Available online at www.sciencedirect.com
SCIENCE DIRECT®
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
ELSEVIER Journal of Computational and Applied Mathematics 167 (2004) 429–447
www.elsevier.com/locate/cam

The Kronecker product and stochastic automata networks

Amy N. Langville[a,*], William J. Stewart[b,1]

[a] Operations Research Program, N. Carolina State University, Raleigh, NC 27695-7913, USA
[b] Department of Computer Science, N. Carolina State University, Raleigh, N.C. 27695-8206, USA

Received 23 May 2003; received in revised form 12 September 2003

Abstract

This paper can be thought of as a companion paper to Van Loan's *The Ubiquitous Kronecker Product* paper (J. Comput. Appl. Math. 123 (2000) 85). We collect and catalog the most useful properties of the Kronecker product and present them in one place. We prove several new properties that we discovered in our search for a stochastic automata network preconditioner. We conclude by describing one application of the Kronecker product, omitted from Van Loan's list of applications, namely stochastic automata networks.

Keywords: Stochastic automata networks; Kronecker products; Kronecker product properties; Preconditioning

1. Introduction

Stochastic automata networks (SANs) have become an increasingly important modeling tool since the 1980s. SANs are used to efficiently model very large Markov chains whose state space is on the order of millions. The key to a SAN's ability to compactly and efficiently model such large Markov chains lies in their extensive use of the Kronecker product operation. In order to understand SANs and their advantages, one needs some familiarity with the Kronecker product. The first half of this paper (Section 2) is meant to provide such familiarity by collecting many of the known names, definitions, and properties of the Kronecker product. In addition, three new properties pertaining to the Kronecker product's compatibility with generalized inverses are proven in Section 2.6. After this theoretical introduction to the Kronecker product, we describe the practical uses of the Kronecker product, listing several applications of the operation, ranging from image processing and generalized spectral analysis to analysis of chess endgames and fast transform algorithms. The number of different

* Corresponding author. Tel.: +919-515-8146; fax: +919-515-3798.
E-mail addresses: anlangvi@eos.ncsu.edu (A.N. Langville), billy@csc.ncsu.edu (W.J. Stewart).
[1] Research supported in part by NSF (CCR-9731856).

0377-0427/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2003.10.010

uses of the Kronecker product has grown recently, prompting Charlie Van Loan to call the operation *ubiquitous*. His 2000 paper describes dozens of interesting applications [23]. We use the second half of this paper (Section 4) to add one more application of the Kronecker product to Van Loan's list: SANs. We use examples and a discussion of the solution methods for SANs to show the Kronecker product's connection to SAN modeling.

2. The Kronecker product

The operation defined by the symbol $\otimes$ was first used by Johann Georg Zehfuss in 1858 [5]. It has since been called by various names, including the Zehfuss product, the Producttransformation, the conjunction, the tensor product, the direct product and the Kronecker product. In the end, the Kronecker product stuck as the name for the symbol and operation, $\otimes$.

2.1. Definition of the Kronecker product

Definition. The *Kronecker product* of $A_{m_A \times n_A} \in \Re^{m_A \times n_A}$ and $B_{m_B \times n_B} \in \Re^{m_A \times n_A}$, written $A \otimes B$, is the tensor algebraic operation defined as

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n_A}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n_A}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m_A,1}B & a_{m_A,2}B & \dots & a_{m_A,n_A}B. \end{pmatrix}.$$

Each $a_{i,j}B$ is a block of size $m_B \times n_B$. $A \otimes B$ is of size $m_A m_B \times n_A n_B$. For example, if

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}, \quad B = \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \\ b_{4,1} & b_{4,2} \end{pmatrix},$$

$$\text{then } A \otimes B = \left(\begin{array}{cc|cc|cc} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} & a_{1,3}b_{1,1} & a_{1,3}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} & a_{1,3}b_{2,1} & a_{1,3}b_{2,2} \\ a_{1,1}b_{3,1} & a_{1,1}b_{3,2} & a_{1,2}b_{3,1} & a_{1,2}b_{3,2} & a_{1,3}b_{3,1} & a_{1,3}b_{3,2} \\ a_{1,1}b_{4,1} & a_{1,1}b_{4,2} & a_{1,2}b_{4,1} & a_{1,2}b_{4,2} & a_{1,3}b_{4,1} & a_{1,3}b_{4,2} \\ \hline a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} & a_{2,3}b_{1,1} & a_{2,3}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} & a_{2,3}b_{2,1} & a_{2,3}b_{2,2} \\ a_{2,1}b_{3,1} & a_{2,1}b_{3,2} & a_{2,2}b_{3,1} & a_{2,2}b_{3,2} & a_{2,3}b_{3,1} & a_{2,3}b_{3,2} \\ a_{2,1}b_{4,1} & a_{2,1}b_{4,2} & a_{2,2}b_{4,1} & a_{2,2}b_{4,2} & a_{2,3}b_{4,1} & a_{2,3}b_{4,2} \end{array}\right).$$

We also mention another Kronecker operation, the Kronecker sum, which is defined as the ordinary sum of Kronecker products. The Kronecker sum, $A \oplus B$, is defined by square matrices A and B and is given by

$$A \oplus B \triangleq A \otimes I_{n_B} + I_{n_A} \otimes B,$$

where n_A is the size of the square matrix A and n_B is the size of the square matrix B.

One advantage of Kronecker products is their compact representation. Consider the linear system $Cx = d$ in which C can be written as the Kronecker product of two much smaller matrices, A and B. The system $(A \otimes B)x = d$ can be solved quickly without ever forming the full matrix $C = A \otimes B$ (as is shown in Section 3); only the smaller matrices A and B need to be stored. An iterative method such as GMRES that uses only matrix–vector multiplications can be used to solve the compact system $(A \otimes B)x = d$ with the Kronecker product–vector multiplication algorithm [3]. Suppose $C_{10\,000\times 10\,000}$ can be expressed as the Kronecker product of $A_{100\times 100}$ and $B_{100\times 100}$. The linear system $Cx = d$ only requires the storage of two 100×100 matrices. In fact, later we will exploit properties of the Kronecker product to solve the special system $(A \otimes B)x = d$ very fast.

2.2. Properties of the Kronecker product

Before we can discuss some of the interesting applications of the Kronecker product, a complete background of its properties is required. These properties are divided into categories by topic. For example, the first four properties listed are basic Kronecker product properties, while the next three deal with structure.

2.3. Basic properties

Graham's book [4] lists the following properties (along with proofs) of the Kronecker product such as:

1. Associativity:

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C.$$

2. Distributivity over ordinary matrix addition:

$$(A + B) \otimes (C + D) = A \otimes C + B \otimes C + A \otimes D + B \otimes D.$$

3. Compatibility with ordinary matrix multiplication:

$$AB \otimes CD = (A \otimes C)(B \otimes D).$$

4. Compatibility with ordinary matrix inversion:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.$$

2.4. Structure and factorization properties

Van Loan's paper, [23], lists the following additional properties of the Kronecker product:

1. Compatibility with ordinary matrix transposition:
$$(A \otimes B)^{\mathrm{T}} = A^{\mathrm{T}} \otimes B^{\mathrm{T}}.$$
2. Structure theorems:
   (a) If A and B are nonsingular, then $A \otimes B$ is nonsingular.
   (b) If A and B are square lower (upper) triangular, then $A \otimes B$ is lower (upper) triangular.
   (c) If A and B are banded, then $A \otimes B$ is banded.
   (d) If A and B are symmetric, then $A \otimes B$ is symmetric.
   (e) If A and B are positive definite, then $A \otimes B$ is positive definite.
   (f) If A and B are stochastic, then $A \otimes B$ is stochastic.
   (g) If A and B are Toeplitz, then $A \otimes B$ is block Toeplitz.
   (h) If A and B are orthogonal, then $A \otimes B$ is orthogonal.
3. Factorizations:
   (a) LU: Let A be a square nonsingular matrix of order m_A with LU factorization $A = P_A^{\mathrm{T}} L_A U_A$ and B be a square nonsingular matrix of order m_B with LU factorization $B = P_B^{\mathrm{T}} L_B U_B$. Then
$$A \otimes B = (P_A^{\mathrm{T}} L_A U_A) \otimes (P_B^{\mathrm{T}} L_B U_B) = (P_A \otimes P_B)^{\mathrm{T}} (L_A \otimes L_B)(U_A \otimes U_B).$$
   (b) Cholesky: Let A be a positive definite matrix of order m_A with Cholesky factor G_A and B be a positive definite matrix of order m_B with Cholesky factor G_B. Then the Cholesky factorization of $A \otimes B$ is
$$A \otimes B = (G_A^{\mathrm{T}} G_A) \otimes (G_B^{\mathrm{T}} G_B) = (G_A \otimes G_B)^{\mathrm{T}} (G_A \otimes G_B).$$
   (c) QR: Let A be an $m_A \times n_A$ matrix with linearly independent columns and QR factorization $A = Q_A R_A$, where Q is an $m_A \times n_A$ matrix with orthonormal columns and R is an $n \times n$ upper triangular matrix. B is similarly defined with $B = Q_B R_B$ as its QR factorization. Then the QR factorization of $A \times B$ is
$$A \otimes B = (Q_A R_A) \otimes (Q_B R_B) = (Q_A \otimes Q_B)(R_A \otimes R_B).$$
   (d) Schur decomposition: Let A be a square matrix of order m_A with Schur decomposition $A = U_A T_A U_A^{\mathrm{T}}$, where U_A is unitary and T_A is upper triangular. Let B be a square matrix of order m_B with Schur decomposition $B = U_B T_B U_B^{\mathrm{T}}$, where U_B is unitary and T_B is upper triangular. Then the Schur decomposition of $A \otimes B$ is
$$A \otimes B = (U_A T_A U_A^{\mathrm{T}}) \otimes (U_B T_B U_B^{\mathrm{T}}) = (U_A \otimes U_B)(T_A \otimes T_B)(U_A \otimes U_B)^{\mathrm{T}}.$$
   (e) Singular value decomposition: Let A be an $m_A \times n_A$ matrix with singular value decomposition $U_A \Sigma_A V_A^{\mathrm{T}}$ and B be an $m_B \times n_B$ matrix with singular value decomposition $U_B \Sigma_B V_B^{\mathrm{T}}$. Let $\mathrm{rank}(A) = r_A$ and $\mathrm{rank}(B) = r_B$. Then $A \otimes B$ has rank $r_A r_B$ and singular value decomposition
$$A \otimes B = (U_A \Sigma_A V_A^{\mathrm{T}}) \otimes (U_B \Sigma_B V_B^{\mathrm{T}}) = (U_A \otimes U_B)(\Sigma_A \otimes \Sigma_B)(V_A \otimes V_B)^{\mathrm{T}}.$$

   Note: All of these factorizations of $C = A \otimes B$ merely require the factorizations of the small A and B matrices!

Most of the theorems have trivial proofs. Many of the proofs in this section can be found in [4,6,10,19,23].

2.5. Measure and numerical properties

Chapter 4 of the book by Horn and Johnson [6], contains a wealth of information on Kronecker products and their properties. Some of the more useful ones are listed below.

1. Trace: if A and B are square, then
$$\mathrm{tr}(A \otimes B) = \mathrm{tr}(A)\,\mathrm{tr}(B) = \mathrm{tr}(B \otimes A).$$
2. Norms: If A is $m_A \times n_A$ and B is $m_B \times n_B$, then for all p-norms
$$\|A \otimes B\| = \|A\|\,\|B\|.$$
3. Rank:
$$\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\mathrm{rank}(B).$$
4. Eigenvalues and eigenvectors:
For A and B square, let λ be a member of the spectrum of A. That is, $\lambda \in \sigma(A)$. Let x_A be a corresponding eigenvector of λ and let $\mu \in \sigma(B)$ and x_B be a corresponding eigenvector. Then $\lambda\mu \in \sigma(A \otimes B)$ and $x_A \otimes x_B$ is the corresponding eigenvector of $A \otimes B$. That is, every eigenvalue of $A \otimes B$ arises as a product of eigenvalues of A and B.
5. Singular values:
Let the $\mathrm{rank}(A) = r_A$ and $\mathrm{rank}(B) = r_B$. Then the nonzero singular values of $A \otimes B$ are the $r_A r_B$ positive numbers $\{\sigma_i(A)\sigma_j(B)\colon\ 1 \leqslant i \leqslant r_A,\ \ 1 \leqslant j \leqslant r_B\}$, where $\sigma_i(A)$ is the ith singular value of A.
6. Determinants: If A is $m \times m$ and B is $n \times n$ then
$$\det(A \otimes B) = [\det(A)]^n\,[\det(B)]^m.$$
7. Powers: If A and B are square then
$$(A \otimes B)^n = A^n \otimes B^n.$$

Below we prove or provide references to the proofs of each of the seven theorems above.

Proof of 1. Let A be $m \times m$ and B be $n \times n$.
$$\mathrm{tr}(A \otimes B) = \sum_{i=1}^{m} \mathrm{tr}(a_{i,i}B) = \sum_{i=1}^{m} a_{i,i}\,\mathrm{tr}(B) = \mathrm{tr}(B) \sum_{i=1}^{m} a_{i,i} = \mathrm{tr}(B)\,\mathrm{tr}(A). \qquad \square$$

Proof of 2. We begin by proving the Frobenius norm case, $\|A \otimes B\|_{\mathrm{F}} = \|A\|_{\mathrm{F}}|B\|_{\mathrm{F}}$.
$$\begin{aligned}\|A \otimes B\|_{\mathrm{F}}^2 &= \mathrm{tr}[(A \otimes B)(A \otimes B)^{\mathrm{T}}] = \mathrm{tr}[(A \otimes B)(A^{\mathrm{T}} \otimes B^{\mathrm{T}})] \\ &= \mathrm{tr}(AA^{\mathrm{T}} \otimes BB^{\mathrm{T}}) = \mathrm{tr}(AA^{\mathrm{T}})\,\mathrm{tr}(BB^{\mathrm{T}}) = \mathrm{tr}(A^{\mathrm{T}}A)\,\mathrm{tr}(B^{\mathrm{T}}B) \\ &= \|A\|_{\mathrm{F}}^2\|B\|_{\mathrm{F}}^2 = (\|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}})^2.\end{aligned}$$
Therefore, $\|A \otimes B\|_{\mathrm{F}} = \|A\|_{\mathrm{F}}\,\|B\|_{\mathrm{F}}$.

Now for the 2-norm;

$$\begin{aligned}\|A\|_2\|B\|_2 &= \sqrt{\lambda_{\max}(A)\lambda_{\max}(B)}\\ &= \sqrt{\lambda_{\max}(A\otimes B)} = \|A\otimes B\|_2.\end{aligned}$$

The 1-norm case,

$$\begin{aligned}\|A\otimes B\|_1 &= \max_{1\leqslant j_A\leqslant n_A}\sum_{i_A=1}^{m_A}|a_{i_Aj_A}B|\\ &= \max_{1\leqslant j_A\leqslant n_A, 1\leqslant j_B\leqslant n_B}\sum_{i_A=1}^{m_A}\sum_{i_B=1}^{m_B}|a_{i_Aj_A}b_{i_Bj_B}|\\ &= \max_{1\leqslant j_A\leqslant n_A}\sum_{i_A=1}^{m_A}|a_{i_Aj_A}|\max_{1\leqslant j_B\leqslant n_B}\sum_{i_B=1}^{m_B}|b_{i_Bj_B}|\\ &= \|A\|_1\|B\|_1.\end{aligned}$$

The ∞-norm is similar to the 1-norm except the largest absolute row sum is used rather than the largest absolute column sum. □

Proof of 3. Let A be $m\times n$ and B be $p\times q$. If $A=Q_AR_A$ and $B=Q_BR_B$ are the QR factorizations, where Q_A is $m\times n$ and Q_B is $p\times q$, then

$$\begin{aligned}\operatorname{rank}(A\otimes B) &= \operatorname{rank}(Q_AR_A\otimes Q_BR_B)\\ &= \operatorname{rank}((Q_A\otimes Q_B)(R_A\otimes R_B))\\ &= \operatorname{rank}(R_A\otimes R_B).\end{aligned}$$

Since R_A and R_B are both upper triangular, then $R_A\otimes R_B$ is upper triangular with upper triangular blocks. Let $\operatorname{rank}(R_A)=r_A$ and $\operatorname{rank}(R_B)=r_B$. Each row of blocks of size R_B has r_B nonzero rows. There are r_A nonzero rows of such blocks. Using this and the upper triangular structure of $R_A\otimes R_B$, we conclude that $\operatorname{rank}(R_A\otimes R_B)=r_Ar_B$. Therefore, $\operatorname{rank}(A\otimes B)=\operatorname{rank}(R_A\otimes R_B)=r_Ar_B=\operatorname{rank}(R_A)\operatorname{rank}(R_B)=\operatorname{rank}(A)\operatorname{rank}(B)$. □

Proofs of 4 and 5. Statement of these theorems and their corresponding proofs can be found in Chapter 4 of the book by Horn and Johnson [6]. □

Proof of 6. Let A be $m\times m$ and B be $n\times n$. A determinant for an $n\times n$ matrix G can be determined from an LU factorization with pivoting [11]. Then $\det(G)=\sigma_G\ u_{G_{1,1}}u_{G_{2,2}}\cdots u_{G_{n,n}}=\sigma_G\operatorname{diagprod}(U_G)$, where $\sigma_G=+1$ if an even number of row interchanges are used to obtain P_G and -1 if an odd number of row interchanges are used to obtain P_G. With $\sigma_{AB}=\sigma_A\sigma_B$;

$$\begin{aligned}\det(A\otimes B) &= \sigma_{AB}\operatorname{diagprod}(U_A\otimes U_B)\\ &= \sigma_{AB}(u_{A_{1,1}})^n\operatorname{diagprod}(U_B)(u_{A_{2,2}})^n\operatorname{diagprod}(U_B)\cdots(u_{A_{m,m}})^n\operatorname{diagprod}(U_B)\\ &= \sigma_{AB}(\operatorname{diagprod}(U_A))^n(\operatorname{diagprod}(U_B))^m\\ &= [\det(A)]^n[\det(B)]^m. \qquad □\end{aligned}$$

Proof of 7. Proof by induction on n. Base case ($n=2$):

$$(A \otimes B)^2 = (A \otimes B)(A \otimes B) = A^2 \otimes B^2.$$

Induction step: Assume $(A \otimes B)^n = A^n \otimes B^n$. Show $(A \otimes B)^{n+1} = A^{n+1} \otimes B^{n+1}$.

$$(A \otimes B)^{n+1} = (A \otimes B)^n (A \otimes B) = (A^n \otimes B^n)(A \otimes B) = A^{n+1} \otimes B^{n+1}. \qquad \square$$

2.6. Pseudoinverse properties

We prove three new properties of the Kronecker product in our work with Markov chains and their preconditioners. To our knowledge, these properties have not been stated or proven elsewhere. Before we state our new theorems and proofs, we define some generalized inverses: the Drazin inverse, the group inverse and the Moore–Penrose pseudoinverse [2,11,12].

- If A is an $n \times n$ singular matrix of index k such that $\operatorname{rank}(A^k) = r$, then there exists a nonsingular matrix Q such that

  $$Q^{-1}AQ = \begin{pmatrix} C_{r\times r} & 0 \\ 0 & N \end{pmatrix},$$

  where C is nonsingular and N is nilpotent of index k. The *Drazin inverse* of A, denoted by A^{D}, is given as

  $$A^{\mathrm{D}} = Q \begin{pmatrix} C^{-1} & 0 \\ 0 & 0 \end{pmatrix} Q^{-1}.$$

- The group inverse is a special case of the Drazin inverse and applies when the index of the matrix A is 1. The group inverse is appropriate for singular $n \times n$ matrices of rank $n-1$ and is denoted by $A^{\#}$.
- If A is an $m \times n$ matrix of rank r, then there exist orthogonal matrices $U_{m\times m}$ and $V_{n\times n}$ such that

  $$A = URV^{\mathrm{T}} = U \begin{pmatrix} C_{r\times r} & 0 \\ 0 & 0 \end{pmatrix}_{m\times n} V^{\mathrm{T}}.$$

  The *Moore–Penrose pseudoinverse* of A, denoted by $A^{\dagger}$ is the $n \times m$ matrix given as

  $$A^{\dagger} = V \begin{pmatrix} C^{-1} & 0 \\ 0 & 0 \end{pmatrix}_{n\times m} U^{\mathrm{T}}.$$

With these definitions we are now ready to state our new theorems.

1. Condition number: For all matrix norms,

   $$\operatorname{cond}(A \otimes B) = \operatorname{cond}(A)\operatorname{cond}(B).$$

2. Compatibility with the Drazin inverse and the group inverse:

   $$(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} \otimes \cdots \otimes A_n^{\mathrm{D}}.$$

   $$(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^{\#} = A_1^{\#} \otimes A_2^{\#} \otimes \cdots \otimes A_n^{\#}.$$

3. Compatibility with the Moore–Penrose pseudoinverse

$$(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^\dagger = A_1^\dagger \otimes A_2^\dagger \otimes \cdots \otimes A_n^\dagger.$$

Proof of 1. *Case* 1: If A and B are nonsingular, then

$$\begin{aligned}\mathrm{cond}(A \otimes B) &= \|A \otimes B\|\,\|(A \otimes B)^{-1}\| \\ &= \|A \otimes B\|\,\|A^{-1} \otimes B^{-1}\| \\ &= \|A\|\,\|B\|\,\|A^{-1}\|\,\|B^{-1}\| \\ &= \mathrm{cond}(A)\,\mathrm{cond}(B).\end{aligned}$$

Case 2: If A and B are singular, then $A \otimes B$ is singular and

$$\begin{aligned}\mathrm{cond}(A \otimes B) &= \|A \otimes B\|\,\|(A \otimes B)^{\dagger}\| \\ &= \|A \otimes B\|\,\|A^{\dagger} \otimes B^{\dagger}\| \\ &= \|A\|\,\|B\|\,\|A^{\dagger}\|\,\|B^{\dagger}\| \\ &= \mathrm{cond}(A)\,\mathrm{cond}(B).\end{aligned}$$

Case 3: If A is nonsingular and B is singular, then $A \otimes B$ is singular and

$$\begin{aligned}\mathrm{cond}(A \otimes B) &= \|A \otimes B\|\,\|(A \otimes B)^{\dagger}\| \\ &= \|A \otimes B\|\,\|A^{\dagger} \otimes B^{\dagger}\| \\ &= \|A\|\,\|B\|\,\|A^{-1}\|\,\|B^{\dagger}\| \\ &= \mathrm{cond}(A)\,\mathrm{cond}(B),\end{aligned}$$

since $A^\dagger = A^{-1}$ for A nonsingular. □

Proof of 2 (Proof by induction on n). We begin with a proof of the base case, $(A_1 \otimes A_2)^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}}$. We derive a different expression for the right-hand side, then show that the left-hand side can also be written this way. Every square singular matrix A_1 of order n_{A_1} can be decomposed as

$$A_1 = P_{A_1} \begin{pmatrix} C_{A_1} & 0 \\ 0 & N_{A_1} \end{pmatrix} P_{A_1}^{-1},$$

where C_{A_1} is nonsingular of size $r_{A_1} \times r_{A_1}$, N_{A_1} is nilpotent of index k_{A_1} and $\mathrm{rank}(A_1^{k_{A_1}}) = r_{A_1}$. Similarly, a square matrix A_2 of order n_{A_2} can be written as

$$A_2 = P_{A_2} \begin{pmatrix} C_{A_2} & 0 \\ 0 & N_{A_2} \end{pmatrix} P_{A_2}^{-1},$$

again where C_{A_2} is nonsingular of size $r_{A_2} \times r_{A_2}$, N_{A_2} is nilpotent of index k_{A_2} and $\operatorname{rank}(A_2^{k_{A_2}}) = r_{A_2}$. According to the definition of the Drazin inverse,

$$A_1^{\mathrm{D}} = P_{A_1} \begin{pmatrix} C_{A_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{A_1}^{-1}$$

and likewise

$$A_2^{\mathrm{D}} = P_{A_2} \begin{pmatrix} C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{A_2}^{-1}.$$

Thus,

$$\begin{aligned} A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} &= \left[P_{A_1} \begin{pmatrix} C_{A_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{A_1}^{-1} \right] \otimes \left[P_{A_2} \begin{pmatrix} C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{A_2}^{-1} \right] \\ &= (P_{A_1} \otimes P_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} (P_{A_1}^{-1} \otimes P_{A_2}^{-1}) \\ &= (P_{A_1} \otimes P_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (P_{A_1}^{-1} \otimes P_{A_2}^{-1}). \end{aligned}$$

Now we show that $(A_1 \otimes A_2)^{\mathrm{D}}$ is equal to the above expression

$$\begin{aligned} (A_1 \otimes A_2)^{\mathrm{D}} &= \left\{ \left[P_{A_1} \begin{pmatrix} C_{A_1} & 0 \\ 0 & N_{A_1} \end{pmatrix} P_{A_1}^{-1} \right] \otimes \left[P_{A_2} \begin{pmatrix} C_{A_2} & 0 \\ 0 & N_{A_2} \end{pmatrix} P_{A_2}^{-1} \right] \right\}^{\mathrm{D}} \\ &= \left\{ (P_{A_1} \otimes P_{A_2}) \left[\begin{pmatrix} C_{A_1} & 0 \\ 0 & N_{A_1} \end{pmatrix} \otimes \begin{pmatrix} C_{A_2} & 0 \\ 0 & N_{A_2} \end{pmatrix} \right] (P_{A_1}^{-1} \otimes P_{A_2}^{-1}) \right\}^{\mathrm{D}} \\ &= \left\{ (P_{A_1} \otimes P_{A_2}) \begin{pmatrix} C_{A_1} \otimes C_{A_2} & 0 & 0 & 0 \\ 0 & C_{A_1} \otimes N_{A_2} & 0 & 0 \\ 0 & 0 & N_{A_1} \otimes C_{A_2} & 0 \\ 0 & 0 & 0 & N_{A_1} \otimes N_{A_2} \end{pmatrix} (P_{A_1}^{-1} \otimes P_{A_2}^{-1}) \right\}^{\mathrm{D}}. \end{aligned}$$

Let N_L be the 3×3 principal submatrix of the middle matrix above. N_L is nilpotent of index $k = \max\{k_{A_1}, k_{A_2}\}$. That is,

$$N_L^k = \begin{pmatrix} C_{A_1} \otimes N_{A_2} & 0 & 0 \\ 0 & N_{A_1} \otimes C_{A_2} & 0 \\ 0 & 0 & N_{A_1} \otimes N_{A_2} \end{pmatrix}^k = 0.$$

This follows from $(C_{A_1} \otimes N_{A_2})^k = C_{A_1}^k \otimes N_{A_2}^k = C_{A_1}^k \otimes 0 = 0$, and similarly for the other two diagonal blocks. Using this fact,

$$(A_1 \otimes A_2)^{\mathrm{D}} = \left\{ (P_{A_1} \otimes P_{A_2}) \begin{pmatrix} C_{A_1} \otimes C_{A_2} & 0 \\ 0 & N_L \end{pmatrix} (P_{A_1}^{-1} \otimes P_{A_2}^{-1}) \right\}^{\mathrm{D}}.$$

Then with the definition of the Drazin inverse and the fact that $(C_{A_1} \otimes C_{A_2})^{-1} = C_{A_1}^{-1} \otimes C_{A_2}^{-1}$, the base case is complete.

$$(A_1 \otimes A_2)^{\mathrm{D}} = (P_{A_1} \otimes P_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (P_{A_1}^{-1} \otimes P_{A_2}^{-1}) = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}}.$$

The base case has been established: $(A_1 \otimes A_2)^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}}$. In the induction hypothesis, we assume that $(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} \otimes \cdots \otimes A_n^{\mathrm{D}}$ and show that $(A_1 \otimes A_2 \otimes \cdots \otimes A_n \otimes A_{n+1})^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} \otimes \cdots \otimes A_n^{\mathrm{D}} \otimes A_{n+1}^{\mathrm{D}}$. Let $Y = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. Then $Y^{\mathrm{D}} = (A_1 \otimes A_2 \otimes \cdots \otimes A_n)^{\mathrm{D}} = A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} \otimes \cdots \otimes A_n^{\mathrm{D}}$ by the induction step. And

$$\begin{aligned} (A_1 \otimes A_2 \otimes \cdots \otimes A_n \otimes A_{n+1})^{\mathrm{D}} &= (Y \otimes A_{n+1})^{\mathrm{D}} \\ &= Y^{\mathrm{D}} \otimes A_{n+1}^{\mathrm{D}} \\ &= A_1^{\mathrm{D}} \otimes A_2^{\mathrm{D}} \otimes \ldots \otimes A_n^{\mathrm{D}} \otimes A_{n+1}^{\mathrm{D}}. \end{aligned}$$

The group inverse of A, denoted by $A^{\#}$, is a special case of the Drazin inverse for singular square matrices with index 1. Thus,

$$(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^{\#} = A_1^{\#} \otimes A_2^{\#} \otimes \cdots \otimes A_n^{\#}. \qquad \square$$

Proof of 3 (Proof by induction on n). We begin with a proof of the base case, $(A_1 \otimes A_2)^{\dagger} = A_1^{\dagger} \otimes A_2^{\dagger}$. We derive a different expression for the right-hand side, then show that the left-hand side can also be written this way. Every real $m_{A_1} \times n_{A_1}$ matrix A_1 has a *URV* factorization

$$A_1 = U_{A_1} \begin{pmatrix} C_{A_1} & 0 \\ 0 & 0 \end{pmatrix} V_{A_1}^{\mathrm{T}},$$

where the orthogonal matrices U_{A_1} and V_{A_1} are of order m_{A_1} and n_{A_1}, respectively, the nonsingular matrix C_{A_1} is size $r_{A_1} \times r_{A_1}$ and $r_{A_1} = \mathrm{rank}(A_1)$. Similarly, a real $m_{A_2} \times n_{A_2}$ matrix A_2 can be written as

$$A_2 = U_{A_2} \begin{pmatrix} C_{A_2} & 0 \\ 0 & 0 \end{pmatrix} V_{A_2}^{\mathrm{T}},$$

where the orthogonal matrices U_{A_2} and V_{A_2} have size $m_{A_2} \times m_{A_2}$ and $n_{A_2} \times n_{A_2}$, respectively, the nonsingular matrix C_{A_2} has size $r_{A_2} \times r_{A_2}$ and $r_{A_2} = \mathrm{rank}(A_2)$. The definition of the Moore–Penrose pseudoinverse of A_1, denoted $A_1^{\dagger}$, gives

$$A_1^{\dagger} = V_{A_1} \begin{pmatrix} C_{A_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U_{A_1}^{\mathrm{T}}$$

and likewise,

$$A_2^{\dagger} = V_{A_2} \begin{pmatrix} C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U_{A_2}^{\mathrm{T}}.$$

Thus,

$$\begin{aligned} A_1^{\dagger} \otimes A_2^{\dagger} &= \left[V_{A_1} \begin{pmatrix} C_{A_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U_{A_1}^{\mathrm{T}} \right] \otimes \left[V_{A_2} \begin{pmatrix} C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U_{A_2}^{\mathrm{T}} \right] \\ &= (V_{A_1} \otimes V_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} (U_{A_1}^{\mathrm{T}} \otimes U_{A_2}^{\mathrm{T}}) \\ &= (V_{A_1} \otimes V_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (U_{A_1}^{\mathrm{T}} \otimes U_{A_2}^{\mathrm{T}}). \end{aligned}$$

Now we show that $(A_1 \otimes A_2)^{\dagger}$ is equal to the above expression.

$$\begin{aligned} (A_1 \otimes A_2)^{\dagger} &= \left\{ \left[U_{A_1} \begin{pmatrix} C_{A_1} & 0 \\ 0 & 0 \end{pmatrix} V_{A_1}^{\mathrm{T}} \right] \otimes \left[U_{A_2} \begin{pmatrix} C_{A_2} & 0 \\ 0 & 0 \end{pmatrix} V_{A_2}^{\mathrm{T}} \right] \right\}^{\dagger} \\ &= \left\{ (U_{A_1} \otimes U_{A_2}) \left[\begin{pmatrix} C_{A_1} & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} C_{A_2} & 0 \\ 0 & 0 \end{pmatrix} \right] (V_{A_1}^{\mathrm{T}} \otimes V_{A_2}^{\mathrm{T}}) \right\}^{\dagger} \\ &= \left\{ (U_{A_1} \otimes U_{A_2}) \begin{pmatrix} C_{A_1} \otimes C_{A_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} (V_{A_1} \otimes V_{A_2})^{\mathrm{T}} \right\}^{\dagger} \\ &= \left\{ (V_{A_1} \otimes V_{A_2}) \begin{pmatrix} (C_{A_1} \otimes C_{A_2})^{-1} & 0 \\ 0 & 0 \end{pmatrix} (U_{A_1} \otimes U_{A_2})^{\mathrm{T}} \right\} \\ &= \left\{ (V_{A_1} \otimes V_{A_2}) \begin{pmatrix} C_{A_1}^{-1} \otimes C_{A_2}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (U_{A_1} \otimes U_{A_2})^{\mathrm{T}} \right\}. \end{aligned}$$

The base case has been established. $(A_1 \otimes A_2)^\dagger = A_1^\dagger \otimes A_2^\dagger$. By the induction hypothesis, we have $(A_1 \otimes A_2 \otimes \cdots \otimes A_n)^\dagger = A_1^\dagger \otimes A_2^\dagger \otimes \cdots \otimes A_n^\dagger$ and show that $(A_1 \otimes A_2 \otimes \cdots \otimes A_n \otimes A_{n+1})^\dagger = A_1^\dagger \otimes A_2^\dagger \otimes \cdots \otimes A_n^\dagger \otimes A_{n+1}^\dagger$. Let $Y = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. Then $Y^\dagger = (A_1 \otimes A_2 \otimes \cdots \otimes A_n)^\dagger = A_1^\dagger \otimes A_2^\dagger \otimes \cdots \otimes A_n^\dagger$ by the induction step. Finally,

$$\begin{aligned}(A_1 \otimes A_2 \otimes \cdots \otimes A_n \otimes A_{n+1})^\dagger &= (Y \otimes A_{n+1})^\dagger \\ &= Y^\dagger \otimes A_{n+1}^\dagger \\ &= A_1^\dagger \otimes A_2^\dagger \otimes \cdots \otimes A_n^\dagger \otimes A_{n+1}^\dagger. \qquad \square\end{aligned}$$

3. Applications of properties of Kronecker products

To demonstrate the usefulness of applying these properties of Kronecker products, we return to the linear system problem, $(A \otimes B)x = d$. Let A be $m \times m$ and B be $m \times m$. Property 3(a) of Section 2.2.2 regarding LU factorizations can be exploited. If $A \otimes B$ is nonsingular then a solution exists and the system can be written as

$$(L_A \otimes L_B)(U_A \otimes U_B)x = d,$$

where $P_A A = L_A U_A$ and $P_B B = L_B U_B$. First the lower triangular system $(L_A \otimes L_B)z = (P_A \otimes P_B)d$ is solved by forward substitution in $O(m^3)$ time. Then $(U_A \otimes U_B)x = z$ is solved by back substitution in $O(m^3)$ time. Without exploiting the structure, Gaussian elimination requires $O(m^6)$ arithmetic operations. The Kronecker structure also avoids the formation of $m^2 \times m^2$ matrices; only the smaller L_A, L_B, U_A, U_B are needed.

For example, consider the forward substitution $(L_A \otimes L_B)z = d$, where A and B are 3×3 matrices and z and d are 9×1 vectors. To simplify the notation, we assume P_A and P_B are identity matrices.

$$\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_9 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_9 \end{pmatrix}.$$

Then,

$$L_A \otimes L_B = \left(\begin{array}{ccc|ccc|ccc} a_{11}b_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{11}b_{21} & a_{11}b_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{11}b_{31} & a_{11}b_{32} & a_{11}b_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline a_{21}b_{11} & 0 & 0 & a_{22}b_{11} & 0 & 0 & 0 & 0 & 0 \\ a_{21}b_{21} & a_{21}b_{22} & 0 & a_{22}b_{21} & a_{22}b_{22} & 0 & 0 & 0 & 0 \\ a_{21}b_{31} & a_{21}b_{32} & a_{11}b_{33} & a_{22}b_{31} & a_{22}b_{32} & a_{22}b_{33} & 0 & 0 & 0 \\ \hline a_{31}b_{11} & 0 & 0 & a_{32}b_{11} & 0 & 0 & a_{33}b_{11} & 0 & 0 \\ a_{31}b_{21} & a_{31}b_{22} & 0 & a_{32}b_{21} & a_{32}b_{22} & 0 & a_{33}b_{21} & a_{33}b_{22} & 0 \\ a_{31}b_{31} & a_{31}b_{32} & a_{31}b_{33} & a_{32}b_{31} & a_{32}b_{32} & a_{32}b_{33} & a_{33}b_{31} & a_{33}b_{32} & a_{33}b_{33} \end{array}\right),$$

which is a unit lower triangular matrix with lower triangular blocks. The first $m=3$ equations of this 9×9 system represent a lower triangular matrix and can be solved in $O(m^2)$ arithmetic operations:

$$a_{11}b_{11}z_1 = d_1,$$

$$a_{11}b_{21}z_1 + a_{11}b_{22}z_2 = d_2,$$

$$a_{11}b_{31}z_1 + a_{11}b_{32}z_2 + a_{11}b_{33}z_3 = d_3.$$

Now the next three equations are:

$$\mathbf{a_{21}b_{11}z_1} + a_{22}b_{11}z_4 = d_4,$$

$$\mathbf{a_{21}b_{21}z_1} + \mathbf{a_{21}b_{22}z_2} + a_{22}b_{21}z_4 + a_{22}b_{22}z_5 = d_5,$$

$$\mathbf{a_{21}b_{31}z_1} + \mathbf{a_{21}b_{32}z_2} + \mathbf{a_{21}b_{33}z_3} + a_{22}b_{31}z_4 + a_{22}b_{32}z_5 + a_{22}b_{33}z_6 = d_6.$$

The boldface expression in the first equation, $\mathbf{a_{21}b_{11}z_1}$, can be computed as $a_{21}d_1/a_{11}$. The second bold-face expression, $\mathbf{a_{21}b_{21}z_1} + \mathbf{a_{21}b_{22}z_2}$, is just $a_{21}d_2/a_{11}$, while the third expression, $\mathbf{a_{21}b_{31}z_1} + \mathbf{a_{21}b_{32}z_2} + \mathbf{a_{21}b_{33}z_3}$, is $a_{21}d_3/a_{11}$. We use the previous expressions for obtaining z_1, z_2 and z_3 in the first set of equations to simplify the second set of three equations. The simplified second set of equations becomes

$$a_{22}b_{11}z_4 = d_4 - \frac{\mathbf{a_{21}d_1}}{\mathbf{a_{11}}},$$

$$a_{22}b_{21}z_4 + a_{22}b_{22}z_5 = d_5 - \frac{\mathbf{a_{21}d_2}}{\mathbf{a_{11}}},$$

$$a_{22}b_{31}z_4 + a_{22}b_{32}z_5 + a_{22}b_{33}z_6 = d_6 - \frac{\mathbf{a_{21}d_3}}{\mathbf{a_{11}}}.$$

Solving the second set of equations takes $O(m)$ arithmetic operations and the forward solve step takes $O(m^2)$ operations, so obtaining z_4, z_5 and z_6 takes $O(m^2)$ time. This simplification and using the work from the previous solution step continues so that solving each of the m sets of m equations takes $O(m^2)$ time, resulting in an overall solution time of $O(m^3)$. Exploiting the Kronecker structure reduces the usual, expected $O(m^4)$ time to solve $(L_A \otimes L_B)z = d$ to $O(m^3)$ time.

One final note regarding the exploitation of the Kronecker structure of the linear system remains. Suppose the matrices A and B are of different sizes. Then, the time required to solve the linear system $(A \otimes B)x = d$ is $O(m_A m_B^2)$, where m_A is the size of A and m_B is the size of B.

Van Loan's paper [23] provides a thorough catalog of further applications of the Kronecker product. We briefly mention a few here. One application receiving growing interest is semidefinite programming. Due to the surge of work on interior point methods, the solution to systems involving the symmetric Kronecker product has been studied recently. The Kronecker product appears in numerous types of least squares problems; one example is the problem of surface fitting with splines. Kronecker products have also been used to unify the field of fast transforms such as the fast Fourier transform, the Hartley transform, and fast wavelet transforms. The Kronecker product plays an instrumental role in many image restoration algorithms. The Kronecker product has also been used to form approximate inverse preconditioners, an application we emphasize in Section 4. One application of the Kronecker product not found in Van Loan's paper is SANs. The remainder of this paper deals with SANs and their connection to the Kronecker product. First, we define SANs, then we discuss their preconditioning problems.

4. SANs

Markov chains can be used to model many physical systems. For example, Markov chains are used frequently to answer performance questions about parallel and distributed computer systems. While Markov chains provide accurate measures of the system, the size of the Markov chain can quickly grow to an enormous and even intractable size. Storing the state space and infinitesimal generator matrix Q for such large Markov chains (on the order of millions) has become a bottleneck. One remedy for this storage problem is SANs, which store the infinitesimal generator of the Markov chain in compact form using Kronecker products. SANs [14] are particularly applicable to parallel and distributed computer systems. The reason for this will become clear after we define SANs.

A SAN consists of several individual stochastic automata which act independently for the most part. Occasionally, these automata may need to coordinate their actions, thus connecting the individual automata in a network of automata which depend on one another. Each individual automaton $A^{(i)}$ has a number of states associated with it. $A^{(i)}$ also has a number of rules which determine its movement from one state to the next. The state of any automaton at time t is the state it occupies at time t. The state of the collective SAN at time t is the state of each of its corresponding automata. Fig. 1 gives the high-level representation of a SAN.

Automaton $A^{(1)}$ contains 4 states, $A^{(2)}$ contains 4 states and $A^{(3)}$ contains 3 states. The current state of each automaton is denoted by the shaded circle. Thus the current state of the SAN is denoted by all three shaded circles. The line connecting $A^{(1)}$ and $A^{(2)}$ represents the interaction between these

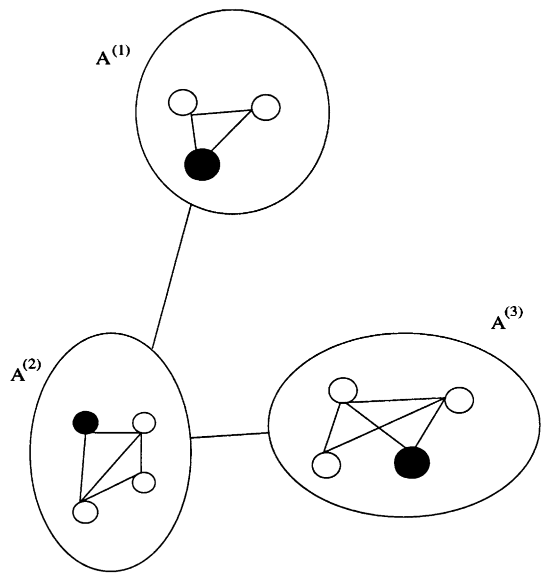

Fig. 1. Stochastic automata network.

two automata. Somehow $A^{(1)}$ and $A^{(2)}$ need to coordinate their actions. Exactly how they might need to do this will be described later. If the lines connecting the automata were not present in the diagram then $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$ would be completely independent systems. $A^{(1)}$'s stochastic behavior could be modeled with a separate Markov chain from $A^{(2)}$ and so on. Thus, SANs are only useful for automata which have some interaction. However, too much interaction among the automata can complicate the SAN to the point that its use is questionable. Clearly, SANs should be restricted to systems with appropriate infrequent interaction.

There are two general ways in which these automata may interact with one another. One concerns the transitions themselves and the other concerns the actual transition rates. First, the global state may change when a transition occurs. Transitions can be either local or synchronizing. Local transitions only affect the corresponding automaton. When an automaton has a local transition, it moves from one of its states to another of its states. Synchronizing transitions are not local. They affect the global state by changing the state of several automata. A synchronizing transition occurs when one automaton enables a transition to occur in two or more other automata.

The second type of interaction draws another distinction between transitions. They can be either constant or functional. A functional transition occurs when an automaton's transition rate is a function of the state of another automaton. Transitions that are not functional are called constant. Constant or functional transitions, unlike synchronizing transitions, affect only the local automata involved. Note that synchronizing transitions may be constant or functional. This information regarding the automata and their types of transitions provides all the information needed to formally define a SAN, as Atif and Plateau have done [16]. While this infrequent interaction (synchronizing transitions and functional transition rates) does complicate SANs, Plateau and her coworkers have shown that the SAN can still be represented in compact form as a sum of Kronecker products, known as the SAN descriptor [14,17,20].

Plateau and Fourneau [17] have shown that SANs with N automata and E synchronizing events should be handled by separating out the local transitions for each automata and writing this local effect as the ordinary sum of N Kronecker products with each Kronecker product involving N smaller matrices. Then the effect of the synchronizing events is added. Each synchronizing event requires two more Kronecker products of N matrices. Thus the infinitesimal generator of a SAN can always be written as

$$Q = \sum_{j=1}^{2E+N} \bigotimes_{i=1}^{N} Q_j^{(i)}.$$

When the infinitesimal generator matrix Q is written and stored in the form $\sum_{j=1}^{2E+N} \otimes_{i=1}^{N} Q_j^{(i)}$, this is called the *SAN descriptor*. The reader should note that the SAN descriptor is the *sum* of Kronecker products.

So far, we have only discussed the effect of synchronizing events on the structure of this SAN descriptor. In summary, we learned that there are two more terms in the descriptor for each synchronizing event. An increasing number of synchronizing events increases the complexity of the SAN model, which is why SANs are restricted to systems with appropriate infrequent interaction. We now mention the effect of functional transitions on the SAN descriptor. By extending the ordinary Kronecker product to the generalized Kronecker product [16,17], the SAN descriptor can still be written as above, but the elements of the $Q_j^{(i)}$ matrices may now be functions. These functional entries require that the appropriate numerical values be computed and substituted each time the functional rate is needed. Thus, functional transitions do not change the structure of the SAN descriptor, but they do add complexity and a computational burden.

In practice, the modeler works from the SAN system and forms and stores each of the $Q_j^{(i)}$ matrices following the rules given in [17,20,21]. We emphasize that the global infinitesimal generator matrix Q is never formed or stored. Herein lies the storage-saving capacity of the SAN formalism. Consider a collection of four automata each of size 100. Suppose the infrequent interaction among these four automata is described by two synchronizing events and there are no functional transitions. The global Q is size 10^8 but only $2E+N=2\times 2+4=8$ sparse matrices of size 100 need to be stored, thanks to the Kronecker product!

4.1. Stationary analysis of a SAN

The computation of the stationary solution π of a continuous-time ergodic Markov chain involves solving the linear system $\pi Q=0$ and $\pi e^{\mathrm{T}}=1$, where Q is the infinitesimal generator of the Markov chain and e represents the unit row vector. Q is singular with rank $n-1$. Thus, finding the stationary solution of a continuous-time Markov chain can be viewed as a linear system problem. Another way to view the same problem is as an eigenvalue problem. P is the transition probability matrix associated with the same system. In fact, $P=I+\Delta t Q$ where $\Delta t \leqslant 1/\max|q_{ii}|$. P is a stochastic matrix with a unit eigenvalue. Then finding the stationary solution π involves solving $\pi=\pi P$, which is an eigenvalue problem. Now this eigenvalue problem can be used to define the power method, an iterative method for finding π by computing iterates with

$$x^{(k+1)}=x^{(k)}P.$$

With a suitable initial iterate $x^{(0)}$, $x^{(k+1)}$ will converge to the eigenvector π which can then be normalized so that π contains the stationary solution.

Very large Markov chains are often represented as SANs using the SAN descriptor in place of Q. Namely, $Q=\sum_{j=1}^{\mathrm{T}}\otimes_{i=1}^{N}Q_j^{(i)}$, where $T=2E+N$, E is the number of synchronizing events and N is the number of automata. Since $P=I+\Delta t Q$, then in the SAN formalism,

$$P=I+\Delta t Q=\bigotimes_{i=1}^{N} I_{n_i}+\sum_{j=1}^{\mathrm{T}}\Delta t\bigotimes_{i=1}^{N} Q_j^{(i)}$$

and the power method for SANs can be written as

$$x^{(k+1)}=x^{(k)}(I+\Delta t Q)=x^{(k)}+\Delta t x^{(k)}\left(\sum_{j=1}^{\mathrm{T}}\bigotimes_{i=1}^{N} Q_j^{(i)}\right).$$

The power method is the simplest of all iterative methods for finding the stationary solution vector π. The Jacobi, Gauss–Seidel and SOR method are three more iterative methods used for solving linear systems, such as our homogeneous linear system $\pi Q=0$. Yet these methods are based on splittings of the transition matrix and thus are not easily transferable to the SAN formalism. Another class of iterative methods is that of projection methods. These methods approximate an exact solution (in our case, the stationary solution) by building better and better approximations which are taken from small-dimension subspaces. Some popular projection methods are Arnoldi, GMRES, CGS, BiCGSTAB and QMR. Such projections methods can be and have been applied to SANs [1,15,22]. In fact, any iterative or projection method which involves a matrix–vector multiply can be used to find the stationary distribution of a SAN. In place of the matrix–vector multiply, the Kronecker product–vector multiplication algorithm invented by Fernandes and his coworkers [3] can be used. Direct methods for solving linear systems, such as those based on LU decompositions, are not immediately amenable to SANs because the SAN's compact descriptor representation of the generator matrix precludes easy access to the L, U factors. Furthermore, SANs are used as a compact, alternative representation for very large Markov models. The size of such models makes direct methods impractical [20].

4.2. Preconditioning for SAN

It is well known that the iterative methods discussed above perform better when preconditioners are used. The convergence of an iterative method depends on the eigenvalues of the system. Any iterative method can converge slowly if the eigenvalue distribution is undesirable for that method. For example, when the subdominant eigenvalue of the iteration matrix is close to the dominant eigenvalue (which is 1 for our transition matrices P), the power method converges slowly. Thus the goal of preconditioning is to modify the eigenvalue distribution of the iteration matrix so that convergence is improved while the solution remains unchanged.

In general, for the linear system $Ax = b$, we introduce the preconditioning matrix M, so that $MAx = Mb$. We hope that M is a good approximation of A^{-1} and thus convergence will be rapid.

For Markov chain problems, the preconditioned power method becomes

$$x^{(k+1)} = x^{(k)}(I - (I - P)M).$$

Since the matrix $(I - P)$ is singular with rank$(n - 1)$, we choose M to be a good approximation of the group inverse of $(I - P)$, written as $(I - P)^{\#}$. Thus for SANs, the preconditioned power method is

$$\begin{aligned}
x^{(k+1)} &= x^{(k)}(I - (I - (I + \Delta t Q))M) \\
&= x^{(k)}(I + \Delta t QM) \\
&= x^{(k)} + \Delta t x^{(k)} QM \\
&= x^{(k)} + \Delta t x^{(k)} \left(\sum_{j=1}^{\mathrm{T}} \bigotimes_{i=1}^{N} Q_j^{(i)} \right) M.
\end{aligned}$$

The problem now becomes that of finding a suitable preconditioner M that fits nicely into the SAN formalism. A popular set of preconditioners, ILU preconditioners, have largely been dismissed from consideration. The problem with adapting ILU preconditioners to SANs (for use in an iterative method, like the preconditioned power method) is that they are based on incomplete LU factorizations of the transition matrix. SANs store the transition matrix information as a sum of Kronecker products. And thus, an LU factorization of a SAN descriptor is not easily accessible.

Numerous other preconditioners have been proposed for SANs but each has been unsuccessful [1,18,22]. Recently, we discovered a nearest Kronecker product (NKP) preconditioner for SANs

[8]. The initial results for the NKP preconditioner look promising [7,9]. Our NKP preconditioner is derived from Pitsianis and Van Loan's work on approximation with Kronecker products. They discovered a method for finding the NKP, $A \otimes B$, for a general matrix R [13]. Since $A \otimes B \approx R$, one would hope that $A^{-1} \otimes B^{-1} \approx R^{-1}$. They took $A^{-1} \otimes B^{-1} = M$ as the preconditioner and tested this on a small example. Their Kronecker preconditioner compared favorably with many other preconditioners. This sparked us to try to extend this to find a suitable SAN preconditioner. For our case of Markov chains, we want to approximate $Q^{\#}$ rather than Q^{-1}. However, the algorithm for finding the A and B almost always results in a nonsingular A and nonsingular B. Thus, we must use the standard inverses, A^{-1} and B^{-1}, to form the preconditioner. In effect, we are using the ideal preconditioner $M = A^{-1} \otimes B^{-1}$ for a nearby system whose coefficient matrix $\hat{Q}$ is almost Q. Finding the small A^{-1} and B^{-1} matrices is not too difficult and the approximation is good for many matrices with nice structure. The advantage of the Kronecker approximation for SANs is that M need never be formed, instead only A^{-1} and B^{-1} need to be stored and used in the vector-Kronecker product multiplication of the iterative methods. In fact, we were able to extend Pitsianis and Van Loan's work to find any number of smaller matrices whose Kronecker product approximates the original matrix Q [8]. Thus, we can find $A, B, \ldots, N$ such that $A \otimes B \otimes \cdots \otimes N \approx Q$. We take $M = A^{-1} \otimes B^{-1} \otimes \cdots \otimes N^{-1}$ as our NKP preconditioner for SANs. Our initial battery of tests of the NKP preconditioner on SANs reports good results [9]. In fact, the NKP SAN preconditioner outperforms all other current preconditioners. We would like to remind the reader that the properties and power of the Kronecker product made this discovery of a SAN preconditioner possible.

5. Conclusion

The use and power of the Kronecker product is indeed ubiquitous as Van Loan [23] has suggested. In this paper, we have gathered and cataloged the most useful properties of the Kronecker product and we also added several new properties to this list. We then used these properties to describe a new application for the Kronecker product, SANs.

References

[1] P. Buchholz, Projection methods for the analysis of stochastic automata networks, in: B. Plateau, W.J. Stewart, M. Silva (Eds.), Numerical Solution of Markov Chains, Prensas Universitarias de Zaragoza, 1999, pp. 149–168.
[2] S.L. Campbell, C.D. Meyer, Generalized Inverses of Linear Transformations, Pitman Publishing, London, 1979.
[3] P. Fernandes, B. Plateau, W.J. Stewart, Efficient descriptor-vector multiplications in stochastic automata networks, J. Assoc. Comput. Mach. 45 (1998) 381–414.
[4] A. Graham, Kronecker Products and Matrix Calculus with Applications, Wiley, New York, 1981.
[5] H.V. Henderson, F. Pukelsheim, S.R. Searle, On the history of the Kronecker product, Linear and Multilinear Algebra 14 (1983) 113–120.
[6] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1991.
[7] A.N. Langville, Preconditioning for stochastic automata networks, Ph.D. Thesis, Operations Research Program, North Carolina State University, May 2002.
[8] A.N. Langville, W.J. Stewart, A Kronecker product approximate inverse preconditioner for SANs, J. Numer. Linear Algebra Appl., to appear.
[9] A.N. Langville, W.J. Stewart, Testing the NKP preconditioner on MCs and SANs, INFORMS J. Comput., to appear.
[10] C.C. MacDuffee, The Theory of Matrices, Chelsea, New York, 1946.

[11] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, Philadelphia, 2000.
[12] C.D. Meyer, The role of the group generalized inverse in the theory of finite Markov chains, SIAM Rev. 17 (1975) 443–464.
[13] N. Pitsianis, C. Van Loan, Approximation with Kronecker products, in: M.S. Moonen, G.H. Golub (Eds.), Linear Algebra for Large Scale and Real Time Applications, Kluwer Academic Publishers, Dordrecht, 1993, pp. 293–314.
[14] B. Plateau, On the stochastic structure of parallelism and synchronization models for distributed algorithms, Performance Evaluation Rev. 13 (1985) 142–154.
[15] B. Plateau, PEPS: a package for solving complex Markov models of parallel systems, in: R. Puigjaner, D. Potier (Eds.), Modelling Techniques and Tools for Computer Performance Evaluation, Plenum Press, New York, 1990, pp. 291–306.
[16] B. Plateau, K. Atif, Stochastic automata network for modelling parallel systems, IEEE Trans. Software Eng. 17 (10) (1991) 1093–1108.
[17] B. Plateau, J.M. Fourneau, A methodology for solving Markov models of parallel systems, J. Parallel Distributed Comput. 12 (1991) 370–387.
[18] B. Plateau, W.J. Stewart, P. Fernandes, On the benefits of using functional transitions in Kronecker modelling, Performance Evaluation (2001), submitted.
[19] W-H. Steeb, Kronecker Product of Matrices and Applications, Wissenschaftsverlag, Mannheim, 1991.
[20] W.J. Stewart, Introduction to the Numerical Solution of Markov Chains, Princeton University Press, Princeton, NJ, 1994.
[21] W.J. Stewart, Stochastic automata networks, in: W.K. Grassmann (Ed.), Computational Probability, Kluwer Academic Press, Boston, 2000, pp. 113–152.
[22] W.J. Stewart, K. Atif, B. Plateau, The numerical solution of stochastic automata networks, European J. Oper. Res. 86 (1995) 503–525.
[23] C.F. Van Loan, The ubiquitous Kronecker product, J. Comput. Appl. Math. 123 (2000) 85–100.