Rep., 544 (2014), pp. 1–122.
[6] S. P. Borgatti, *Centrality and network flow*, Soc. Networks, 27 (2005), pp. 55–71.
[7] S. P. Borgatti and M. Everett, *A graph-theoretic framework for classifying centrality measures*, Soc. Networks, 28 (2006), pp. 466–484.
[8] D. Calvetti, L. Reichel, and F. Sgallari, *Application of anti-Gauss quadrature rules in linear algebra*, in Applications and Computation of Orthogonal Polynomials, W. Gautschi, G. H. Golub, and G. Opfer, eds., Birkhäuser, Basel, 1999, pp. 41–56.
[9] I. Chen, M. Benzi, H. H. Chang, and V. S. Hertzberg, *Dynamic communicability and epidemic spread: A case study on an empirical dynamic contact network*, J. Complex Netw. (2016), doi:10.1093/comnet/cnw017.
[10] P. Erdős and A. Rényi, *On random graphs*, Publ. Math. Debrecen, 6 (1959), pp. 290–297.
[11] E. Estrada and D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 696–714.

[12] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, *Block Gauss and anti-Gauss quadrature with application to networks*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1655–1684.

[13] L. Freeman, *Centrality in networks: I. Conceptual clarification*, Soc. Networks, 1 (1979), pp. 215–239.

[14] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, NJ, 2010.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 4th ed., Johns Hopkins University Press, Baltimore, MD, 2012.

[16] T. Graham and S. Wright, *Discursive equality and everyday talk online: The impact of "superparticipants"*, J. Comput. Mediat. Commun., 19 (2014), pp. 625–642.

[17] P. Grindrod and D. J. Higham, *A matrix iteration for dynamic network summaries*, SIAM Rev., 55 (2013), pp. 118–128.

[18] P. Grindrod and D. J. Higham, *A dynamical systems view of network centrality*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 470 (2014), 20130835.

[19] P. Grindrod, D. J. Higham, and M. C. Parsons, *Bistability through triadic closure*, Internet Math., 8 (2012), pp. 402–423.

[20] P. Grindrod, D. J. Higham, M. C. Parsons, and E. Estrada, *Communicability across evolving networks*, Phys. Rev. E, 83 (2011), 046120.

[21] G. G. Grinstein, C. Plaisant, S. J. Laskowski, T. O'Connell, J. Scholtz, and M. A. Whiting, *Vast* 2008 *challenge: Introducing mini-challenges*, in Proceedings of the 2008 IEEE Symposium on Visual Analytics Science and Technology, IEEE, 2008, pp. 195–196.

[22] P. Holme and J. Saramäki, *Temporal networks*, Phys. Rep., 519 (2012), pp. 97–125.

[23] D. Huffaker, J. Wang, J. Treem, M. Ahmad, L. Fullerton, D. Williams, M. Poole, and N. Contractor, *The social behaviors of experts in massive multiplayer online role-playing games*, in Proceedings of the 2009 International Conference on Computational Science and Engineering, IEEE, 2009, pp. 326–331.

[24] L. Katz, *A new index derived from sociometric data analysis*, Psychometrika, 18 (1953), pp. 39–43.

[25] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, *Multilayer networks*, J. Complex Netw., 2 (2014), pp. 203–271.

[26] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.

[27] P. Laflin, A. V. Mantzaris, P. Grindrod, F. Ainley, A. Otley, and D. J. Higham, *Discovering and validating influence in a dynamic online social network*, Soc. Netw. Anal. Min., 3 (2013), pp. 1311–1323.

[28] D. P. Laurie, *Anti-Gaussian quadrature formulas*, Math. Comp., 65 (1996), pp. 739–747.

[29] A. V. Mantzaris and D. J. Higham, *Asymmetry through time dependency*, Eur. Phys. J. B, 89 (2016), pp. 1–8.

[30] S. Ragnarsson and C. F. Van Loan, *Block tensor unfoldings*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 149–169.

[31] V. Sekara, A. Stopczynski, and S. Lehmann, *Fundamental structures of dynamic social networks*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. 9977–9982.

[32] L. Solá, M. Romance, R. Criado, J. Flores, A. García del Amo, and S. Boccaletti, *Eigenvector centrality of nodes in multiplex networks*, Chaos, 23 (2013), 033131.

[33] A. Taylor and D. J. Higham, *CONTEST: A controllable test matrix toolbox for MATLAB*, ACM Trans. Math. Software, 35 (2009), 26.

[34] D. Taylor, S. A. Myers, A. Clauset, M. A. Porter, and P. J. Mucha, *Eigenvector-based centrality measures for temporal networks*, SIAM J. Multiscale Model. Simul., 15, pp. 537–574, doi:10.1137/16M1066142.