

C.1

# **Matrix Methods in Queueing and Dynamic Programming**

by

**Bernard Fernand Lamond**

**B.Sc., Université de Montréal, 1978**

**M.Sc., Université de Montréal, 1980**

**A Thesis Submitted in Partial Fulfillment of**

**the Requirements for the Degree of**

**DOCTOR OF PHILOSOPHY**

in

**The Faculty of Graduate Studies**

**Faculty of Commerce and Business Administration**

**We accept this thesis as conforming**

**to the required standard**

**The University of British Columbia**

**December 1985**

**© Bernard Fernand Lamond, 1985**

84

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Commerce

The University of British Columbia  
1956 Main Mall  
Vancouver, Canada  
V6T 1Y3

Date March 12, 1986

## Abstract

We investigate some modern matrix methods for the solution of finite state stochastic models with an infinite time horizon. Markov and semi-Markov decision processes and finite queues in tandem with exponential service times are considered. The methods are based on the Drazin generalized inverse and use matrix decomposition.

Unlike the related Jordan canonical form, the decompositions considered are numerically tractable and use real arithmetic when the original matrix has real entries. The spectral structure of the transition matrix of a Markov chain, deduced from non-negative matrix theory, provides a decomposition from which the limiting and deviation matrices are directly obtained.

The matrix decomposition approach to the solution of Markov reward processes provides a new, simple derivation of the Laurent expansion of the resolvent. Many other basic results of dynamic programming are easily derived in a similar fashion and the extension to semi-Markov decision processes is straightforward. Further, numerical algorithms for matrix decomposition can be used efficiently in the policy iteration method, for evaluating the terms of the Laurent series.

The problem of finding the stationary distribution of a system with two finite queues in tandem, when the service times have an exponential distribution, can also be expressed in matrix form. Two numerical methods, one iterative and one using matrix decomposition, are reviewed for computing the stationary probabilities. Job-local-balance is used to derive some bounds on the call congestion. A numerical investigation of the bounds is included. It suggests that the bounds are insensitive to the distribution of the service times.

## Table of Contents

Abstract . . . . .	ii
List of Tables . . . . .	v
List of Figures . . . . .	vi
Acknowledgements . . . . .	viii
Introduction . . . . .	1
I. Matrix decomposition and Drazin generalized inverse . . . . .	3
1.1. Review of linear algebra . . . . .	4
1.2. The Drazin generalized inverse . . . . .	14
1.3. Matrix decomposition algorithms . . . . .	20
1.4. Solution of singular systems of equations . . . . .	36
1.5. Other algorithms for Drazin inversion . . . . .	42
II. Nonnegative matrix theory . . . . .	44
2.1. Matrices and directed graphs . . . . .	45
2.2. Eigenvalue structure of a stochastic matrix . . . . .	49
2.3. Limiting and fundamental matrices . . . . .	53
2.4. Geometric convergence in the aperiodic case . . . . .	59
III. Discrete time Markov decision processes . . . . .	63
3.1. Markov reward processes . . . . .	64
3.2. Singular systems of equations . . . . .	68
3.3. Computation of gain, bias and other terms . . . . .	77

3.4.	Iterative computation of gain and bias . . . . .	78
3.5.	Markov decision processes . . . . .	82
IV.	Continuous time Markov decision processes . . . . .	91
4.1.	Continuous time Markov reward processes . . . . .	91
4.2.	Algebraic structure of the resolvent . . . . .	94
4.3.	Laurent expansion of the discounted rewards . . . . .	96
4.4.	Equivalent Markov decision processes . . . . .	98
V.	Semi-Markov decision processes . . . . .	102
5.1.	Semi-Markov reward processes . . . . .	103
5.2.	Decomposition of the first moment . . . . .	106
5.3.	Series expansion of the discounted rewards . . . . .	112
5.4.	Semi-Markov decision processes . . . . .	116
5.5.	Three special cases . . . . .	118
VI.	Stationary characteristics of tandem queues . . . . .	124
6.1.	Global balance equations . . . . .	126
6.2.	Computation of the stationary probabilities . . . . .	128
6.3.	The matrix decomposition method . . . . .	134
6.4.	Job local balance and bounds . . . . .	138
6.5.	Performance results . . . . .	144
	Conclusion . . . . .	153
	Bibliography . . . . .	154

## List of Tables

6.1.1. Transition rates out of state $(n_1, n_2)$ . . . . .	127
6.5.1. Numerical values with $\lambda = 1$ , varying $\mu_1$ and $\mu_2$ . . . . .	145
6.5.2. Numerical values with $\lambda = \mu_1 = \mu_2 = 1$ . . . . .	146
6.5.3. Comparison with hyperexponential service times . . . . .	151

## List of Figures

1.1.1. Directed graphs of $J_k(\Lambda)$ and $L_{2k}(\lambda)$ . . . . .	12
1.3.1. Compute $x = A^{-1}b$ , where $A = PLU$ . . . . .	21
1.3.2. Operations counts for Gaussian elimination . . . . .	23
1.3.3. The slow decomposition algorithm . . . . .	26
1.3.4. Operations count for the slow decomposition algo- rithm . . . . .	28
1.3.5. The fast decomposition algorithm . . . . .	33
1.3.6. Storage considerations for the fast algorithm . . . . .	34
1.4.1. The slow solution algorithm . . . . .	37
1.4.2. Algorithm to compute $u_1 = B_{11}^{-1}x_1$ . . . . .	40
1.4.3. The fast solution algorithm . . . . .	40
2.1.1. The directed graph $G(A)$ . . . . .	45
2.1.2. Directed graph of a periodic matrix . . . . .	48
3.2.1. Directed graph of $C(2)$ . . . . .	75
3.4.1. Power method for gain and bias . . . . .	79
3.5.1. Policy iteration method for $\rho$ -optimal policy . . . . .	89
3.5.2. Policy iteration method for $k$ discount optimal policy . . . . .	89
6.2.1. The power method for stationary probabilities . . . . .	132
6.2.2. Pascal procedures for transition rates . . . . .	133
6.3.1. Unstable algorithm for stationary probabilities . . . . .	136

6.3.2. Modified algorithm for stationary probabilities . . . . .	138
6.4.1. Failure of job local balance . . . . .	140
6.5.1. Blocking probability vs service rates . . . . .	147
6.5.2. Blocking probability contour curves . . . . .	149

## Acknowledgements

Some of the material in Chapters II and III is adapted from Lamond and Puterman (1985), and some of the material in Chapter VI is adapted from van Dijk and Lamond (1985).

I am grateful to my research advisor, Martin L. Puterman, for his encouragement and patience in reading numerous drafts of all parts of this thesis.

My work on numerical algorithms has benefited from discussions with A. Laurie Johnston at the beginning, and more recently from comments by Uri Ascher.

My interest in queueing has been stimulated by my joint work with Nico van Dijk, and by frequent discussions with Shelby L. Brumelle.

Recent comments by Uriel G. Rothblum and Arthur F. Veinott, Jr. have helped clarify some important ideas concerning the use of generalized inverses in Markov decision processes.

The numerical algorithms discussed in this thesis have been coded in Pascal. The text of the thesis, as well as the tables and most figures were produced with  $\text{\TeX}$ . The figures involving curves were produced with Tell-a-Graf.

Part of this research was done while the author was receiving an NSERC postgraduate scholarship, and later an SSHRC doctoral fellowship. Generous support from NSERC Grant A-5527 is also gratefully acknowledged.

**To my parents**

## Introduction

The six chapters of this thesis can be grouped into three distinct parts. The first part, which consists of Chapters I and II, deals with matrix decomposition. The second part contains Chapters III, IV and V. It explores the applications of the matrix decomposition approach to Markov and semi-Markov decision processes. Finally, Chapter VI is concerned with tandem queueing systems.

In Chapter I, a brief review of matrix algebra is given, concentrating on the Jordan canonical form. For a singular matrix, the generalized inverse of Drazin (1958) is presented. A significant improvement of the algorithm of Wilkinson (1982) is obtained, for computing the Drazin inverse of a matrix of index 1. Our fast algorithm uses only one third as many arithmetic operations as the recent shuffle algorithm of Anstreicher and Rothblum (1985).

In Chapter II, the structure of the Jordan canonical form of a stochastic matrix is derived from nonnegative matrix theory, using results of Varga (1962). The matrix decomposition approach is then applied to study the limiting and fundamental matrices of Kemeny and Snell (1960), and the deviation matrix of Veinott (1969) and (1974). The approach used is inspired from Campbell and Meyer (1979).

In Chapter III, the matrix decomposition results are applied to discrete time Markov decision processes. A new proof of many results of Blackwell (1962) and Veinott (1969) and (1974) is given. In particular, a direct derivation of the Laurent expansion of the resolvent, for a small interest rate, is given. The matrix decomposition algorithm of Chapter I is used in a new policy evaluation method.

In Chapter IV, the formulation of a continuous time decision process is reviewed, based on the continuous time Markov chain of Çinlar (1975). Matrix decomposition is used to show the similarity between continuous time and discrete time processes. An equivalence considered by Howard (1960), Veinott (1969), Schweitzer (1971) and Serfozo (1979) is reviewed.

In Chapter V, semi-Markov decision processes are considered. Matrix decomposition is used to obtain a new derivation of many results of Denardo (1971). In particular, the decomposition of the first moment of the transition probability function is derived. A new solution algorithm is also obtained. The models of Chapters III and IV are formulated as special cases. Also, an equivalence used by Ross (1970) is discussed.

In Chapter VI, a queueing system with two service stations in series is formulated, using the notation of Disney and König (1985). The numerical computation of the stationary probabilities is investigated. The iterative power method and the matrix decomposition method of Wong et al. (1977) are reviewed. Bounds are introduced, based on results in Hordijk and Van Dijk (1981) and the job-local-balance property. A numerical investigation of the bounds is included.

This research suggests that matrix decomposition is a powerful, as well as versatile, approach to the study and solution of finite state stochastic systems. Not only does it provide simple derivations and efficient algorithms, it also appears to unify the treatment of many models.

## Chapter I

### Matrix decomposition and Drazin generalized inverse

In this chapter, we present the mathematical basis for the subsequent chapters. The notions of Jordan canonical form and Drazin generalized inverse are reviewed, and some efficient algorithms are developed, to compute a matrix decomposition and use it as a data transformation to produce the solution of a singular system of equations.

In Section 1.1, we review the Jordan canonical form and related well known results. Our approach is similar to that of Golub and Wilkinson (1976), and reflects the general approach that will be followed throughout the thesis, although we will not be concerned with actually computing the Jordan canonical form.

Further, as this is to be applied to matrices with real entries, we also introduce a modified Jordan canonical form which involves only real arithmetic, but still retains the basic properties of the original Jordan canonical form. This leads to a new, direct proof of the existence of a real decomposition of a matrix, in a form that provides a suitable representation of the Drazin inverse. The idea of replacing a pair of complex conjugate eigenvalues by a  $2 \times 2$  matrix has been exploited in numerical analysis for computing the eigenvalues of a real matrix by the QR method [see e.g., Golub and Van Loan (1983), Section 7.4].

In Section 1.2, we review the definition of the generalized inverse of Drazin (1958). We derive its representation using a matrix decomposition based on the modified Jordan canonical form. This derivation follows very much that of Wilkinson (1982).

In Section 1.3, we review the Gaussian elimination algorithm for computing the

PLU decomposition of a matrix [see e.g., Wilkinson (1965)]. Then we specialize the method of Wilkinson (1982) for computing the Drazin inverse of a general matrix, to the special case when the matrix has an index of 1, which is the case of interest in Markov decision processes.

The main original contribution here is the fast algorithm, which produces a representation of the Drazin inverse, using essentially the same amount of arithmetic as for the *PLU* decomposition of a nonsingular matrix. In fact, its operations count is one third that of the recent shuffle algorithm of Anstreicher and Rothblum (1985) and, in the context of dynamic programming, it is essentially the same as for the method of Howard (1960). Our algorithm, however, is well suited for use in the lexicographic policy iteration method of Miller and Veinott (1969), as we will see in Chapter III.

In Section 1.4, it is shown how to use the matrix decompositions to efficiently compute the solution vectors. We call these vectors gain and bias, in anticipation of the dynamic programming applications of Chapter III.

Finally, Section 1.5 reviews some other algorithms which have been studied in the literature.

### 1.1. Review of linear algebra

Let  $x_1, x_2, \dots, x_m$  be  $n$ -vectors with complex entries. If the system of equations

$$\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_m x_m = 0$$

has the unique solution  $\alpha_1 = \dots = \alpha_m = 0$ , then the vectors are said to be *linearly independent*. Otherwise, they are *linearly dependent* and those with  $\alpha_i \neq 0$  can be expressed as a linear combination of the others.

A complex  $n \times n$  matrix  $A$  is said to be *singular* if its columns are linearly dependent. Otherwise,  $A$  is said to be *nonsingular*. It is well known that  $A$  is singular iff its *determinant*, denoted  $\det(A)$ , is equal to zero. Let  $b$  be a given  $n$ -vector, called the

*right-hand side*. Then the system of linear equations

$$Ax = b \tag{1.1.1}$$

has a unique solution  $x = A^{-1}b$  if the matrix  $A$  is nonsingular. The matrix  $A^{-1}$  is uniquely determined by  $A$  and is called its *inverse*. Hence a nonsingular matrix is also said to be *invertible*.

On the other hand, if  $A$  is singular, the system (1.1.1) does not necessarily have a solution for every right-hand side vector  $b$ . A vector  $b$  for which there is a solution is said to be *compatible* with  $A$ . A system of equations with singular  $A$  and compatible  $b$  has infinitely many solutions.

The algebraic structure of a complex  $n \times n$  matrix  $A$  is described by its eigenvalues and its eigenvectors. A *right-hand eigenvector* of  $A$  is a nonzero column vector  $x$  such that  $Ax = \lambda x$ , for some complex number  $\lambda$ . This number  $\lambda$  is the *eigenvalue* of  $A$  associated with  $x$ . Similarly, a *left-hand eigenvector* of  $A$  is a nonzero row vector  $y$  such that  $yA = \lambda y$ . Let  $A'$  be the transposed of the matrix  $A$ . Then  $y'$  is a right-hand eigenvector of  $A'$ . For simplicity, we will use the term *eigenvector* (alone) to denote a right-hand eigenvector.

The above definition implies that an eigenvector  $x$  is any nonzero solution of the system of linear equations

$$(\lambda I - A)x = 0, \tag{1.1.2}$$

where  $I$  denotes the  $n \times n$  identity matrix. This system has a nonzero solution  $x$  iff the matrix  $(\lambda I - A)$  is singular, that is, iff

$$\det(\lambda I - A) = 0.$$

Since  $\det(\lambda I - A)$  is a polynomial of degree  $n$  in  $\lambda$ ,  $A$  has at least one and at most  $n$  distinct eigenvalues. Hence Equation (1.1.2) always has a nonzero solution.

Let  $S$  be a nonsingular matrix and  $S^{-1}$  its inverse. Then a matrix  $B$  such that

$$B = S^{-1}AS \tag{1.1.3}$$

is said to be *similar* to  $A$ , and equation (1.1.3) is a *similarity transformation*. Much of the material in this thesis will be concerned with similarity transformations which transform a given matrix  $A$  into a new matrix  $B$  with a more convenient structure. An important property of similarity transformations is that they preserve the eigenvalues of a matrix. Indeed, we have

$$\lambda I - B = \lambda S^{-1}IS - S^{-1}AS = S^{-1}(\lambda I - A)S,$$

so that  $\det(\lambda I - B) = 0$  iff  $\det(\lambda I - A) = 0$ . In the context of linear systems of equations, a similarity transformation can be seen as a change of variables. Let  $y = S^{-1}x$  and  $c = S^{-1}b$ . Then  $Ax = b$  iff  $By = c$ . This last property is important. It says that  $A$  and  $b$  are compatible iff  $B$  and  $c$  are compatible. When a solution exists, then  $x$  is a solution of  $Ax = b$  iff  $y$  is a solution of  $By = c$ . This fact will be used extensively in our discussion of Markov decision processes.

One structure of much interest is that of a *diagonal* matrix. We denote by  $D = \text{diag}(d_i)$  a matrix with diagonal entries  $d_i$ ,  $i = 1, \dots, n$ . The following well known lemma indicates when a matrix  $A$  is similar to a diagonal matrix.

**Lemma 1.1.1.** An  $n \times n$  matrix  $A$  is similar to a diagonal matrix iff it has  $n$  linearly independent eigenvectors.

**Proof.** Let  $x_1, \dots, x_n$  be  $n$  linearly independent eigenvectors associated each with an eigenvalue  $\lambda_i$ ,  $i = 1, \dots, n$ . Then the matrix  $X = (x_1, \dots, x_n)$  has an inverse  $X^{-1}$ . Hence we have  $AX = X\Lambda$ , where  $\Lambda = \text{diag}(\lambda_i)$ , so that  $X^{-1}AX = \Lambda$ . This proves the “if” part. For the “only if” part, suppose that  $S^{-1}AS = D$ , for some matrices  $S$  and  $D = \text{diag}(d_i)$ . Then the  $n$  columns of  $S$  are linearly independent, and  $AS = SD$  implies that the  $i$ -th column of  $S$  is an eigenvector of  $A$  associated with the eigenvalue  $d_i$ ,  $i = 1, \dots, n$ . □

For example, the matrix

$$A = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$

has two linearly independent eigenvectors

$$x_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } x_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

with eigenvalues  $\lambda_1 = 2$  and  $\lambda_2 = 4$ , respectively, so that

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}.$$

Observe that a matrix satisfying Lemma 1.1.1 may have an eigenvalue with multiplicity greater than 1, that is,  $\lambda_i = \lambda_j$  for some  $i \neq j$ . Some matrices have fewer than  $n$  linearly independent eigenvectors. Some have only one. Such a matrix is bound to have at least one eigenvalue with multiplicity greater than 1. An example is the *Jordan submatrix of order  $n$* , denoted by  $J_n(\lambda)$ , with all diagonal entries equal to  $\lambda$  and subdiagonal entries equal to 1. All other entries are zero. For example,  $J_1(\lambda) = (\lambda)$  is diagonal and

$$J_4(\lambda) = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 1 & \lambda & 0 & 0 \\ 0 & 1 & \lambda & 0 \\ 0 & 0 & 1 & \lambda \end{pmatrix}.$$

It is easy to verify that any eigenvector of  $J_n(\lambda)$  is a nonzero multiple of  $e_n$ , the  $n$ -th unit vector. The following results illustrate how the Jordan submatrix provides the building blocks for a general matrix decomposition.

**Lemma 1.1.2.** Suppose that  $A$  is a singular, complex  $n \times n$  matrix. Suppose also that there are  $k$  nonzero vectors  $x_1, \dots, x_k$  such that  $Ax_1 = 0$  (with  $x_1 \neq 0$ ) and

$$Ax_i = x_{i-1}, \quad i = 2, \dots, k, \tag{1.1.4}$$

where  $1 \leq k \leq n$ . Then  $x_1, \dots, x_k$  are linearly independent.

**Proof.** By induction on  $k$ . The result is trivially true for  $k = 1$  (because such an  $x_1$  exists since  $A$  is singular). Now suppose  $x_1, \dots, x_{k-1}$  are linearly independent and that  $Ax_k = x_{k-1}$  for some nonzero vector  $x_k$ . If

$$x_k = \alpha_1 x_1 + \dots + \alpha_{k-1} x_{k-1}$$

then

$$\begin{aligned} Ax_k &= \alpha_1 Ax_1 + \alpha_2 Ax_2 + \dots + \alpha_{k-1} Ax_{k-1} \\ &= 0 + \alpha_2 x_1 + \dots + \alpha_{k-1} x_{k-2} \\ &\neq x_{k-1} \end{aligned}$$

because  $x_1, \dots, x_{k-1}$  are linearly independent. Hence  $x_k$  may not be a linear combination of  $x_1, \dots, x_{k-1}$  in order to be a solution of Equation (1.1.4), and the result follows.  $\square$

**Lemma 1.1.3.** Suppose  $A$  is a complex  $n \times n$  matrix, and let

$$S = (x_1, \dots, x_n),$$

where  $x_1, \dots, x_n$  are linearly independent  $n$ -vectors. Then

$$A = SJ_n(\lambda)S^{-1} \tag{1.1.5}$$

iff the vectors  $x_i$  satisfy

$$(A - \lambda I)x_n = 0$$

and

$$(A - \lambda I)x_i = x_{i+1}, \quad i = 1, \dots, n-1. \tag{1.1.6}$$

**Proof.** This can be seen directly by writing Equation (1.1.5) as  $AS = SJ_n(\lambda)$  and expanding  $J_n(\lambda)$  and  $S$  from their definition. For example, with  $n = 3$  we have

$$A(x_1 \ x_2 \ x_3) = (x_1 \ x_2 \ x_3) \begin{pmatrix} \lambda & 0 & 0 \\ 1 & \lambda & 0 \\ 0 & 1 & \lambda \end{pmatrix},$$

which is as required.  $\square$

**Lemma 1.1.4.** If  $A$  satisfies Equation (1.1.5), then  $A$  has exactly one linearly independent eigenvector.

**Proof.** The vector  $x_n$  is an eigenvector of  $A$  because it satisfies  $Ax_n = \lambda x_n$ . Suppose  $y$  is another eigenvector of  $A$ , with eigenvalue  $\gamma$ . Then  $Ay = \gamma y$  implies that

$$SJ_n(\lambda)S^{-1}y = \gamma y.$$

But this implies that  $J_n(\lambda)z = \gamma z$ , where  $z = S^{-1}y$ , so that  $\gamma = \lambda$  and  $z = \alpha e_n$ , for some nonzero scalar  $\alpha$ , so that  $y = Sz = \alpha S e_n = \alpha x_n$ . Hence  $x_n$  and  $y$  are linearly dependent.  $\square$

**Lemma 1.1.5.** Suppose there is a nonsingular matrix  $S$  such that

$$A = S \begin{pmatrix} B & 0 \\ 0 & J_k(\lambda) \end{pmatrix} S^{-1},$$

where  $1 \leq k \leq n - 1$ , with  $B$  an  $(n - k) \times (n - k)$  matrix. Then  $A$  has at least two linearly independent eigenvectors.

**Proof.** The matrix  $B$  has at least one eigenvector. Let  $z$  be an arbitrary eigenvector of  $B$ . Then the vectors

$$x = S \begin{pmatrix} 0 \\ e_k \end{pmatrix} \text{ and } y = S \begin{pmatrix} z \\ 0 \end{pmatrix}$$

are linearly independent, and both are eigenvectors of  $A$ .  $\square$

The next two results are given without proof. See Gantmacher (1959) for a complete derivation of the Jordan canonical form.

**Lemma 1.1.6.** A complex  $n \times n$  matrix  $A$  is similar to  $J_n(\lambda)$  iff it has exactly one linearly independent eigenvector.

**Theorem 1.1.7. (Jordan canonical form).** An  $n \times n$  complex matrix  $A$  has exactly  $k$  linearly independent eigenvectors ( $1 \leq k \leq n$ ) iff it is similar to a block-diagonal matrix  $B = \text{diag}(B_i)$  such that  $B_i = J_{n_i}(\lambda_i)$ , for  $i = 1, \dots, k$ , and  $\sum_{i=1}^k n_i = n$ .

For example,

$$A = \begin{pmatrix} 1.5 & 0.5 & -1 \\ -0.5 & 1.5 & 0 \\ -0.5 & -0.5 & 2 \end{pmatrix}$$

has two linearly independent eigenvectors

$$y_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \text{ and } y_2 = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$$

with eigenvalues  $\lambda_1 = 1$  and  $\lambda_2 = 2$ , respectively. Solving Equation (1.1.6) with  $x_3 = y_2$  and  $\lambda_2 = 2$ , we get

$$x_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix},$$

so that  $\lambda_2$  has multiplicity 2. Letting  $x_1 = y_1$  and  $S = (x_1, x_2, x_3)$ , we get the Jordan canonical form of  $A$  as

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0 \end{pmatrix}.$$

Hence we have  $B_1 = J_1(1)$  and  $B_2 = J_2(2)$ .

Some observations are in order here. First, Lemma 1.1.1 is a special case of the above theorem with  $k = n$ , while Lemma 1.1.6 is the special case with  $k = 1$ . Second, our formulation of Theorem 1.1.7 emphasizes the fact that each Jordan submatrix is associated with exactly one eigenvector of the matrix  $A$ . Third, even when all the entries of a matrix  $A$  are real, it is possible that some of its eigenvalues be complex, in which case the transformation matrix  $S$  of Theorem 1.1.7 has some complex entries too. For example, the matrix

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

has two imaginary eigenvalues  $\lambda_1 = \iota$  and  $\lambda_2 = -\iota$ , where  $\iota^2 = -1$ .

It should be clear, from our derivation of the Jordan canonical form, that if  $A$  is real and if all the eigenvalues are real, then  $S$  is real as well, because then the columns of  $S$  are solutions of linear systems of equations with real coefficients. Also, since all the coefficients of the characteristic polynomial  $\det(\lambda I - A)$  are real, the eigenvalues of the real matrix  $A$  which are not real are arranged in complex conjugate pairs. Hence if  $\lambda^+ = \alpha + \iota\beta$  is an eigenvalue of  $A$ , so is  $\lambda^- = \alpha - \iota\beta$ . We use this property to derive a *modified Jordan canonical form* which has all entries real.

First, observe that if  $x^+ = u + \iota v$  is an eigenvector of  $A$  with eigenvalue  $\lambda^+ = \alpha + \iota\beta$ , then  $x^- = u - \iota v$  is also an eigenvector of  $A$ , but with eigenvalue  $\lambda^- = \alpha - \iota\beta$ . Then suppose that  $A$  has  $k_1$  real linearly independent eigenvectors  $x_i$  with eigenvalues  $\lambda_i$ ,

$i = 1, \dots, k_1$ , and  $k_2$  pairs of complex conjugate linearly independent eigenvectors  $(x_i^+, x_i^-)$ ,  $i = k_1 + 1, \dots, k_1 + k_2$ . By Theorem 1.1.7, for every complex conjugate pair  $(x_i^+, x_i^-)$  there is a corresponding pair of Jordan blocks  $(J_{n_i}(\lambda_i^+), J_{n_i}(\lambda_i^-))$ . We describe how such a pair of complex conjugate Jordan blocks can be replaced by a completely real submatrix.

Consider a complex eigenvalue  $\lambda = \alpha + i\beta$ . We define the  $2 \times 2$  matrices  $K$  and  $\Lambda$  such that

$$K = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \quad (1.1.8)$$

and

$$\Lambda = \begin{pmatrix} \alpha + i\beta & 0 \\ 0 & \alpha - i\beta \end{pmatrix}. \quad (1.1.9)$$

**Lemma 1.1.8.**  $K$  is similar to  $\Lambda$ .

**Proof.** Because  $K = X\Lambda X^{-1}$ , where

$$X = \begin{pmatrix} i\beta & -i\beta \\ \beta & \beta \end{pmatrix} \quad \text{and} \quad X^{-1} = \begin{pmatrix} -i/2\beta & 1/2\beta \\ i/2\beta & 1/2\beta \end{pmatrix}. \quad (1.1.10)$$

□

Now we define the *block Jordan submatrix*  $J_k(A)$ , where  $A$  is a square matrix, so that its subdiagonal blocks are equal to  $I$ , the identity. Then, for example,  $J_1(A) = A$  and

$$J_4(A) = \begin{pmatrix} A & 0 & 0 & 0 \\ I & A & 0 & 0 \\ 0 & I & A & 0 \\ 0 & 0 & I & A \end{pmatrix}.$$

**Lemma 1.1.9.** Let  $K$  be defined as in Equation (1.1.8). Then  $J_k(K)$  is similar to  $L_{2k}(\lambda)$ , where

$$L_{2k}(\lambda) = \begin{pmatrix} J_k(\alpha + i\beta) & 0 \\ 0 & J_k(\alpha - i\beta) \end{pmatrix}.$$

**Proof.** First observe that  $J_k(K)$  is similar to  $J_k(\Lambda)$ , by Lemma 1.1.8. To see this, take  $S = \text{diag}(X_i)$ , with  $X_i = X$  as in Equation (1.1.10). Then  $J_k(K) = SJ_k(\Lambda)S^{-1}$ . Next we show that  $J_k(\Lambda)$  is similar to  $L_{2k}(\lambda)$ . In fact,  $J_k(\Lambda) = TL_{2k}(\lambda)T^{-1}$ , where

$T$  is a permutation matrix. To see this, let us associate to each matrix a directed graph as follows. There are  $2k$  nodes, one per row, labelled 1 to  $2k$ , and for every nonzero entry, there is a corresponding arc labelled by the value of that entry. The graphs of  $J_k(\Lambda)$  and  $L_{2k}(\lambda)$  are shown in Figure 1.1.1, in which  $\lambda^*$  denotes the complex conjugate of  $\lambda$ . One can convince oneself that the graphs are correct by checking them with  $k = 2$ .

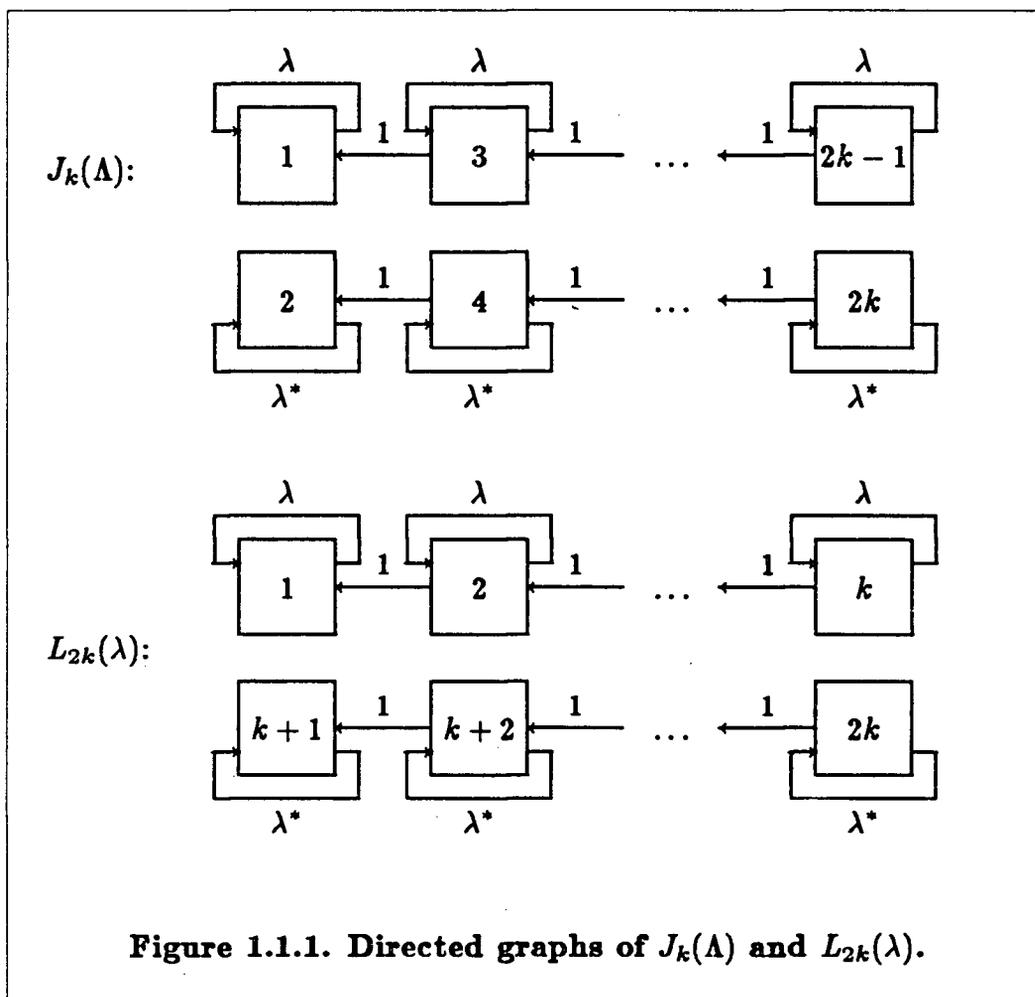


Figure 1.1.1. Directed graphs of  $J_k(\Lambda)$  and  $L_{2k}(\lambda)$ .

We have

$$J_2(\Lambda) = \begin{pmatrix} \Lambda & 0 \\ I & \Lambda \end{pmatrix} = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda^* & 0 & 0 \\ 1 & 0 & \lambda & 0 \\ 0 & 1 & 0 & \lambda^* \end{pmatrix}$$

and

$$L_4(\lambda) = \begin{pmatrix} J_2(\lambda) & 0 \\ 0 & J_2(\lambda^*) \end{pmatrix} = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 1 & \lambda & 0 & 0 \\ 0 & 0 & \lambda^* & 0 \\ 0 & 0 & 1 & \lambda^* \end{pmatrix}.$$

Now both graphs in Figure 1.1.1 are identical, except for the node labels. Hence  $J_k(\Lambda)$  is obtained by permuting the rows and columns of  $L_{2k}(\lambda)$  accordingly, which completes the proof.  $\square$

**Theorem 1.1.10.** (*Modified Jordan canonical form*). Suppose  $A$  is a real  $n \times n$  matrix. Then  $A$  has exactly  $k_1$  linearly independent real eigenvectors and  $k_2$  linearly independent complex conjugate pairs of eigenvectors iff it is similar to a block diagonal matrix  $B = \text{diag}(B_i)$  such that  $B_i = J_{n_i}(\lambda_i)$ , for  $i = 1, \dots, k_1$ , and  $B_i = J_{n_i}(K_i)$ , for  $i = k_1 + 1, \dots, k_1 + k_2$ , with

$$\sum_{i=1}^{k_1} n_i + 2 \sum_{i=k_1+1}^{k_1+k_2} n_i = n.$$

**Proof.** First apply Theorem 1.1.7 to  $A$ . We have the real Jordan blocks  $B_i = J_{n_i}(\lambda_i)$ , for  $i = 1, \dots, k_1$ , directly. Then organize the complex conjugate pairs as  $L_{2n_i}(\lambda_i)$ , for  $i = k_1 + 1, \dots, k_1 + k_2$ . The result follows because by Lemma 1.1.9,  $B_i = J_{n_i}(K_i)$  is similar to  $L_{2n_i}(\lambda_i)$ .  $\square$

For example, the matrix

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & -1 & 2 & -1 \\ 2 & -1 & 2 & 0 \end{pmatrix}$$

has only one pair of complex conjugate eigenvectors, with eigenvalues  $1 + \iota$  and  $1 - \iota$ , each with multiplicity 2. Then  $A$  is similar to

$$L_4(1 + \iota) = \begin{pmatrix} J_2(1 + \iota) & 0 \\ 0 & J_2(1 - \iota) \end{pmatrix} = \begin{pmatrix} 1 + \iota & 0 & 0 & 0 \\ 1 & 1 + \iota & 0 & 0 \\ 0 & 0 & 1 - \iota & 0 \\ 0 & 0 & 1 & 1 - \iota \end{pmatrix}.$$

We have the modified Jordan canonical form  $A = SJ_2(K)S^{-1}$ , where

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad S^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

and

$$J_2(K) = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{pmatrix} = XJ_2(\Delta)X^{-1},$$

with

$$X = \begin{pmatrix} \iota & -\iota & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & \iota & -\iota \\ 0 & 0 & 1 & 1 \end{pmatrix} \text{ and } X^{-1} = \begin{pmatrix} -\iota/2 & 1/2 & 0 & 0 \\ \iota/2 & 1/2 & 0 & 0 \\ 0 & 0 & -\iota/2 & 1/2 \\ 0 & 0 & \iota/2 & 1/2 \end{pmatrix}.$$

Hence the original Jordan canonical form of  $A$  is

$$A = SXTL_4(1 + \iota)T^{-1}X^{-1}S^{-1},$$

where  $T$  is the permutation matrix that interchanges rows 2 and 3.

The modified Jordan canonical form effectively contains all the information of the original canonical form, but it is defined with real arithmetic only. For, suppose  $A = SBS^{-1}$  where  $B$  is as in Theorem 1.1.10, then  $S$  is a solution of the equation  $AS = SB$ , which has real coefficients only. Hence Theorem 1.1.10 will be useful in establishing the existence of a matrix decomposition suitable for the definition and representation of the Drazin inverse of a real matrix. Moreover, Theorem 1.1.10 suggests an easy way to generate test matrices for evaluating the performance of computer algorithms to compute the Drazin solution of singular equations. [This method has its limitations, however. See Golub and Wilkinson (1976), Page 583].

## 1.2. The Drazin generalized inverse

Let  $A$  be a real  $n \times n$  matrix such that  $A^k = 0$ , for some positive integer  $k$ . Then  $A$  is said to be *nilpotent*. The smallest such integer  $k \geq 1$  is called the *index of nilpotency* of  $A$  [see e.g., Gantmacher (1959)]. For example, the index of nilpotency of the Jordan submatrix  $J_n(0)$  is  $n$ , because  $J_n(0)^n = 0$  while  $J_n(0)^{n-1} \neq 0$ . However,  $J_n(\lambda)$  is not nilpotent if  $\lambda \neq 0$ .

**Lemma 1.2.1.** Suppose  $A$  is a real  $n \times n$  matrix. Then  $A$  is nilpotent iff zero is its only eigenvalue. Moreover, the index of nilpotency of a nilpotent matrix  $A$  is the dimension

of the largest block of its Jordan canonical form.

**Proof.** First, suppose  $A$  is similar to some matrix  $B$ . Observe that  $A$  is nilpotent iff  $B$  is nilpotent. Moreover, if  $A$  is nilpotent then it has the same index of nilpotency as  $B$ . This is because  $A^k = SB^kS^{-1}$ , for any  $k \geq 0$ , so that  $A^k = 0$  iff  $B^k = 0$ .

Next, suppose  $A$  has  $m$  linearly independent eigenvectors, and consider the block-diagonal matrix  $B$  of the Jordan canonical form of  $A$  of Theorem 1.1.7. If  $m = 1$ , then  $A$  is similar to  $J_n(\lambda)$  and hence  $A$  is nilpotent iff  $\lambda = 0$ . Moreover, if  $A$  is nilpotent, its index of nilpotency is  $n$ .

If  $m > 1$ , the Jordan blocks are  $J_{n_1}(\lambda_1), \dots, J_{n_m}(\lambda_m)$ . Again,  $A$  is nilpotent iff  $\lambda_1 = \dots = \lambda_m = 0$ , and if  $A$  is nilpotent, its index of nilpotency is  $\max\{n_i : i = 1, \dots, m\}$ .  $\square$

Now consider a general real  $n \times n$  matrix  $A$ . The *index* of  $A$ , denoted  $\text{ind}(A)$ , is the smallest nonnegative integer  $k$  such that  $\text{rank}(A^{k+1}) = \text{rank}(A^k)$  [see e.g., Campbell and Meyer (1979) and Berman and Plemmons (1979)]. This definition is an extension of the above index of nilpotency. For, if  $A$  is nilpotent, then  $\text{ind}(A)$  is the index of nilpotency of  $A$ . If  $A$  is nonsingular, then  $\text{ind}(A) = 0$ . If  $A$  is singular but not nilpotent, we have the following decomposition.

**Theorem 1.2.2.** Suppose  $A$  is a real  $n \times n$  matrix which is singular but not nilpotent. Then there is a nonsingular matrix  $S$  such that

$$A = S^{-1} \begin{pmatrix} B & 0 \\ 0 & N \end{pmatrix} S, \quad (1.2.1)$$

where  $B$  is a non-singular, real  $(n - m) \times (n - m)$  matrix and  $N$  is a nilpotent  $m \times m$  matrix, with  $1 \leq m \leq n - 1$ . Moreover, the index of  $A$  equals the index of nilpotency of  $N$ .

**Proof.** One such decomposition is the modified Jordan canonical form of Theorem 1.1.10. The matrix  $B$  contains all the modified blocks with nonzero eigenvalues, and  $N$  has all the Jordan blocks with eigenvalue zero. The result about the index of  $A$  follows because  $\text{rank}(A^{k+1}) = \text{rank}(A^k)$  iff  $N^k = 0$ .  $\square$

Hence the index of a singular matrix  $A$  is the index of nilpotency of its nilpotent component  $N$ , which is equal to the dimension of the largest Jordan block with eigenvalue zero. Note that if  $\text{ind}(A) = 1$ , then  $N = 0$ , the zero matrix. This special case will be of particular interest in the analysis of Markov processes.

Following Drazin (1958), we say that a real  $n \times n$  matrix  $A^D$  is a *Drazin generalized inverse*, or simply a *Drazin inverse*, of the  $n \times n$  matrix  $A$  if it satisfies the three following conditions:

$$A^D A = A A^D \tag{1.2.2}$$

$$A^m = A^{m+1} A^D \quad \text{for some } m \geq 0 \tag{1.2.3}$$

$$A^D = (A^D)^2 A. \tag{1.2.4}$$

Drazin's definition, however, is valid in a much more general context (i.e.,  $A$  may be an element of an arbitrary associative ring or semigroup, not necessarily a finite square matrix). While in a general algebraic structure there may be some element  $A$  for which no such  $A^D$  exist, it is the case that every real square matrix has a Drazin inverse. We prove this as in Wilkinson (1982), after some preliminary lemmas.

**Lemma 1.2.3.** Suppose  $A = S^{-1}BS$  and that  $B^D$  exists. Then

$$A^D = S^{-1}B^D S. \tag{1.2.5}$$

**Proof.** It is easy to verify that if  $B$  and  $B^D$  satisfy (1.2.2), (1.2.3) and (1.2.4), then so do  $A$  and  $A^D$  given by (1.2.5). We can see (1.2.2) as follows.

$$A^D A = S^{-1}B^D S S^{-1}BS = S^{-1}B^D BS$$

and

$$A A^D = S^{-1}B S S^{-1}B^D S = S^{-1}B B^D S,$$

so that

$$B^D B = B B^D \Rightarrow S^{-1}B^D BS = S^{-1}B B^D S,$$

and the result follows. The other conditions are left as an exercise to the reader.  $\square$

**Lemma 1.2.4.** If  $A$  is nonsingular, then  $A^D = A^{-1}$  is the unique matrix that satisfies (1.2.2), (1.2.3) and (1.2.4).

**Proof.** Because  $X = A^{-1}$  is the unique solution of  $AX = I$ , which is Equation (1.2.3) with  $m = 0$ . The other conditions are trivial.  $\square$

**Lemma 1.2.5.** If  $A$  is nilpotent, then  $A^D = 0$  is the unique solution of (1.2.2), (1.2.3) and (1.2.4).

**Proof.** It will be convenient, for later results, to use the notation  $N = A$  and  $T = A^D$ . We show that  $X = 0$  is the unique solution of Equation (1.2.4). The other conditions are trivial. We have

$$\begin{aligned} T &= T^2 N = T(T)N = T(T^2 N)N \\ &= T^2(T)N^2 = \dots \\ &= T^k(T)N^k = 0, \end{aligned}$$

where  $k = \text{ind}(N)$ .  $\square$

**Lemma 1.2.6.** Suppose

$$A = \begin{pmatrix} B & 0 \\ 0 & N \end{pmatrix},$$

where  $B$  is nonsingular and  $N$  is nilpotent. Then

$$A^D = \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

is the unique solution of (1.2.2), (1.2.3) and (1.2.4).

**Proof.** Suppose

$$A^D = \begin{pmatrix} X & Y \\ Z & T \end{pmatrix},$$

and let  $k = \text{ind}(A)$ . Then (1.2.3) with  $m = k$  gives

$$\begin{pmatrix} B^k & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} B^{k+1}X & 0 \\ 0 & 0 \end{pmatrix},$$

so that  $X = B^{-1}$  is uniquely determined. Next (1.2.2) gives

$$\begin{pmatrix} I & YN \\ ZB & TN \end{pmatrix} = \begin{pmatrix} I & BY \\ NZ & NT \end{pmatrix},$$

so that

$$YN = BY \tag{1.2.6}$$

and

$$ZB = NZ. \tag{1.2.7}$$

We use (1.2.6) to show that  $Y = 0$ . We have

$$B(YN^{k-1}) = (BY)N^{k-1} = (YN)N^{k-1} = Y(N^k) = Y(0) = 0,$$

so that  $YN^{k-1} = 0$  because  $B$  is nonsingular. Continuing this way, we get  $YN^{k-2} = 0, \dots, YN = 0$  and finally,  $Y = 0$ . Similarly, we can use (1.2.7) to show that  $Z = 0$ . Hence  $Y$  and  $Z$  are uniquely determined by Equation (1.2.2). Finally, Equation (1.2.4) gives

$$\begin{pmatrix} B^{-1} & 0 \\ 0 & T \end{pmatrix} = \begin{pmatrix} B^{-1} & 0 \\ 0 & T^2 N \end{pmatrix},$$

from which  $T = 0$  is uniquely determined as in Lemma 1.2.5. □

**Theorem 1.2.7.** Suppose  $A$  is a real  $n \times n$  matrix. Then its Drazin inverse  $A^D$  is unique and is given by

$$A^D = \begin{cases} A^{-1} & \text{if } A \text{ is non-singular,} \\ 0 & \text{if } A \text{ is nilpotent,} \\ S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} S & \text{otherwise,} \end{cases}$$

where  $B$  is the nonsingular matrix of Equation (1.2.1).

**Proof.** Immediate from Lemmas 1.2.3, 1.2.4, 1.2.5 and 1.2.6. Uniqueness for the non-singular and nilpotent cases has been shown already. There is an apparent difficulty in the other case, because the decomposition (1.2.1) is not unique. Observe, however, that for a given similarity transformation  $S$ , there is a one-to-one correspondence between  $A$  and  $C = \begin{pmatrix} B & 0 \\ 0 & N \end{pmatrix}$ . Hence the uniqueness of  $C^D$  implies that of  $A^D$ , by Lemma 1.2.3. □

As will be seen in Chapter III, the Drazin inverse  $A^D$  will be useful for expressing the solution of the resolvent equations which are encountered in dynamic programming. Let the matrix  $W$  be a projection onto the null space of  $A^D$ .

**Lemma 1.2.8.** If  $A$  is decomposed as in Equation (1.2.1), then

$$W = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S. \quad (1.2.8)$$

**Proof.** Denote the null space of  $A^D$  by  $\text{null}(A^D)$ . By definition,  $x \in \text{null}(A^D)$  iff  $A^D x = 0$ . Also,  $W$  is a projection onto  $\text{null}(A^D)$  if  $Wx = x$  for  $x \in \text{null}(A^D)$ ,  $A^D W y = 0$  and  $W A^D y = 0$  for an arbitrary vector  $y$ .

Now write

$$x = S^{-1} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \text{ and } W = S^{-1} \begin{pmatrix} X & Y \\ Z & T \end{pmatrix} S.$$

Then  $A^D x = 0$  iff

$$S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

that is, iff  $u_1 = 0$ . Hence  $Wx = x$  iff

$$\begin{pmatrix} X & Y \\ Z & T \end{pmatrix} \begin{pmatrix} 0 \\ u_2 \end{pmatrix} = \begin{pmatrix} 0 \\ u_2 \end{pmatrix},$$

so that  $Y = 0$  and  $T = I$ , the identity matrix.

The condition  $A^D W y = 0$  for arbitrary  $y$  is equivalent to  $A^D W = 0$ , that is

$$\begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X & 0 \\ Z & I \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

which implies  $X = 0$ .

Finally, the condition  $W A^D y = 0$  is equivalent to  $W A^D = 0$ , which implies

$$\begin{pmatrix} 0 & 0 \\ Z & I \end{pmatrix} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

so that  $Z = 0$ . Hence the result. □

The following useful representation for  $W$  is immediate.

**Corollary 1.2.9.**  $W = I - AA^D$ .

**Proof.**

$$I - AA^D = S^{-1} \begin{pmatrix} I - BB^{-1} & 0 \\ 0 & I - N \times 0 \end{pmatrix} S = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S = W. \quad \square$$

### 1.3. Matrix decomposition algorithms

Throughout the thesis, we will be concerned mainly with matrices  $A$  such that  $\text{ind}(A) = 1$ . Then by Theorem 1.2.2, we have

$$A = S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} S. \quad (1.3.1)$$

In this special case, the Drazin inverse of  $A$  is called its *group inverse* [see Campbell and Meyer (1979) and Berman and Plemmons (1979)], and is denoted by  $A^\#$ . By Theorem 1.2.7, we have

$$A^\# = S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} S. \quad (1.3.2)$$

We will consider two algorithms for producing the group inverse of a matrix. Both methods consist essentially of two major steps: first produce a similarity transformation to decompose  $A$  as in (1.3.1) and second, invert the matrix  $B$  as in (1.3.2). They are specializations of the more general algorithm of Wilkinson (1982) to compute the Drazin inverse  $A^D$  of a general matrix. Suppose  $\text{ind}(A) = k \geq 1$ . Then Wilkinson's method uses  $k$  steps to decompose  $A$  as in Equation (1.2.1) and one further step to invert  $B$ .

Although we will encounter some matrices of index  $k > 1$  in dynamic programming, we will not need Wilkinson's algorithm in all its generality, because the special structure of the problem will make it possible to break it down into a sequence of subproblems of index 1. Hence there is no need, within the scope of this thesis, for a general algorithm.

Before actually discussing the algorithms, it is important to realize that, as pointed out by Wilkinson (1982), there is no need for producing the matrix  $A^\#$  (nor  $B^{-1}$  and  $W = I - AA^\#$ ) explicitly, as we are interested in producing the vectors  $h = A^\#b$  and  $g = Wb$ , for a given right hand side vector  $b$ . For this purpose, the algorithms will produce a representation of  $B^{-1}$ ,  $A^\#$  and  $W$  in factorized form. Then a solution can be obtained by applying that sequence of factors to the right hand side  $b$  directly.

To illustrate the idea, let us consider a nonsingular matrix  $A$ . By applying Gaussian elimination with row interchanges [see e.g., Wilkinson (1965)], we obtain an upper

triangular matrix  $U$  such that

$$A = PLU, \quad (1.3.3)$$

where  $P$  is a permutation matrix and  $L$  is a *unit* lower triangular matrix (*unit* means that its diagonal entries are equal to 1). Typically, only  $U$  is obtained explicitly.  $L$  is represented by a product of elementary row operations and it is needed only to store the  $n(n-1)$  multipliers of the Gaussian elimination.  $P$  is represented by  $(n-1)$  integers giving the rows which were interchanged during the elimination process.

Now, to solve the system of equations  $Ax = b$  and obtain  $x = A^{-1}b$ , we use the factorization (1.3.3) to rewrite our system as  $(PLU)x = b$ . We can then solve the latter system in three steps, as in Figure 1.3.1.

- Step 1. Solve  $Pz = b$ , giving  $z = P^{-1}b$   
(permute the entries of  $b$ ).
- Step 2. Solve  $Ly = z$ , giving  $y = L^{-1}z$   
(apply the multipliers of the Gaussian elimination to  $y$ ).
- Step 3. Solve  $Ux = y$ , giving  $x = U^{-1}y$   
(use back substitution, producing  $x_n, x_{n-1}, \dots, x_1$ ).

**Figure 1.3.1.** Compute  $x = A^{-1}b$ , where  $A = PLU$ .

Hence we do get

$$x = U^{-1}L^{-1}P^{-1}b = A^{-1}b$$

without producing  $A^{-1}$ ,  $U^{-1}$ ,  $L^{-1}$  or  $P^{-1}$ . In fact,  $P$  and  $L$  were not produced explicitly either. A simple example will clarify this. Let us solve  $Ax = b$ , where

$$A = \begin{pmatrix} 1 & 0.1 & 1 \\ 5 & -2 & 2 \\ 2 & -1.8 & 1.8 \end{pmatrix} \text{ and } b = \begin{pmatrix} 4.2 \\ 7 \\ 3.8 \end{pmatrix}.$$

Using Gaussian elimination with partial pivoting [the pivot elements are chosen as big as possible, see e.g., Wilkinson (1965)], a matrix ALU and a vector NPIV are produced, such that

$$\text{ALU} = \begin{pmatrix} 5 & -2 & 2 \\ \underline{-0.4} & -1 & 1 \\ \underline{-0.2} & \underline{0.5} & 1.1 \end{pmatrix} \text{ and NPIV} = \begin{pmatrix} 2 \\ 3 \\ - \end{pmatrix}.$$

The upper part of ALU is the upper triangular matrix  $U$  itself, and the underscored entries are the multipliers that were used to produce the zeroes where they are stored. The vector NPIV tells that the rows of  $A$  were interchanged as follows: first row 1 and row 2, then row 2 and row 3 [see e.g., Johnston (1982), Chapter 2]. The solution  $x$  can then be computed as follows, as in Figure 1.3.1.

$$\text{Step 1. } z = P^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4.2 \\ 7 \\ 3.8 \end{pmatrix} = \begin{pmatrix} 7 \\ 3.8 \\ 4.2 \end{pmatrix}$$

$$\text{Step 2. } y = L^{-1}z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -0.4 & 1 & 0 \\ -0.2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 7 \\ 3.8 \\ 4.2 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \\ 3.3 \end{pmatrix}$$

$$\text{Step 3. } x = U^{-1}y : \begin{pmatrix} 5 & -2 & 2 \\ 0 & -1 & 1 \\ 0 & 0 & 1.1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \\ 3.3 \end{pmatrix}, \text{ giving } x = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Since our algorithms will use the PLU decomposition, it is worthwhile to look at the cost of producing the PLU factorization of a matrix and computing the solution for a given right hand side. We consider three types of operations. A *flop*, short for *floating point operation*, is a pair of arithmetic operations that consists of a multiplication/division and an addition/subtraction (for example  $\sum_{i=1}^n a_i/b_i$  can be computed with  $n$  flops). A *swap* is the interchange of two entries of a vector or a matrix (hence interchanging two rows of an  $n \times n$  matrix requires  $n$  swaps). Finally, a *comparison* is a relational operation (i.e.  $\leq, \geq$ ). The operations counts for Gaussian elimination, as well as for solving a system of linear equations, are given in Figure 1.3.2.

(The comparison operations are done when searching for the largest pivot). For  $n$  reasonably large, the terms in  $n^2$  are negligible in the Gaussian elimination procedure. We will use those counts as a reference, to compare with the Drazin inverse algorithms.

Gaussian elimination of a nonsingular matrix  $A = PLU$ :

$\frac{1}{3}n^3$  flops +  $n^2$  swaps +  $\frac{1}{2}n^2$  comparisons.

Permutations  $Px$  or  $P^{-1}x$ :  $n$  swaps.

Triangular products  $Lx$ ,  $L^{-1}x$ ,  $Ux$ ,  $U^{-1}x$ :  $\frac{1}{2}n^2$  flops.

Solution of RHS  $x = U^{-1}L^{-1}P^{-1}b$ :  $n^2$  flops +  $n$  swaps.

**Figure 1.3.2. Operations counts for Gaussian elimination.**

Before developing the algorithms, however, it will be convenient to derive the following results.

**Lemma 1.3.1.** Suppose  $A$  is an  $n \times n$  real matrix such that  $\text{rank}(A) = n - m$ , with  $1 \leq m < n$ , and

$$A = S \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} S^{-1} \quad (1.3.4)$$

for some nonsingular  $S$ , where  $B_{11}$  is an  $(n - m) \times (n - m)$  matrix. Then  $B_{11}$  is nonsingular iff  $\text{ind}(A) = 1$ .

**Proof.** Assume first that  $B_{11}$  is nonsingular. Then

$$A = (SV) \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix} (SV)^{-1},$$

where

$$V = \begin{pmatrix} I & -B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix} \quad \text{and} \quad V^{-1} = \begin{pmatrix} I & B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix}.$$

This implies that  $\text{ind}(A) = 1$ , because  $\text{rank}(A^2) = \text{rank}(A)$ , and the “only if” part is proved. For the “if” part, assume that  $\text{ind}(A) = 1$ . By Equation (1.3.1), there is a nonsingular matrix  $U$  and a nonsingular  $(n - m) \times (n - m)$  matrix  $C_{11}$  such that

$$A = U \begin{pmatrix} C_{11} & 0 \\ 0 & 0 \end{pmatrix} U^{-1}.$$

By Equation (1.3.4), we have then

$$S \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} S^{-1} = U \begin{pmatrix} C_{11} & 0 \\ 0 & 0 \end{pmatrix} U^{-1}. \quad (1.3.5)$$

Letting  $V = S^{-1}U$  and writing

$$V = \begin{pmatrix} X & Y \\ Z & T \end{pmatrix},$$

Equation (1.3.5) becomes

$$\begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X & Y \\ Z & T \end{pmatrix} = \begin{pmatrix} X & Y \\ Z & T \end{pmatrix} \begin{pmatrix} C_{11} & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence we get

$$B_{11}X + B_{12}Z = XC_{11} \tag{1.3.6}$$

$$B_{11}Y + B_{12}T = 0$$

$$0 = ZC_{11}. \tag{1.3.7}$$

Now since  $C_{11}$  is nonsingular, Equation (1.3.7) implies that  $Z = 0$ . Hence for  $V$  to be nonsingular, both  $X$  and  $T$  must be nonsingular as well. But then Equation (1.3.6) implies that  $B_{11} = XC_{11}X^{-1}$ , and thus  $B_{11}$  is nonsingular.  $\square$

**Theorem 1.3.2.** Suppose  $A$  is an  $n \times n$  matrix satisfying Equation (1.3.4) with  $B_{11}$  nonsingular. Then

$$A^\# = S \begin{pmatrix} B_{11}^{-1} & B_{11}^{-1}B_{11}^{-1}B_{12} \\ 0 & 0 \end{pmatrix} S^{-1} \tag{1.3.8}$$

and

$$W = I - AA^\# = S \begin{pmatrix} 0 & -B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix} S^{-1}. \tag{1.3.9}$$

**Proof.** As in the first part of the proof of Lemma 1.3.1, let

$$V = \begin{pmatrix} I & -B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix}.$$

Then

$$A = (SV) \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix} (SV)^{-1},$$

so that by Theorem 1.2.7 and Equation 1.2.8,

$$A^\# = SV \begin{pmatrix} B_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} V^{-1}S^{-1} = S \begin{pmatrix} B_{11}^{-1} & B_{11}^{-1}B_{11}^{-1}B_{12} \\ 0 & 0 \end{pmatrix} S^{-1}$$

and

$$W = SV \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} V^{-1} S^{-1} = S \begin{pmatrix} 0 & -B_{11}^{-1} B_{12} \\ 0 & I \end{pmatrix} S^{-1}.$$

□

Both algorithms will use Gaussian elimination to decompose a singular matrix  $A$  as in Equation (1.3.4). But rather than transforming the matrix as in Equation (1.3.1), the Drazin inverse will be represented directly using Theorem 1.3.2. Moreover, the case  $\text{ind}(A) > 1$  will be flagged automatically when  $B_{11}$  is found to be singular, by Lemma 1.3.1. Hence our algorithms will produce a representation of the group inverse  $A^\#$  of a matrix  $A$  if  $\text{ind}(A) = 1$ , and return an error condition when  $\text{ind}(A) > 1$ . A  $2 \times 2$  example of the latter situation is when

$$A = S \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} S^{-1}.$$

Then  $\text{ind}(A) = 2$  and  $B_{11} = 0$ .

Let us call *slow algorithm* the first, and most obvious, of our algorithms. It is shown in Figure 1.3.3. Steps 1.1 and 1.2 essentially obtain a representation of Equation (1.3.1), and steps 2.1 and 2.2 obtain a representation of (1.3.2).

The reason for calling it the slow algorithm will become clear after we analyze the amount of arithmetic it requires. Before that, let us see what the algorithm does. In Step 1.1, Gaussian elimination is used to compute a PLU factorization of  $A$  such that the upper triangular matrix  $U$  has  $m$  zero rows, with  $\text{rank}(A) = n - m$ , so that

$$U = \begin{pmatrix} U_{11} & U_{12} \\ 0 & 0 \end{pmatrix}.$$

Then in Step 1.2, the matrix  $B = UPL$  is such that

$$B = \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix}.$$

Moreover,  $B$  is similar to  $A$ , because

$$A = PLU = (PL)U(PL)^{-1} = (PL)B(PL)^{-1}.$$

- Step 1.1. Compute  $P$ ,  $L$  and  $U$  such that  $A = PLU$   
(using Gaussian elimination).
- Step 1.2. Compute  $B = UPL$   
(applying  $P$  and  $L$  to the nonzero rows of  $U$ ).
- Step 2.1. Compute  $R$ ,  $S$  and  $T$  such that  $B_{11} = RST$   
(using Gaussian elimination).  
(Note:  $R$  is permutation,  $S$  lower,  $T$  upper triangular).
- Step 2.2. If  $B_{11}$  is singular then "Error:  $\text{ind}(A) > 1$ "  
else stop (we have all the necessary information).

**Figure 1.3.3. The slow decomposition algorithm.**

Now let  $X = UP$ , and write

$$X = \begin{pmatrix} X_{11} & X_{12} \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}. \quad (1.3.10)$$

That  $B$  has the required form can be seen directly:

$$\begin{aligned} B = UPL = XL &= \begin{pmatrix} X_{11} & X_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \\ &= \begin{pmatrix} X_{11}L_{11} + X_{12}L_{21} & X_{12}L_{22} \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

The above argument is well known, and is used by Wilkinson (1982) in particular. It is important to note that while  $U_{11}$  is likely to be singular in general, the matrix  $B_{11}$  is always nonsingular, provided that  $\text{ind}(A) = 1$ , by Lemma 1.3.1. Hence Step 2.1 simply computes the  $PLU$  decomposition of the nonsingular,  $(n - m) \times (n - m)$  matrix  $B_{11}$ , giving a permutation matrix  $R$ , a unit lower triangular matrix  $S$  and a nonsingular upper triangular matrix  $T$ .

For example, let us apply the algorithm to a  $3 \times 3$  matrix  $A$  such that  $\text{ind}(A) = 1$ :

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix}.$$

Using Gaussian elimination, we get  $A = PLU$ , where  $P = I$ , the identity matrix,

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Observe that

$$U_{11} = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$$

is singular. Then we produce  $B = UPL = UL$ :

$$B = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{so that} \quad B_{11} = \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix}.$$

Finally, using Gaussian elimination we get  $B = RST$ , where  $R = I$ ,

$$S = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix}.$$

Hence

$$B_{11}^{-1} = T^{-1}S^{-1}R^{-1} = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}.$$

This is all we need in order to compute  $B^\#$ , using Theorem 1.3.2, and

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

We get

$$B^\# = \begin{pmatrix} 0 & -1 & 1 \\ -1 & -1 & 2 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad A^\# = \begin{pmatrix} 1 & -1 & 1 \\ 1 & -1 & 2 \\ -1 & 1 & -1 \end{pmatrix}.$$

We should keep in mind, however, that in general we will not be interested in obtaining the numerical values of  $B^\#$  and  $A^\#$ . The factorization will be applied directly to the right hand side  $b$  to compute a solution, using the algorithms of the next section. The computational effort required by our decomposition algorithm is shown in Figure 1.3.4.

- Step 1.1.  $\frac{1}{3}(n^3 - m^3)$  flops +  $\frac{1}{2}n(n - m)$  swaps +  $\frac{1}{2}n^2$  comparisons  
 Step 1.2.  $\frac{1}{2}(n - m)n^2$  flops +  $(n - m)n$  swaps  
 Step 2.1.  $\frac{1}{3}(n - m)^3$  flops +  $\frac{1}{2}(n - m)^2$  swaps +  $\frac{1}{2}(n - m)^2$  comparisons

**Figure 1.3.4. Operations count for the slow decomposition algorithm.**

The count for Step 1.1 is for the worst case, that is, when the matrix  $U_{11}$  happens to be nonsingular. This is the case when all the zero pivots are obtained in the last  $m$  stages of the Gaussian elimination process. Then the whole process is the same as for a nonsingular matrix, except that the last  $m$  stages do not need to be done. This takes  $\frac{1}{3}m^3$  flops off the  $\frac{1}{3}n^3$  flops required for a nonsingular matrix. Also, Step 1.2 can be implemented such that  $P$  and  $L$  are applied to each one of the  $(n - m)$  rows of  $U$  in turn. That is, the elements of row 1 of  $U$  are permuted (by  $P$ ), and the multipliers are applied (i.e.,  $L$ ), giving the first row of  $B$ . Then the process is repeated for row 2, and so forth. Finally, Step 2.1 is just the regular Gaussian elimination of a nonsingular  $(n - m) \times (n - m)$  matrix, and Step 2.2 does not require any arithmetic.

Now let us consider the case when  $m$  is much smaller than  $n$ . This is the case when  $m = 0$  (nonsingular  $A$ ) and  $m = 1$  (unichain Markov process), for instance, and also whenever there are several states per recurrent subchain, in a dynamic programming problem. With  $m = 0$ , the total number of flops is  $\frac{1}{3}n^3 + \frac{1}{2}n^3 + \frac{1}{3}n^3 = \frac{7}{6}n^3$  flops. This is 3.5 times the number of flops required for applying Gaussian elimination to a nonsingular matrix (i.e.,  $\frac{1}{3}n^3$  flops). This fact justifies the name *slow algorithm* that we have been using.

Our second algorithm, which we call the *fast algorithm*, will use only  $\frac{1}{3}n^3$  flops when  $m$  is small, the same as when applying Gaussian elimination to a nonsingular matrix. It is based on a modified Gaussian elimination procedure in which not only the rows, but also the columns are interchanged. Hence for a singular matrix  $A$  such

that  $\text{rank}(A) = n - m$ , the modified procedure obtains the four matrices  $P$ ,  $L$ ,  $U$  and  $Q$  such that  $A = PLUQ$ , where  $P$  and  $Q$  are permutation matrices,

$$L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} U_{11} & U_{12} \\ 0 & 0 \end{pmatrix}$$

with  $L_{11}$  a unit lower triangular matrix and  $U_{11}$  a nonsingular upper triangular matrix.

Specifically, the Gaussian elimination procedure is modified as follows. Suppose the first zero column is encountered after  $k$  elimination steps have been performed. For example, with  $n = 5$  and  $k = 2$ , we have a matrix  $A^{(k)}$  with the following pattern:

$$A^{(2)} = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x \end{pmatrix},$$

where the nonzero entries are denoted by  $x$ .

Now let  $Q^{(1)}$  be the permutation matrix that exchanges the columns  $k + 1$  and  $n$ .

We get

$$A^{(2)}Q^{(1)} = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & 0 \\ 0 & 0 & x & x & 0 \\ 0 & 0 & x & x & 0 \end{pmatrix},$$

and the elimination procedure is resumed. Suppose that the second zero column is encountered after the  $\ell$ -th elimination step. Then  $Q^{(2)}$  exchanges the columns  $\ell + 1$  and  $n - 1$ .

Similarly,  $Q^{(i)}$  will exchange the  $i$ -th zero column with column  $n - i + 1$ . At the end, we have  $A = PLUQ$ , where  $Q = Q^{(1)}Q^{(2)} \dots Q^{(m)}$ . With  $n = 5$  and  $m = 2$ , we have

$$U = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

as required.

To illustrate the idea, let us apply modified Gaussian elimination to the  $3 \times 3$  matrix of the last example. We get  $A = PLUQ$ , where  $P = I$ ,

$$U = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

and  $Q$  interchanges the columns 2 and 3 of  $A$ . Here  $U_{11} = I$  is indeed nonsingular.

This modified Gaussian elimination procedure PLUQ does exactly the same number of flops as the original PLU procedure, and requires only an extra  $mn$  swaps to exchange the columns. But the fact that  $U_{11}$  is nonsingular will allow us to find a representation for  $B_{11}^{-1}$  which requires the PLU factorization of a small  $m \times m$  matrix, instead of the large  $(n - m) \times (n - m)$  matrix  $B_{11}$  itself. The idea is based on the following result of Faddeeva (1959), for inverting a matrix by partitioning. The proof is included for the sake of completeness.

**Lemma 1.3.3.** Let  $X$  be a nonsingular  $n \times n$  matrix and  $Y$  its inverse, such that

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}.$$

Suppose also that  $X_{11}$  is nonsingular. Then  $Y_{22}$  is also nonsingular and

$$X_{11}^{-1} = Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}. \quad (1.3.11)$$

**Proof.** Since  $X_{11}$  is nonsingular, then  $X$  is nonsingular iff the matrix

$$\begin{pmatrix} I & 0 \\ -X_{21}X_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} \\ 0 & X_{22} - X_{21}X_{11}^{-1}X_{12} \end{pmatrix}$$

is nonsingular too, which is the case iff  $(X_{22} - X_{21}X_{11}^{-1}X_{12})$  is nonsingular. Now writing the condition  $XY = I$ , we get

$$X_{11}Y_{11} + X_{12}Y_{21} = I \quad (1.3.12)$$

$$X_{11}Y_{12} + X_{12}Y_{22} = 0 \quad (1.3.13)$$

$$X_{21}Y_{11} + X_{22}Y_{21} = 0$$

$$X_{21}Y_{12} + X_{22}Y_{22} = I. \quad (1.3.14)$$

Using (1.3.13), we have  $Y_{12} = -X_{11}^{-1}X_{12}Y_{22}$ , so that (1.3.14) becomes

$$(X_{22} - X_{21}X_{11}^{-1}X_{12})Y_{22} = I,$$

which implies that  $Y_{22}$  is nonsingular. Using (1.3.13) again, we have

$$X_{12} = -X_{11}Y_{12}Y_{22}^{-1}.$$

Finally, Equation (1.3.11) follows by substituting  $X_{12}$  into Equation (1.3.12).  $\square$

For our fast algorithm, we need to construct a matrix  $X$  such that  $X_{11} = B_{11}$  and for which the inverse  $Y$  is easily obtained. Then  $B_{11}^{-1}$  can be represented using Equation (1.3.11). We proceed as follows. Suppose that  $A = PLUQ$ , and let  $C = QPL$ .

Then write

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

We want  $B = (PL)^{-1}A(PL) = UQPL = UC$ , so that

$$B = \begin{pmatrix} U_{11} & U_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix}$$

as in Equation (1.3.4), with  $S = PL$ . Now define  $D$  such that

$$D = \begin{pmatrix} U_{11} & U_{12} \\ 0 & I \end{pmatrix}.$$

Then  $D$  is nonsingular, because  $U_{11}$  is nonsingular in the PLUQ decomposition, and we have

$$D^{-1} = \begin{pmatrix} U_{11}^{-1} & -U_{11}^{-1}U_{12} \\ 0 & I \end{pmatrix}.$$

We finally obtain our matrix  $X = DC$ , and it is easy to verify that

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

Now its inverse  $Y = X^{-1}$  is obtained directly, because

$$Y = C^{-1}D^{-1} = L^{-1}P^{-1}Q^{-1}D^{-1}. \quad (1.3.15)$$

**Theorem 1.3.4.** Both  $B_{11}$  and  $Y_{22}$  are nonsingular iff  $\text{ind}(A) = 1$ . Moreover,

$$B_{11}^{-1} = Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}.$$

**Proof.** Immediate from Lemmas 1.3.1 and 1.3.3. □

We now have all the mathematical tools necessary for constructing our fast algorithm. Still, if we implement Equation (1.3.15) to produce the whole matrix  $Y$ , it will cost  $\frac{1}{2}(n-m)^2n$  flops to compute  $D^{-1}$  plus  $\frac{1}{2}n^3$  to do the product by  $L^{-1}$ . The total effort for such an algorithm would then be  $\frac{4}{3}n^3$  flops (including the initial PLUQ decomposition and ignoring the terms of degree less than 3 in  $n$ ). This is even worse than the long algorithm! The key here is that we need only to produce  $Y_{22}$ , which can be done as follows:

$$\begin{pmatrix} Y_{12} \\ Y_{22} \end{pmatrix} = L^{-1}P^{-1}Q^{-1} \begin{pmatrix} -U_{11}^{-1}U_{12} \\ I \end{pmatrix}. \quad (1.3.16)$$

The computations are performed as follows. Back substitution is applied to each of the  $m$  columns of  $U_{12}$  in turn. Then the permutation matrices  $Q^{-1}$  and  $P^{-1}$  are applied. Finally, the multipliers of the Gaussian elimination are applied.

We get even more than we need: we have to compute  $Y_{12}$ , in addition to  $Y_{22}$ , because the matrix  $L^{-1}$  is represented in product form (so that  $L_{21}$  is not known). Even so, the total cost of computing (1.3.16) is  $\frac{1}{2}m(n-m)^2 + \frac{1}{2}mn^2$  flops, which is negligible when  $m$  is small. Hence for  $m$  small, the fast algorithm costs only  $\frac{1}{3}n^3$  flops, which is the cost of the initial PLUQ decomposition. The fast algorithm is shown in Figure 1.3.5.

Let us continue our  $3 \times 3$  example. We have

$$D = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } D^{-1} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

so that

$$Y = L^{-1}P^{-1}Q^{-1}D^{-1} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Hence we have

$$Y_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad Y_{12} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad Y_{21} = (1 \quad 1) \text{ and } Y_{22} = (1).$$

- Step 1.1. Compute  $P, L, U$  and  $Q$  such that  $A = PLUQ$   
 (using modified Gaussian elimination, with  $U_{11}$  nonsingular).
- Step 1.2. Compute  $Y_{22}$  as in Equation (1.3.16)
- Step 2.1. Compute  $R, S$  and  $T$  such that  $Y_{22} = RST$   
 (using Gaussian elimination).  
 (Note:  $R$  is permutation,  $S$  lower,  $T$  upper triangular).
- Step 2.2. If  $Y_{22}$  is singular then "Error:  $\text{ind}(A) > 1$ "  
 else stop (we have all the necessary information).

**Figure 1.3.5. The fast decomposition algorithm.**

Moreover,

$$B = UQPL = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

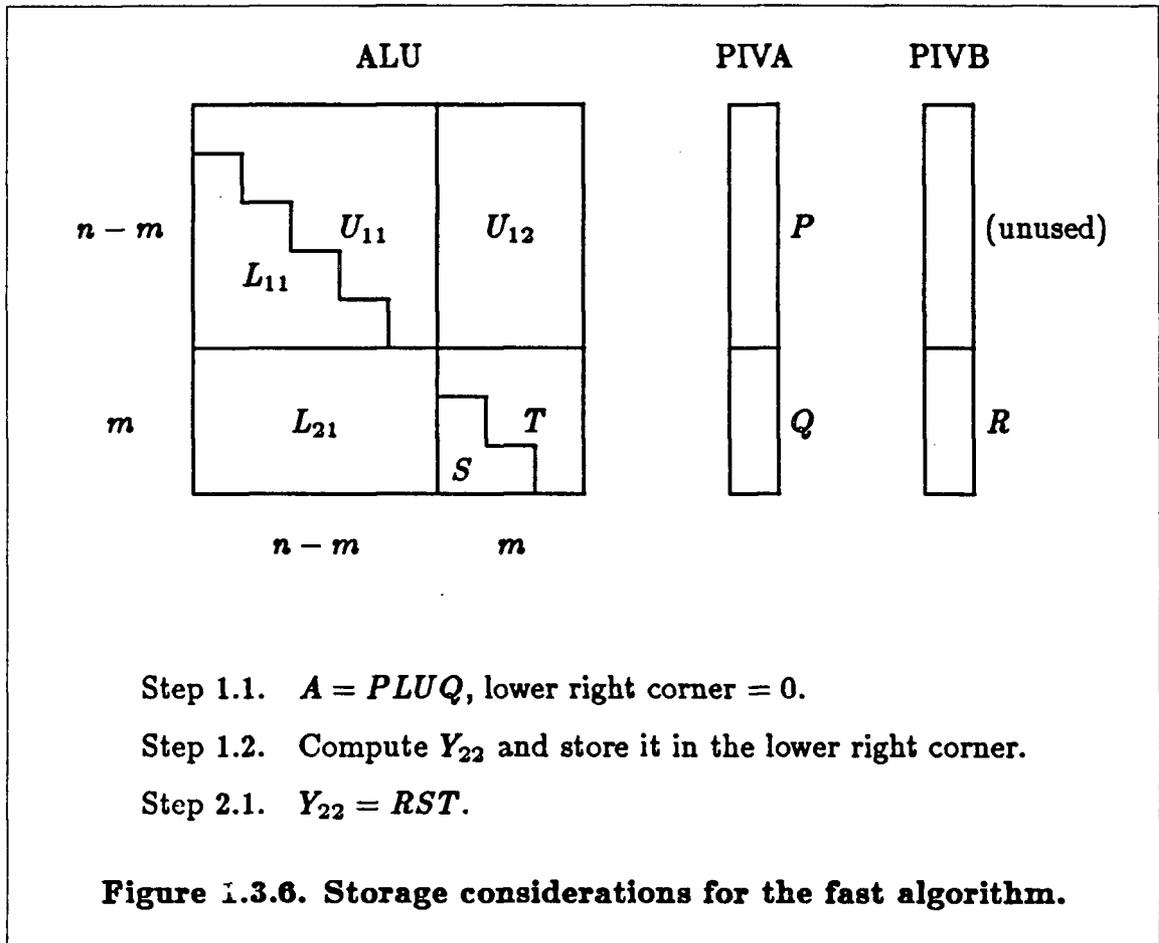
and we have  $B_{11}^{-1} = Y_{11} - Y_{12}Y_{22}^{-1}Y_{21}$ . Indeed,

$$Y_{11} - Y_{12}Y_{22}^{-1}Y_{21} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix} = B_{11}^{-1}.$$

The algorithms of the next section, however, will not produce  $Y_{11}$  and  $Y_{12}$ , applying rather the product representation directly to the right hand side  $b$ .

The organization of the computations is worth some discussion. It is possible to perform all the computations of the fast algorithm using one real array ALU of dimension  $n \times n$ , two integer arrays PIVA and PIVB of dimension  $n$  and one real array WK of dimension  $n$ . Figure 1.3.6 shows how the various submatrices can be stored.

The PLUQ decomposition of Step 1.1 can be organized to overwrite the original matrix  $A$  as in the nonsingular case, and we get an  $m \times m$  lower right submatrix of zeroes. We can also implement Step 1.2 so that it produces one column of  $Y$  at a time, storing it in the real array WK (which is not shown in the figure). The  $(n - m)$



components that belong to  $Y_{12}$  are discarded, and we need only to store the  $m$  entries corresponding to  $Y_{22}$ . Those are conveniently stored in the zero lower right corner of ALU. As we will see in the next section, there is no loss of efficiency in dropping the submatrix  $Y_{12}$ , as it will be just as efficient to apply it to a vector using the product representation. Finally, the  $PLU$  decomposition of  $Y_{22}$  can be done so that  $S$  and  $T$  overwrite  $Y_{22}$ . The permutation matrices  $P$ ,  $Q$  and  $R$  are represented by the integer vectors PIVA and PIVB.

The slow algorithm, on the other hand, cannot be organized as efficiently. This is because the submatrices  $B_{11}$  and  $B_{12}$  have to be produced explicitly. While  $B_{12}$  may very well overwrite  $U_{12}$ , it is not possible for  $B_{11}$  to overwrite  $L_{11}$  and  $U_{11}$ . This

is because while  $U_{11}$  and  $U_{12}$  will not be required any more after  $B$  is produced, the matrix  $L$  will still have to be applied when computing a solution, as will be seen in the next section. Hence not only is the long algorithm inefficient in terms of computational effort, but it also needs more memory than the fast algorithm.

One aspect of both algorithms which we have not considered yet is the effect of rounding errors on the Gaussian elimination of a singular matrix. The problem is that, due to rounding errors, it is likely that the entries which would be set to exactly zero when using exact arithmetic will in fact be set to some small, nonzero numbers. The problem is to select a tolerance level, say  $\epsilon$ , such that an entry will be considered zero if its absolute value is less than  $\epsilon$ .

Some theoretical bounds are known when the elimination is done using the QR method, or singular value decomposition [see Stewart (1984)]. This device is not very reliable, however, when applying Gaussian elimination to a general singular matrix  $A$ . In fact, Gaussian elimination with partial pivoting is considered to be numerically unstable when  $A$  is singular [see Golub and Van Loan (1983)]. The question, then, is why did we develop our algorithms using Gaussian elimination.

The answer is three-fold. First, Gaussian elimination is mathematically well defined and is known to the expected audience of this thesis. Second, the formulation of our algorithm in terms of QR decomposition or singular value decomposition is straightforward. The QR version of the fast algorithm requires  $\frac{2}{3}n^3$  flops, twice that of Gaussian elimination. Hence our results about the operations count need only be multiplied by a factor of 2 if the QR method is used. Third, the intended application of our algorithms is with a matrix  $A = I - P$ , where  $P$  is stochastic (see Chapter II). In this case, the rank of  $A$  is known in advance of the elimination. Also, Gaussian elimination is known to be stable even without row interchanges [see Harrod and Plemmons (1984) and Funderlic and Plemmons (1981)]. (Of course, row interchanges are necessary for our application, so that the zeroes be at the bottom).

#### 1.4. Solution of singular systems of equations

Consider an  $n \times n$  matrix  $A$  of index 1 and an arbitrary vector  $b$ , and let  $c = Sb$ , where  $S$  is the transformation matrix of Equation (1.3.1). We are interested in the system of equations  $Ax = b$ . Writing  $y = Sx$ , we have

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix},$$

where

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \text{ and } c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Hence the system has a solution iff  $c_2 = 0$ . Moreover, if the solution exists, it has  $y_1 = B^{-1}c_1$ , and  $y_2$  is arbitrary.

For an arbitrary vector  $b$ , compatible with  $A$  or not, we define the *gain* as the vector  $g$  such that

$$g = S^{-1} \begin{pmatrix} 0 \\ c_2 \end{pmatrix} = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} Sb = Wb$$

by Equation (1.2.8). Hence a vector  $b$  is compatible with  $A$  iff  $g = 0$ . Moreover, we define the *bias* as the vector  $h$  such that

$$h = S^{-1} \begin{pmatrix} B^{-1}c_1 \\ 0 \end{pmatrix} = S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} Sb = A\#b,$$

by Equation (1.3.2).

In this context, the *gain* is the projection of the right-hand side onto the null space of  $A$ , while the *bias* is the unique solution which belongs to the range of  $A$ . That is, of all the possible solutions,  $h$  is the only one such that  $Wh = 0$ . This property of the Drazin inverse solution is crucial for the dynamic programming application, for which other generalized inverses would not be appropriate.

The terms *gain* and *bias* are well known in dynamic programming and their interpretation in this context will be considered in Chapter 3. We will now construct two algorithms for computing the gain and the bias for a given vector  $b$ , using the decompositions of the matrix  $A$  obtained with the algorithms of the previous section.

These decompositions have the form of Equation (1.3.4), so that  $A^\#$  and  $W$  are given by (1.3.8) and (1.3.9), respectively. Hence we have

$$\begin{aligned} g = Wb &= S \begin{pmatrix} 0 & -B_{11}^{-1}B_{12} \\ 0 & I \end{pmatrix} S^{-1} S \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \\ &= S \begin{pmatrix} -B_{11}^{-1}B_{12}c_2 \\ c_2 \end{pmatrix} \end{aligned} \quad (1.4.1)$$

and

$$\begin{aligned} h = A^\# b &= S \begin{pmatrix} B_{11}^{-1} & B_{11}^{-1}B_{11}^{-1}B_{12} \\ 0 & 0 \end{pmatrix} S^{-1} S \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \\ &= S \begin{pmatrix} B_{11}^{-1}(c_1 + B_{11}^{-1}B_{12}c_2) \\ 0 \end{pmatrix}, \end{aligned} \quad (1.4.2)$$

where  $c = S^{-1}b$ . In both decomposition algorithms, the transformation matrix is  $S = PL$ , with  $P$  and  $L$  obtained at Step 1.1 by Gaussian elimination. The corresponding solution algorithms will have three major steps: transformation of the right hand side, computation of the transformed solution vectors, and finally, back transformation of the solution vectors. The *slow* solution algorithm is described in Figure 1.4.1.

- |  |
|--|
| <p>Step 1.1. Compute <math>c = L^{-1}P^{-1}b</math>.</p> <p>Step 2.1. Compute <math>u_1 = B_{12}c_2</math>.</p> <p>Step 2.2. Compute <math>y_1 = T^{-1}S^{-1}R^{-1}u_1</math>.</p> <p>Step 2.3. Compute <math>z_1 = T^{-1}S^{-1}R^{-1}(c_1 + y_1)</math>.</p> <p>Step 3.1. Compute <math>g = PL \begin{pmatrix} -y_1 \\ c_2 \end{pmatrix}</math>.</p> <p>Step 3.2. Compute <math>h = PL \begin{pmatrix} z_1 \\ 0 \end{pmatrix}</math>.</p> |
|--|

**Figure 1.4.1. The slow solution algorithm**

That it effectively gives the gain and bias vectors as in (1.4.1) and (1.4.2) follows from the fact that  $B_{11} = RST$ , where  $R$ ,  $S$  and  $T$  are the matrices produced by using Gaussian elimination, at Step 2.1 of the long decomposition algorithm. The total

number of floating point operations required to compute the gain and the bias is then

$$\frac{1}{2}n^2 + m(n - m) + 2(n - m)^2 + n^2.$$

Neglecting the terms of degree less than 2 in  $n$ , we get  $3.5n^2$  flops when  $m$  is small. The standard solution of a nonsingular system, with  $g = 0$  and  $h = A^{-1}b$ , would cost only  $n^2$  flops. It is to be emphasized, however, that it is the decomposition which has the dominant cost, with  $3.5n^3$  flops for the slow algorithm.

To illustrate the solution process, let us consider again the matrix

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix}, \text{ with } b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

The matrices  $P$ ,  $R$ ,  $S$ ,  $T$  and  $B_{12}$  were obtained by the slow decomposition algorithm. Applying the slow solution algorithm, we obtain:

$$\begin{aligned} c &= L^{-1}P^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} \\ u_1 &= B_{12}c_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} (4) = \begin{pmatrix} 0 \\ 4 \end{pmatrix} \\ y_1 &= T^{-1}S^{-1}R^{-1}u_1 = \begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} -4 \\ -4 \end{pmatrix} \\ z_1 &= \begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1-4 \\ 2-4 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \\ g &= PL \begin{pmatrix} -y_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 0 \end{pmatrix} \\ h &= PL \begin{pmatrix} z_1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ -2 \end{pmatrix}. \end{aligned}$$

Let us now proceed to the construction of a solution algorithm that will use the output of the fast decomposition algorithm. This fast solution algorithm will work much like the algorithm of Figure 1.4.1, except that we do not have the matrix  $B_{12}$ . Moreover, the fast decomposition is based on the idea that

$$B_{11}^{-1} = Y_{11} - Y_{12}Y_{22}^{-1}Y_{21},$$

but we do not have  $Y_{11}$ ,  $Y_{12}$  and  $Y_{21}$  [although  $Y_{12}$  had been produced in Equation (1.3.16) and then discarded to save memory storage]. We do have  $Y_{22} = RST$ , Where  $R$ ,  $S$  and  $T$  were obtained by Gaussian elimination at Step 2.1 of the fast decomposition algorithm.

But we still need to derive a way to express  $B_{12}$ ,  $Y_{11}$ ,  $Y_{12}$  and  $Y_{21}$  in terms of  $P$ ,  $L$ ,  $U$  and  $Q$ , which were obtained at Step 1.1. We do this as follows. First, recall that  $B = UQPL$ , with

$$B = \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} \\ 0 & 0 \end{pmatrix} \text{ and } L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I \end{pmatrix}.$$

We do not know  $L_{11}$  and  $L_{12}$  explicitly, but we do not need them either. It is the structure of  $L$  which is important here, and we have

$$\begin{aligned} B_{12}x_2 &= (B_{11} \quad B_{12}) \begin{pmatrix} 0 \\ x_2 \end{pmatrix} = (U_{11} \quad U_{12})QP \begin{pmatrix} L_{11} & 0 \\ L_{21} & I \end{pmatrix} \begin{pmatrix} 0 \\ x_2 \end{pmatrix} \\ &= (U_{11} \quad U_{12})QP \begin{pmatrix} 0 \\ x_2 \end{pmatrix}. \end{aligned} \quad (1.4.3)$$

Hence the product  $B_{12}x_2$  can be done efficiently by first permuting the vector and then multiplying it by  $U_{11}$  and  $U_{12}$ .

Next, the product by  $Y_{12}$  can be done using Equation (1.3.16). We get  $y_1 = Y_{12}x_2$  by computing

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} Y_{12}x_2 \\ Y_{22}x_2 \end{pmatrix} = L^{-1}P^{-1}Q^{-1} \begin{pmatrix} -U_{11}^{-1}U_{12}x_2 \\ x_2 \end{pmatrix}. \quad (1.4.4)$$

We also get  $y_2$ , and since we do not need it, we just discard it. Similarly, we apply  $Y_{11}$  and  $Y_{21}$  to a vector as follows, using (1.3.15) to get

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} Y_{11}x_1 \\ Y_{21}x_1 \end{pmatrix} = L^{-1}P^{-1}Q^{-1} \begin{pmatrix} U_{11}^{-1}x_1 \\ 0 \end{pmatrix}. \quad (1.4.5)$$

The fact that  $Y_{11}$  and  $Y_{21}$  must be applied simultaneously to the same vector  $x_1$  is not a problem. It is indeed exactly what we need because by Theorem 1.3.4, we have

$$\begin{aligned} B_{11}^{-1}x_1 &= (Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})x_1 \\ &= (Y_{11}x_1) - Y_{12}Y_{22}^{-1}(Y_{21}x_1). \end{aligned}$$

- Step 1. Compute  $z_1 = Y_{11}x_1$  and  $z_2 = Y_{21}x_1$  using Equation (1.4.5).  
 Step 2. Compute  $v_2 = T^{-1}S^{-1}R^{-1}z_2$ .  
 Step 3. Compute  $w_1 = Y_{12}v_2$  using Equation (1.4.4).  
 Step 4. Compute  $u_1 = z_1 - w_1$ .

**Figure 1.4.2. Algorithm to compute  $u_1 = B_{11}^{-1}x_1$ .**

- Step 1.1. Compute  $c = L^{-1}P^{-1}b$ .  
 Step 2.1. Compute  $u_1 = B_{12}c_2$  using Equation (1.4.3).  
 Step 2.2. Compute  $y_1 = B_{11}^{-1}u_1$  as in Figure 1.4.2.  
 Step 2.3. Compute  $z_1 = B_{11}^{-1}(c_1 + y_1)$  as in Figure 1.4.2.  
 Step 3.1. Compute  $g = PL\begin{pmatrix} -y_1 \\ c_2 \end{pmatrix}$ .  
 Step 3.2. Compute  $h = PL\begin{pmatrix} z_1 \\ 0 \end{pmatrix}$ .

**Figure 1.4.3. The fast solution algorithm.**

This leads to the algorithm of Figure 1.4.2 for computing  $u_1 = B_{11}^{-1}x_1$ .

The matrices  $R$ ,  $S$  and  $T$  of Step 2 were obtained by Gaussian elimination of  $Y_{22}$ , at Step 2.1 of the fast decomposition algorithm, in Figure 1.3.5. The fast solution algorithm is shown in Figure 1.4.3.

Observe that the algorithm of Figure 1.4.2 for applying  $B_{11}^{-1}$  to a vector is used twice as a subroutine by the fast solution algorithm. The total number of flops used for this product is

$$\begin{aligned} & \left[ \frac{1}{2}n^2 + \frac{1}{2}(n-m)^2 \right] + m^2 + \left[ \frac{1}{2}n^2 + \frac{1}{2}(n-m)^2 + m(n-m) \right] \\ & = n^2 + (n-m)^2 + mn. \end{aligned}$$

Hence the total number of flops required by the fast solution algorithm is

$$\begin{aligned} & \frac{1}{2}n^2 + \left[\frac{1}{2}(n-m)^2 + m(n-m)\right] + 2[n^2 + (n-m)^2 + mn] + 2\left[\frac{1}{2}n^2\right] \\ & = 6n^2 - 2mn + 1.5m^2. \end{aligned}$$

When  $m = 0$ , the fast solution algorithm takes then  $6n^2$  flops, which is more work than the slow solution algorithm. We must keep in mind, however, that the fast decomposition algorithm takes only  $\frac{1}{3}n^3$  flops. Hence, to compute the gain and bias of a singular system by the fast algorithm takes about the same amount of flops as for solving a nonsingular system with 6 right hand sides.

It should be clear from Figure 1.4.3 that the only difference between our two solution algorithms is the manner in which the products by  $B_{12}$  and  $B_{11}^{-1}$  are implemented. In the slow algorithm,  $B_{12}$  is used explicitly and the product by  $B_{11}^{-1}$  is produced from the  $PLU$  decomposition of  $B_{11}$  (i.e.,  $R$ ,  $S$  and  $T$ ). In the fast algorithm, however, the product by  $B_{12}$  is done by Equation (1.4.3), and the product by  $B_{11}^{-1}$  is done by the algorithm of Figure 1.4.2.

Let us apply those products to our numerical example. Recall that

$$U = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix},$$

$P = I$  and  $Q$  interchanges columns (or rows) 2 and 3. Also,  $Y_{22} = (1)$  and  $R = S = T = (1)$ . Let us do Step 2.1 of the fast solution algorithm. We have  $c_2 = (4)$ , so that

$$\begin{aligned} u_1 = B_{12}c_2 &= \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} QP \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \end{aligned}$$

which is the same value as with the long algorithm. Now we do Step 2.2 to get  $y_1 =$

$B_{11}^{-1}u_1$ :

$$\begin{aligned}
z &= L^{-1}P^{-1}Q^{-1} \begin{pmatrix} U_{11}^{-1}u_1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} \\
z_1 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad z_2 = (4) \\
v_2 &= T^{-1}S^{-1}R^{-1}z_2 = (4) \\
w &= L^{-1}P^{-1}Q^{-1} \begin{pmatrix} -U_{11}^{-1}U_{12}v_2 \\ v_2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix} \\
w_1 &= \begin{pmatrix} 4 \\ 4 \end{pmatrix} \\
y_1 &= z_1 - w_1 = \begin{pmatrix} 0 - 4 \\ 0 - 4 \end{pmatrix} = \begin{pmatrix} -4 \\ -4 \end{pmatrix}.
\end{aligned}$$

This value of  $y_1$  is again the same as that we obtained with the long algorithm. Finally, Step 2.3 would work the same way, giving  $z_1$  as before.

### 1.5. Other algorithms for Drazin inversion

Both the long and the fast algorithms of the previous sections are specializations of the general decomposition algorithm of Wilkinson (1982). We implemented them using Gaussian elimination, but we could have used singular value decomposition, or the  $QR$  decomposition [see e.g., Stewart (1984)], just as well. In fact, the singular value decomposition version of Wilkinson's method is described in Campbell and Meyer (1979), Algorithm 12.5.1. We used Gaussian elimination because it is more economical than both other decompositions [see e.g., Johnston (1982)], and it is stable when  $A = I - P$ , with  $P$  stochastic.

The shuffle algorithm of Anstreicher and Rothblum (1985) uses a different approach. Instead of producing a factorization of the Drazin inverse in terms of elementary matrices, it produces a full matrix from which the Drazin inverse is obtained. More

specifically, for a matrix  $A$  of index  $k$ , the algorithm performs  $k$  steps in which Gaussian elimination and a “shuffle” operation are used, in which a matrix  $B$  is produced, such that  $A^D = B^{k+1}A^k$ , at a cost of  $n^3$  flops (if  $m$  is small). This is three times the amount of work required for applying Gaussian elimination to a nonsingular matrix. Recall that our slow algorithm is just slightly slower, with a ratio of 3.5, and that our fast algorithm is three times as fast (with a ratio of 1).

In the special case of interest, with  $\text{ind}(A) = 1$ , the gain and bias vectors can be computed as follows, for a given right hand side  $b$ :

$$h = A^\# b = B^2 A b$$

$$g = W b = (I - A A^\#) b = b - A h.$$

Hence the bias can be computed with  $3n^2$  flops, and the gain with another  $n^2$  flops, for a total of  $4n^2$  flops. Recall that the slow solution algorithm of Section 1.4 takes only  $3n^2$  flops, while the fast algorithm takes  $6n^2$  flops.

Other methods are based on sequences of matrix multiplications and are not computationally attractive. See Campbell and Meyer (1979), Greville (1973), Hartwig (1976) and Anstreicher and Rothblum (1985).

## Chapter II

### Nonnegative matrix theory

In this chapter, we show how the Perron-Frobenius theory of nonnegative matrices can be used to deduce the eigenvalue structure of the transition matrix of a Markov chain. The matrix decomposition so obtained leads to a simple characterization of the limiting and fundamental matrices, using the Drazin inverse notion of Chapter I. Further, the Jordan canonical form is used to derive the geometric convergence of the power method.

First, the ideas of reducibility and periodicity of a matrix are reviewed, in Section 2.1, and the correspondence between matrices and directed graphs is established.

Next, the theory of nonnegative matrices is reviewed in Section 2.2. Our approach is inspired from Varga (1962), but the derivations we provide are consistent with our exposition of the Jordan canonical form in Chapter I. Lemmas 2.2.1, 2.2.2, 2.2.3 and 2.2.8 are given without proof. Although elementary, their derivation is not directly relevant to the material of the thesis.

In Section 2.3, the matrix decomposition approach is used in a new derivation of the properties of the limiting matrix, the deviation matrix and the fundamental matrix of a discrete time Markov chain. We show that the probabilistic interpretation of these matrices, using Cesàro limits, requires the boundedness of the powers of the transition matrix. This property is not required for defining the matrices in terms of the Drazin inverse, though.

Finally, it is shown, in Section 2.4, that when the transition matrix is aperiodic, its

powers converge to the limiting matrix, at a geometric rate. This result is a specialization of a result of Varga (1962).

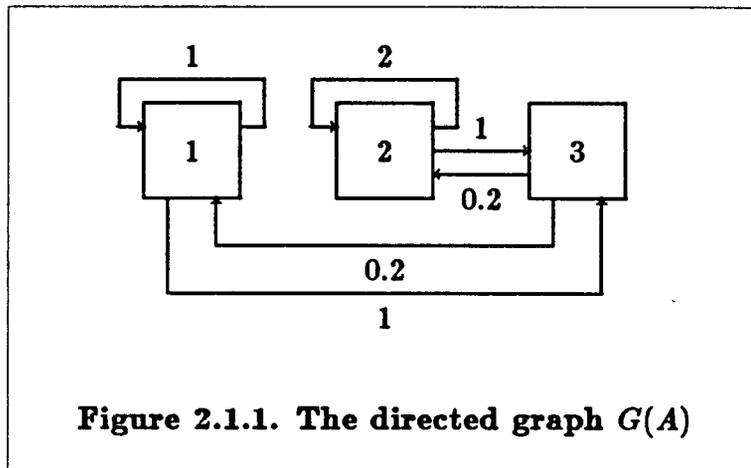
## 2.1. Matrices and directed graphs

It is often convenient to associate the entries of a complex  $n \times n$  matrix  $A$  with a directed graph  $G(A)$  with  $n$  nodes, labelled  $1, \dots, n$ , and with a directed arc for every nonzero entry of  $A$ . Hence  $G(A)$  has an arc from node  $i$  to node  $j$ , labelled  $a_{ij}$ , iff  $a_{ij} \neq 0$ , for  $i = 1, \dots, n$  and  $j = 1, \dots, n$ .

For example, the matrix

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0.2 & 0.2 & 0 \end{pmatrix}$$

has the directed graph of Figure 2.1.1.



In the context of queueing and dynamic programming, the directed graph naturally suggests that the nodes represent each a state of a stochastic system, while the arcs represent a possible transition from one state to another. Hence the entries of  $A$  are transition rates or probabilities (depending on the specific application).

We have already used the directed graph representation of a matrix in Chapter I (see Figure 1.1.1). The first well known result of interest is as follows.

**Lemma 2.1.1.** Let  $A$  and  $B$  be two complex  $n \times n$  matrices. Then there is a permutation matrix  $T$  such that  $B = TAT^{-1}$  iff  $G(B)$  is the same labelled graph as  $G(A)$ , with the node labels permuted.

**Proof.** Left to the reader. □

Let  $N = \{1, \dots, n\}$  be the set of all the nodes of a directed graph  $G$ . This set  $N$  can be partitioned into  $m$  equivalence classes  $S_k$ ,  $k = 1, \dots, m$ , with  $1 \leq m \leq n$ , such that two nodes  $i, j \in N$  are equivalent iff there is a path from  $i$  to  $j$  and a path from  $j$  to  $i$ . The equivalence classes  $S_k$ ,  $k = 1, \dots, m$ , are called *strongly connected components* [some authors use this term for the subgraphs  $(S_k, E_k)$  where  $E_k$  is the set of all arcs between the nodes in  $S_k$ ]. If a graph  $G$  has exactly one strongly connected component (i.e.,  $m = 1$ ), then  $G$  is said to be *strongly connected*.

There is a similar concept in matrix theory. A complex  $n \times n$  matrix  $A$  is said to be *irreducible* if there is no permutation matrix  $T$  such that  $A = TBT^{-1}$  with

$$B = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix}, \quad (2.1.1)$$

where  $B_{11}$  and  $B_{22}$  are square submatrices. If such a matrix  $T$  exists, then  $A$  is said to be *reducible*.

**Lemma 2.1.2.** A matrix  $A$  is irreducible iff its directed graph  $G(A)$  is strongly connected.

**Proof.** If  $A$  is not irreducible then let  $S_1$  and  $S_2$  be the subsets of  $N$  corresponding to  $B_{11}$  and  $B_{22}$  of Equation (2.1.1). Let  $i \in S_1$  and  $j \in S_2$ . Since  $B_{21} = 0$ , there is no path going from  $j$  to  $i$  and hence  $G(A)$  is not strongly connected. To show the converse, suppose there is no path from  $j$  to  $i$ . Let  $S_2$  be the strongly connected component of  $G(A)$  that contains  $i$ , and let  $S_1 = N - S_2$ . Then a matrix  $B$  satisfying (2.1.1) can be constructed, and hence  $A$  is not irreducible. □

Suppose a graph  $G$  has  $m > 1$  strongly connected components. Suppose also that there is an arc from some state  $i \in S_k$  to some state  $j \in N - S_k$ . Then  $S_k$  is a *transient*

class and all the nodes in  $S_k$  are said to be *transient*. If no such outgoing arcs originate from  $S_k$ , then  $S_k$  is a *recurrent* class and all the nodes in  $S_k$  are recurrent.

**Theorem 2.1.3.** Suppose  $A$  is a complex  $n \times n$  matrix. Suppose also that  $G(A)$  has  $\ell$  transient classes and  $m$  recurrent classes. Then there is a permutation matrix  $T$  such that

$$A = T \begin{pmatrix} B & C \\ 0 & D \end{pmatrix} T^{-1}, \quad (2.1.2)$$

with

$$B = \begin{pmatrix} B_{11} & B_{12} & \dots & B_{1\ell} \\ 0 & B_{22} & \dots & B_{2\ell} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & B_{\ell\ell} \end{pmatrix} \text{ and } D = \begin{pmatrix} D_{11} & 0 & \dots & 0 \\ 0 & D_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & D_{mm} \end{pmatrix}, \quad (2.1.3)$$

where  $B_{11}, \dots, B_{\ell\ell}$  and  $D_{11}, \dots, D_{mm}$  are irreducible, square submatrices.

**Proof.** Each connected component of  $G(A)$  has an associated submatrix. The  $\ell$  transient classes correspond to the submatrices  $B_{11}, \dots, B_{\ell\ell}$ , and the  $m$  recurrent classes correspond to  $D_{11}, \dots, D_{mm}$ . The submatrix  $D$  is block diagonal because there are no outgoing arcs from the recurrent classes. The submatrix  $B$  is block upper triangular because if both  $B_{ij}$  and  $B_{ji}$  were nonzero, with  $1 \leq i < j \leq \ell$ , then all the nodes corresponding to the submatrices  $B_{ii}$  and  $B_{jj}$  would belong to the same strongly connected component. Hence  $G(A)$  would have only  $\ell - 1$  transient classes. This implies that at least one block must be zero, and we can sort the components so that  $B_{ij} = 0$ .  $\square$

This parsing of a graph into strongly connected components can be done with a number of arithmetic operations proportional to the number of arcs in the graph, that is, the number of nonzero entries of the associated matrix. One such algorithm uses the method of depth-first search [see e.g., Aho et al. (1974)]. As we will see in the next section, Theorem 2.1.3 has an important consequence on the eigenvalue structure of a nonnegative matrix.

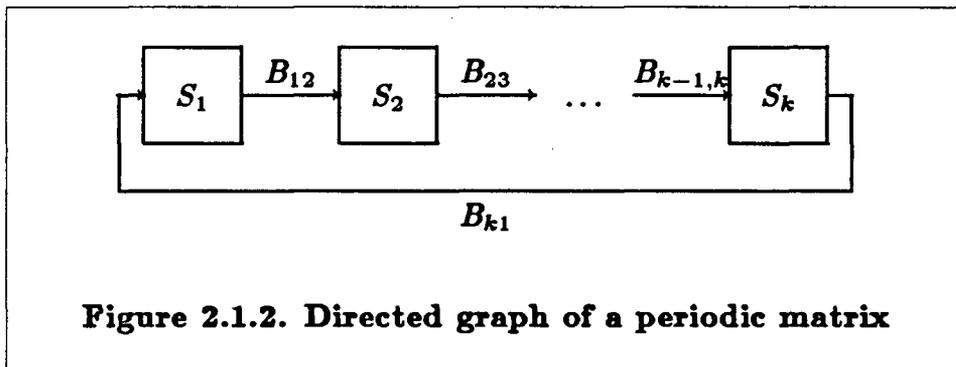
Also of importance is the periodic (or cyclic) structure of a graph and hence, of a matrix. Suppose  $A$  is an irreducible, complex  $n \times n$  matrix with directed graph  $G(A)$ .

Then  $G(A)$ , and  $A$  itself, is said to be *periodic with period  $k$* , with  $2 \leq k \leq n$ , if the set  $N$  of its nodes can be partitioned into  $k$  subsets  $S_1, \dots, S_k$  such that all arcs go from  $S_i$  to  $S_{i+1}$ , for  $i = 1, \dots, k-1$ , or from  $S_k$  to  $S_1$ . That is, suppose there is an arc from node  $r$  to node  $s$ . Then either  $r \in S_i$  and  $s \in S_{i+1}$ , if  $1 \leq i \leq k-1$ , or else  $r \in S_k$  and  $s \in S_1$ . The subsets  $S_1, \dots, S_k$  are called *periodic classes*. If no such partition of  $N$  exists, then  $G(A)$  is said to be *aperiodic*.

**Lemma 2.1.4.** An irreducible, complex  $n \times n$  matrix is periodic with period  $k$  iff  $k > 2$  is the largest integer for which there is a permutation matrix  $T$  such that  $A = TBT^{-1}$ , with

$$B = \begin{pmatrix} 0 & B_{12} & 0 & \dots & 0 \\ 0 & 0 & B_{23} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & B_{k-1,k} \\ B_{k1} & 0 & \dots & 0 & 0 \end{pmatrix}. \quad (2.1.4)$$

**Proof.** Because by definition, the directed graph  $G(A)$  has the structure shown in Figure 2.1.2 when  $A$  is periodic with period  $k$ . In this representation, a "node"  $S_i$  denotes the set of nodes in  $S_i$ , and an "arc"  $B_{ij}$  denotes the set of arcs going from nodes in  $S_i$  to nodes in  $S_j$ . There are no arcs going from nodes in  $S_i$  to other nodes in the same set  $S_i$ . □



**Figure 2.1.2. Directed graph of a periodic matrix**

Another definition of the period  $k$  of a graph is as the greatest common divisor of the lengths of all the paths going from any node to itself. We feel that the above

definition, in terms of periodic classes, is more immediate. Moreover, it will be used directly in the queueing application of Chapter VI.

The terminology we use is worth some comments. We refer to Aho et al. (1974) for the *strongly connected components* of a graph and their computation. Varga (1962) defined a matrix as either *irreducible* or *reducible*, and observed that an irreducible matrix has a *strongly connected* graph.

When the matrix represents the transition probabilities of a Markov chain, Karlin (1969) used the terms *transient* and *recurrent* to describe the *irreducible* subchains. Kemeny and Snell (1960) also used the term *ergodic* to denote a recurrent set.

Karlin (1969) also used the terms *periodic* and *aperiodic* for an irreducible Markov chain. Kemeny and Snell (1960) used the terms *cyclic* and *regular*, respectively. The terms *cyclic* and *primitive* were used by Varga (1962) when the matrix is nonnegative. The *period* is then called the *cyclic index* of a periodic matrix. For a complex, not necessarily nonnegative matrix, Varga (1962) also used the term *weakly cyclic*.

## 2.2. Eigenvalue structure of a stochastic matrix

A real  $n \times n$  matrix  $A$  is said to be *nonnegative* if  $a_{ij} \geq 0$ , and *positive* if  $a_{ij} > 0$ , for  $i, j = 1, \dots, n$ . A nonnegative  $n \times n$  matrix  $P$  is said to be *stochastic* if  $\sum_{j=1}^n p_{ij} = 1$ , for  $i = 1, \dots, n$ . Moreover, a nonnegative, irreducible  $n \times n$  matrix  $Q$  is said to be *substochastic* if  $\sum_{j=1}^n q_{ij} \leq 1$ , for  $i = 1, \dots, n$ , with strict inequality for at least one row.

Suppose a stochastic matrix  $P$  is reducible, with  $\ell$  transient classes and  $m$  recurrent classes. Then the transient submatrices  $B_{11}, \dots, B_{\ell\ell}$  of Theorem 2.1.3 are substochastic, while the recurrent submatrices  $D_{11}, \dots, D_{mm}$  are stochastic.

We will derive some well known results about the eigenvalue structure of an irreducible, nonnegative matrix, and use them to obtain the eigenvalue structure of a general stochastic matrix. Recall that the *spectral radius* of a matrix  $A$ , denoted  $\sigma(A)$ ,

is defined as  $\sigma(A) = |\lambda|$ , where  $\lambda$  is a dominant eigenvalue of  $A$ . That is, if  $\gamma$  is an eigenvalue of  $A$ , then  $|\gamma| \leq \sigma(A)$ .

We begin by discussing a few basic results from Varga (1962), concerning the Perron-Frobenius theory of irreducible nonnegative matrices. Suppose  $A$  is an irreducible, nonnegative  $n \times n$  matrix and let  $x \geq 0$  be a nonzero vector. We define the quantity

$$r_x = \min_{i: x_i > 0} \left\{ \sum_{j=1}^n a_{ij} x_j / x_i \right\}. \quad (2.2.1)$$

Then  $r_x = \sup\{\rho : Ax \geq \rho x, \rho \geq 0\}$ , and we consider the quantity

$$r = \sup\{r_x : x \geq 0, x \neq 0\}. \quad (2.2.2)$$

**Lemma 2.2.1.** Suppose  $A$  is an irreducible, nonnegative  $n \times n$  matrix. Then  $r > 0$ , and there is a positive eigenvector  $x$  such that  $Ax = rx$ .

**Lemma 2.2.2.** Suppose  $A$  is an irreducible, nonnegative  $n \times n$  matrix. Then  $\sigma(A) = r$ , i.e., if  $\lambda$  is an eigenvalue of  $A$ , then  $|\lambda| \leq r$ .

**Lemma 2.2.3.** Suppose  $A$  is an irreducible, nonnegative  $n \times n$  matrix. If  $x$  and  $y$  are such that  $Ax = rx$  and  $Ay = ry$ , then  $x$  and  $y$  are linearly dependent.

**Theorem 2.2.4.** Suppose  $A$  is an irreducible, nonnegative  $n \times n$  matrix. Then there is a nonsingular matrix  $S$  such that

$$A = S \begin{pmatrix} B & 0 \\ 0 & r \end{pmatrix} S^{-1}, \quad (2.2.3)$$

where  $B$  is an  $(n-1) \times (n-1)$  matrix such that  $\sigma(B) \leq r$ , and  $r$  is not an eigenvalue of  $B$ .

**Proof.** Such matrices  $B$  and  $S$  are defined by Theorem 1.1.10. That  $\sigma(B) \leq r$  follows from Lemma 2.2.2, and Lemma 2.2.3 implies that there is exactly one Jordan block corresponding to the eigenvalue  $r$ . We have to show that it has dimension 1.

We do this as in Section 1.1, by showing that Equation (1.1.6) has no solution. Suppose  $x > 0$  is the eigenvector of  $A$  of Lemma 2.2.1. We need to show that  $(rI - A)y =$

$x$  has no solution  $y$ . Let  $w > 0$  be a left-hand eigenvector of  $A$ , such that  $w(rI - A) = 0$ . Such a vector exists by Lemma 2.2.1, because  $w'$  is an eigenvector of  $A'$ , which is also nonnegative. Hence we have

$$0 = w(rI - A)y = wx > 0,$$

so that the system is not compatible. □

**Lemma 2.2.5.** Suppose  $P$  is an irreducible stochastic  $n \times n$  matrix. Then  $r = 1$ .

**Proof.** That 1 is an eigenvalue of  $P$  follows from the fact that  $Pe = e$ , where  $e$  is a vector with all entries equal to 1. Now suppose  $\lambda$  is any eigenvalue of  $P$ , and let  $x$  be an eigenvector associated with  $\lambda$ . Let  $k$  be such that  $|x_k| \geq |x_i|$ , for  $i = 1, \dots, n$ . Then  $Px = \lambda x$  implies that

$$|\lambda| \cdot |x_k| \leq \sum_{j=1}^n p_{kj} |x_j| \leq \left( \sum_{j=1}^n p_{kj} \right) |x_k| = |x_k|,$$

so that  $|\lambda| \leq 1$ . Hence  $r = \sigma(P) = 1$ . □

**Lemma 2.2.6.** Suppose  $Q$  is an irreducible substochastic  $n \times n$  matrix. Then  $r < 1$ .

**Proof.** Suppose  $\lambda$  is an eigenvalue of  $Q$ , with eigenvector  $x$ . Suppose also that the rows and columns of  $Q$  have been permuted in such a way that

$$|x_1| = \dots = |x_k| > |x_{k+1}| \geq \dots \geq |x_n|.$$

If  $k < n$ , there is at least one  $i \leq k$  for which  $q_{ij} > 0$  for some  $j > k$ , because  $Q$  is irreducible. Then  $Qx = \lambda x$  implies

$$\begin{aligned} |\lambda| \cdot |x_i| &\leq \sum_{j=1}^n q_{ij} |x_j| = \sum_{j=1}^k q_{ij} |x_j| + \sum_{j=k+1}^n q_{ij} |x_j| \\ &< \left( \sum_{j=1}^n q_{ij} \right) |x_i| \leq |x_i|, \end{aligned}$$

so that  $|\lambda| < 1$ .

If  $k = n$ , there is at least one  $i \leq n$  such that  $\sum_{j=1}^n q_{ij} < 1$ . Then  $Qx = \lambda x$  implies

$$|\lambda| \cdot |x_i| \leq \sum_{j=1}^n q_{ij} |x_j| = \left( \sum_{j=1}^n q_{ij} \right) |x_i| < |x_i|,$$

so that  $|\lambda| < 1$  as well. □

**Theorem 2.2.7.** Suppose  $P$  is a stochastic  $n \times n$  matrix with  $m$  recurrent classes. Then there is a nonsingular  $n \times n$  matrix  $S$  such that

$$P = S \begin{pmatrix} Q & 0 \\ 0 & I \end{pmatrix} S^{-1}, \quad (2.2.4)$$

where  $I$  is the  $m \times m$  identity matrix and  $Q$  is an  $(n - m) \times (n - m)$  matrix such that  $\sigma(Q) \leq 1$  and 1 is not an eigenvalue of  $Q$ .

**Proof.** By Theorem 2.1.3, every connected class corresponds to an irreducible submatrix. The eigenvalues of the transient classes all have modulus less than 1, by Lemma 2.2.6, and each of the  $m$  recurrent classes has exactly one Jordan block  $J_1(1)$ , by Theorem 2.2.4 and Lemma 2.2.5. Hence  $P$  is similar to the matrix of Equation (2.2.4).  $\square$

Finally, we will need some further results which refine Theorems 2.2.4 and 2.2.7.

**Lemma 2.2.8.** Suppose  $A$  is an aperiodic, irreducible, nonnegative  $n \times n$  matrix, and suppose  $r = \sigma(A)$ . If  $\lambda$  is an eigenvalue of  $A$  then either  $\lambda = r$  or  $|\lambda| < r$ .

**Theorem 2.2.9.** Suppose  $A$  is a periodic, irreducible nonnegative  $n \times n$  matrix with period  $k \geq 2$ , and suppose  $r = \sigma(A)$ . Then  $A$  has exactly  $k$  eigenvalues of modulus  $r$ , such that  $\lambda_j = r \exp(i\pi 2j/k)$ , for  $j = 1, \dots, k$ , each with multiplicity 1. Moreover, if  $\lambda$  is any other eigenvalue of  $A$ , then  $|\lambda| < 1$ .

**Proof.** Let  $T$  be the permutation matrix of Lemma 2.1.4, and let  $A = TBT^{-1}$  as in Equation (2.1.4). By Lemma 2.2.1,  $r$  is an eigenvalue of  $B$  and there is a positive vector  $x$  such that  $Bx = rx$ . Partition  $x$  as in Equation (2.1.4), so that

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix}.$$

It is straightforward to verify that  $\lambda_j = r \exp(i\pi 2j/k)$  is an eigenvalue of  $B$  with eigenvector

$$x^{(j)} = \begin{pmatrix} \exp(i\pi 2j/k)x_1 \\ \exp(i\pi 4j/k)x_2 \\ \dots \\ \exp(i\pi 2kj/k)x_k \end{pmatrix},$$

for  $j = 1, \dots, k$ .

We will see that each of these  $k$  eigenvalues  $\lambda_j$  has multiplicity 1, as follows. Consider the matrix  $B$  of Equation (2.1.4). Then  $B^k = \text{diag}(C_j)$ , with  $j = 1, \dots, k$ , where

$$C_1 = B_{12}B_{23} \dots B_{k1} \quad \dots \quad C_k = B_{k1}B_{12} \dots B_{k-1,k}.$$

Now each  $C_j$ , for  $j = 1, \dots, k$ , is an aperiodic, irreducible nonnegative matrix. By Lemma 2.2.8,  $\lambda$  is an eigenvalue of  $C_j$  implies that  $\lambda = \sigma(C_j)$  or else  $|\lambda| < \sigma(C_j)$ . Now if  $\gamma$  is an eigenvalue of  $B$ , then  $\gamma^k$  is an eigenvalue of  $B^k$ . This implies that  $B^k$  must have  $r^k$  as an eigenvalue, with multiplicity at least  $k$ , and that  $|\lambda| \leq r^k$ . We conclude that the multiplicity of  $r^k$  is exactly  $k$ , because each of the  $k$  blocks  $C_j$  has exactly one eigenvalue  $\lambda = r^k$ .  $\square$

**Corollary 2.2.10.** Suppose  $P$  is a stochastic matrix such that every recurrent class is aperiodic. Then the matrix  $Q$  of Theorem 2.2.7 has  $\sigma(Q) < 1$ .

### 2.3. Limiting and fundamental matrices

In the context of Markov chains and Markov decision processes, Theorem 2.2.7 has some important consequences. Let  $P$  be the transition matrix of a Markov chain with  $n$  states. Then  $P$  is a stochastic  $n \times n$  matrix, hence it satisfies Equation (2.2.4). We use this fact to provide an alternate proof of the following result of Kemeny and Snell (1960) that was fundamental in Blackwell (1962). Our approach is also different from that of Campbell and Meyer (1979), because our argument is entirely based on the decomposition property.

Consider a stochastic  $n \times n$  matrix  $P$  with  $m$  recurrent classes. By Theorem 2.1.3, we can write

$$P = T \begin{pmatrix} P_{00} & P_{01} & P_{02} & \dots & P_{0m} \\ 0 & P_{11} & 0 & \dots & 0 \\ \vdots & \ddots & P_{22} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & P_{mm} \end{pmatrix} T^{-1},$$

where  $T$  is a permutation matrix. Here, the matrices  $B$  and  $C$  of Equation (2.1.2) are given by

$$B = P_{00} \quad \text{and} \quad C = (P_{01} \quad \dots \quad P_{0m}),$$

and the matrix  $D$  of Equation (2.1.3) is such that  $D_{kk} = P_{kk}$ , for  $k = 1, \dots, m$ . In the following theorem, a matrix  $P^*$  is defined. Then the same permutation  $T$  is applied to the matrix  $P^*$ , giving the blocks  $P_{ij}^*$ , for  $i, j = 1, \dots, m$ .

**Theorem 2.3.1.** If  $P$  is a stochastic matrix with  $m$  recurrent classes, then there is a unique matrix  $P^*$  such that

$$PP^* = P^*P = P^*P^* = P^* \tag{2.3.1}$$

and  $P_{kk}^* > 0$  for every recurrent class, for  $k = 1, \dots, m$ . Moreover

$$P^* = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S \tag{2.3.2}$$

and  $Z = (I - P + P^*)^{-1}$  exists.

**Proof.** Decompose  $P$  as in equation (2.2.4), and apply the same similarity transformation  $S$  to  $P^*$ , to obtain

$$P^* = S^{-1} \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} S$$

where  $R_{11}$  is  $(n - m) \times (n - m)$ ,  $R_{22}$  is  $m \times m$  and  $R_{12}$  and  $R_{21}$  have the appropriate dimensions. If  $m = n$ , then  $P = I = P^*$  and we are done. Otherwise, using (2.2.4) and (2.3.1), we have (because  $B = I - Q$  is non-singular):

$$\begin{aligned} PP^* = P^* &\implies QR_{11} = R_{11} &\implies R_{11} = 0 \\ &QR_{12} = R_{12} &\implies R_{12} = 0 \\ P^*P = P^* &\implies R_{21}Q = R_{21} &\implies R_{21} = 0 \\ P^*P^* = P^* &\implies R_{22}R_{22} = R_{22} &\implies R_{22} \text{ is a projection.} \end{aligned}$$

Now  $P_{kk}^* > 0$  for all recurrent classes implies that  $\text{rank}(P^*) = m$ . This implies that  $\text{rank}(R_{22}) = m$ , and hence  $R_{22} = I$  since the only projection of full rank is the identity. So we have (2.3.2). Also, it trivially follows that

$$I - P + P^* = S^{-1} \begin{pmatrix} B & 0 \\ 0 & I \end{pmatrix} S \tag{2.3.3}$$

is indeed non-singular. □

We recall that  $P^*$  is called the *limiting matrix* of  $P$ , while  $Z$  is called the *fundamental matrix* [Kemeny and Snell (1960)]. Let  $A = I - P$ . By Theorem 2.2.7, the matrix  $A$  satisfies Equation (1.3.1) with  $B = I - Q$ , and hence its group inverse  $A^\#$  exists and is given by Equation (1.3.2). Then  $A^\#$  is the matrix  $H$  of Blackwell (1962), who derived the three following properties using a partial Laurent expansion of the resolvent of  $P$  (see Chapter III). They follow immediately from our representation.

**Corollary 2.3.2.** Let  $A^\#$  and  $P^*$  be defined as above. Then

$$(i) \quad (I - P)A^\# = A^\#(I - P) = I - P^*,$$

$$(ii) \quad A^\#P^* = P^*A^\# = 0 \quad \text{and}$$

$$(iii) \quad (I - P^*)A^\# = A^\#(I - P^*) = A^\#.$$

**Proof.** We show only (i), the rest follow in a similar manner and are omitted.

$$\begin{aligned} (I - P)A^\# &= S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} S S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} S = S^{-1} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} S \\ &= S^{-1} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} S - S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S = I - P^*. \end{aligned}$$

□

$A^\#$  is also called the *deviation matrix* by Veinott (1974), while  $P^*$  is called the *eigenprojection of  $A$  at  $\lambda = 1$* , by Rothblum (1981). The following representations for  $P^*$  and  $Z$  are useful and are easily derived using (2.2.4) and (2.3.2).

**Corollary 2.3.3.**

$$P^* = I - AA^\#$$

and

$$Z = (A + P^*)^{-1} = A^\# + P^*.$$

**Proof.** We have

$$\begin{aligned} I - AA^\# &= S^{-1} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} S - S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} S \\ &= S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S = P^* \end{aligned}$$

by (2.3.2), and

$$A^\# + P^* = S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & I \end{pmatrix} S = (A + P^*)^{-1}$$

by (2.3.3). □

**Corollary 2.3.4.**  $ZP^* = P^*$ .

**Proof.**

$$\begin{aligned} ZP^* &= S^{-1} \begin{pmatrix} B^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S \\ &= S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S = P^* \end{aligned}$$

□

The probabilistic interpretation of the limiting and deviation matrices is obtained using Cesàro limits. We give a new derivation of some results of Veinott (1974).

Consider a sequence  $\{A_i : i = 0, 1, 2, \dots\}$  of  $n \times n$  matrices, and let

$$B_N = \sum_{i=0}^{N-1} A_i, \quad \text{for } N = 0, 1, 2, \dots$$

If  $\lim_{N \rightarrow \infty} N^{-1} B_N = A$  exists, then  $A$  is called the *Cesàro limit of order one* of  $\{A_i\}$ , and we write

$$\lim_{N \rightarrow \infty} A_N = A (C, 1).$$

Further, if  $\lim_{N \rightarrow \infty} B_N = B (C, 1)$ , then we write

$$\sum_{N=0}^{\infty} A_N = B (C, 1).$$

We investigate the limits of sums of powers of a matrix.

**Lemma 2.3.5.** Suppose  $I - Q$  is nonsingular. Then

$$\sum_{i=0}^N Q^i = (I - Q)^{-1} (I - Q^{N+1}), \quad \text{for } N \geq 0.$$

**Proof.** Because

$$(I - Q) \sum_{i=0}^N Q^i = I - Q^{N+1}. \quad (2.3.4)$$

□

**Lemma 2.3.6.** Suppose  $\lim_{N \rightarrow \infty} Q^N = 0$ . Then  $(I - Q)$  is nonsingular and

$$(I - Q)^{-1} = \sum_{i=0}^{\infty} Q^i. \quad (2.3.5)$$

**Proof.** For  $N$  large enough, the matrix  $I - Q^{N+1}$  of Equation (2.3.4) is nonsingular because the entries of  $Q^{N+1}$  are small. This implies that  $I - Q$  is nonsingular. The result follows by taking limits.  $\square$

**Lemma 2.3.7.** Suppose  $Q^N$  is bounded, for  $N = 0, 1, 2, \dots$ , and  $I - Q$  is nonsingular. Then

$$\lim_{N \rightarrow \infty} Q^N = 0 \quad (C, 1).$$

**Proof.** By Lemma 2.3.5, we have

$$\lim_{N \rightarrow \infty} N^{-1} \sum_{i=0}^{N-1} Q^i = \lim_{N \rightarrow \infty} N^{-1} (I - Q)^{-1} (I - Q^N).$$

The result follows because  $Q^N$  is bounded.  $\square$

**Lemma 2.3.8.** Suppose  $Q^N$  is bounded, for  $N = 0, 1, 2, \dots$ , and  $I - Q$  is nonsingular. Then

$$\sum_{N=0}^{\infty} Q^N = (I - Q)^{-1} \quad (C, 1).$$

**Proof.** Let

$$X = \lim_{N \rightarrow \infty} N^{-1} \sum_{i=0}^{N-1} \sum_{j=0}^i Q^j.$$

By Lemma 2.3.5, we have

$$X = \lim_{N \rightarrow \infty} N^{-1} \sum_{i=0}^{N-1} (I - Q)^{-1} (I - Q^{i+1}).$$

Define

$$Y = \lim_{N \rightarrow \infty} N^{-1} \sum_{i=0}^{N-1} I \quad \text{and} \quad Z = \lim_{N \rightarrow \infty} N^{-1} \sum_{i=0}^{N-1} Q^i.$$

Then  $Y = I$ , and Lemma 2.3.7 implies that  $Z = 0$ . Hence

$$X = (I - Q)^{-1} (Y - QZ) = (I - Q)^{-1},$$

and the result follows.  $\square$

**Theorem 2.3.9.** Suppose  $P$  is a stochastic matrix, and let  $P^*$  be defined as in Theorem 2.3.1. Then

$$\lim_{N \rightarrow \infty} P^N = P^* (C, 1).$$

Further, if  $P$  is aperiodic, then the ordinary limit can be used instead of  $(C, 1)$ .

**Proof.** Decompose  $P$  as in Equation (2.2.4). Then

$$P^N = S^{-1} \begin{pmatrix} Q^N & 0 \\ 0 & I \end{pmatrix} S. \quad (2.3.6)$$

Since  $P^N$  is a stochastic matrix, we have that  $P^N$  is bounded, and hence so is  $Q^N$ . Moreover,  $I - Q$  is nonsingular, by Theorem 2.2.7. The result follows from Lemma 2.3.7 and Equation (2.3.2).

If  $P$  is aperiodic, then  $\sigma(Q) < 1$ , by Corollary 2.2.10. This implies that

$$\lim_{N \rightarrow \infty} Q^N = 0,$$

by Theorem 2.4.2. □

**Theorem 2.3.10.** Suppose  $P$  is a stochastic matrix, and let  $A = I - P$ . Then

$$\sum_{N=0}^{\infty} (P^N - P^*) = A^{\#} (C, 1).$$

Further, if  $P$  is aperiodic, then the ordinary limit can be used instead of  $(C, 1)$ .

**Proof.** Using Equations (2.2.4) and (2.3.2), we have

$$P^N - P^* = S^{-1} \begin{pmatrix} Q^N & 0 \\ 0 & 0 \end{pmatrix} S. \quad (2.3.7)$$

The result follows from Lemma 2.3.8, with  $A^{\#}$  defined by Equation (1.3.2).

If  $P$  is aperiodic, the result follows from Lemma 2.3.6. □

The consequence of Theorem 2.3.9 is that, for a Markov chain with stochastic transition matrix  $P$ , the entry  $\pi_{ij}$  of the limiting matrix  $P^*$  is the expected fraction of the time spent in state  $j$ , given that the system started in state  $i$ .

By Theorem 2.3.10, the entry  $\alpha_{ij}$  of the deviation matrix  $A^\#$  is the expected difference between the expected number of visits to state  $j$  when the system starts in state  $i$ , and the expected number of visits to state  $j$  when the system starts with the stationary distribution.

We conclude this section with a simple numerical example, provided by the transition matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which has no transient states and only one recurrent class (with period 2). We have that

$$P^* = \begin{pmatrix} 1 & 1/2 \\ -1 & 1/2 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & -1/2 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

and

$$Z = A^\# + P^* = \begin{pmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}.$$

## 2.4. Geometric convergence in the aperiodic case

In the previous section, we expressed  $P^*$  and  $A^\#$  as Cesàro limits using sequences of powers of the transition matrix  $P$ . The rate of convergence of those limits is only harmonic, however, as the proofs of Theorems 2.3.9 and 2.3.10 indicate.

When the transition matrix  $P$  is aperiodic, ordinary limits converge. In this section, we show that the rate of convergence is geometric in this case.

Consider the limiting matrix  $P^*$ . By Theorem 2.3.9, if  $P$  is aperiodic then

$$\lim_{N \rightarrow \infty} P^N = P^*.$$

Define the error matrix  $E^{(N)}$  of the  $N$ -th power as

$$E^{(N)} = P^N - P^*. \tag{2.4.1}$$

From Equation (2.3.7), we have

$$E^{(N)} = S^{-1} \begin{pmatrix} Q^N & 0 \\ 0 & 0 \end{pmatrix} S, \tag{2.4.2}$$

where  $Q$  is as in Equation (2.2.4). Specifically, choose  $Q$  to be the Jordan canonical form, i.e.,

$$Q = \text{diag}(J_{n_i}(\lambda_i)), \text{ with } |\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_k|,$$

where  $k$  is the number of Jordan blocks. Let  $\gamma = |\lambda_1|$ . By Corollary 2.2.10, we have  $\gamma < 1$  for an aperiodic transition matrix  $P$ .

**Lemma 2.4.1.** [See e.g., Varga (1962), Equation (1.28)]. Suppose  $d_{ij}^{(N)}$  is the  $(i, j)$ -th entry of the matrix  $(J_n(\lambda))^{(N)}$ . Then

$$d_{ij}^{(N)} = \begin{cases} 0 & \text{if } i < j \\ \binom{N}{i-j} \lambda^{N-i+j} & \text{if } j \leq i \leq \min(n, N+j) \\ 0 & \text{if } N+j < i \leq n \end{cases}$$

where

$$\binom{N}{k} = \frac{N!}{k!(N-k)!}.$$

**Theorem 2.4.2.** Suppose  $\sigma(Q) < 1$ . Then  $\lim_{N \rightarrow \infty} Q^N = 0$ .

**Proof.** Because

$$Q^N = \text{diag}\left(\left(J_{n_i}(\lambda_i)\right)^N\right)$$

and, by Lemma 2.4.1, we have

$$\lim_{N \rightarrow \infty} \left(J_{n_i}(\lambda_i)\right)^N = 0$$

when  $|\lambda_i| < 1$ . □

**Corollary 2.4.3.** Suppose  $P$  is an aperiodic stochastic matrix. Then

$$\lim_{N \rightarrow \infty} E^{(N)} = 0.$$

**Proof.** By applying the theorem to Equation (2.4.2). □

The geometric rate of convergence is a direct consequence of Lemma 2.4.1. The idea is more precisely expressed using norms. Following Varga (1962), Section 1.3, we define the *Euclidean norm* of a real vector  $x$  as

$$\|x\|_2 = (x'x)^{1/2}.$$

Moreover, the *spectral norm* of a real matrix  $A$  is defined as

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Also, by Theorem 1.3 of Varga (1962), we have

$$\|A\|_2 = (\sigma(A'A))^{1/2}.$$

The following result is an immediate consequence of Theorem 3.1 of Varga (1962), which is based on Lemma 2.4.1.

**Theorem 2.4.4.** Suppose  $P$  is an aperiodic stochastic matrix, and let  $\gamma = \sigma(Q)$ . Then

$$\lim_{N \rightarrow \infty} \frac{\|E^{(N+1)}\|_2}{\|E^{(N)}\|_2} = \gamma.$$

This result implies that for large  $N$ , every power of  $P$  reduces the error by a factor  $\gamma$ , the subdominant eigenvalue of  $P$ . Now suppose  $P$  is irreducible. By Lemma 2.2.3, there is a unique row probability vector  $\pi$  such that  $\pi P = \pi$ . Then  $\pi$  is the *stationary distribution* of  $P$ , and  $\pi P^* = \pi$ .

**Corollary 2.4.5.** Suppose  $P$  is an aperiodic, irreducible stochastic matrix, and let  $\pi^{(0)}$  be an arbitrary probability vector. Then

$$\pi = \lim_{N \rightarrow \infty} \pi^{(0)} P^N$$

is the stationary distribution of  $P$ .

**Proof.** Let  $\pi^{(N)} = \pi^{(0)} P^N$ . By Theorem 2.3.9, the limit exists when  $P$  is aperiodic, and is given by  $\pi = \pi^{(0)} P^*$ . Since  $P^* P = P^*$ , we have  $\pi P = \pi$ , so that  $\pi$  is the unique eigenvector of  $P$  of Lemma 2.2.3, with eigenvalue 1. Hence  $\pi$  is the stationary distribution of  $P$ .  $\square$

**Lemma 2.4.6.** Let  $E^{(N)} = \pi^{(N)} - \pi$  be the vector of errors on the  $N$ -th iterate  $\pi^{(N)}$ .

Then

$$e^{(N)} = e^{(0)} E^{(N)}, \tag{2.4.3}$$

where  $e^{(0)} = \pi^{(0)} - \pi$  is the initial error.

**Proof.** Since  $\pi = \pi^{(0)}P^*$ , we have

$$e^{(N)} = \pi^{(N)} - \pi = \pi^{(0)}(P^N - P^*).$$

Now  $\pi P^N = \pi$  implies that  $\pi(P^N - P^*) = 0$ , so we get

$$\begin{aligned} e^{(N)} &= \pi^{(0)}(P^N - P^*) - \pi(P^N - P^*) \\ &= (\pi^{(0)} - \pi)(P^N - P^*) = e^{(0)}E^{(N)}, \end{aligned}$$

by Equation (2.4.1). □

To show that the rate of convergence of  $e^{(N)}$  is geometric, we use a measure  $\epsilon^{(N)}$  of the error, such that  $\|e^{(N)}\|_2 \leq \epsilon^{(N)}$ . We choose

$$\epsilon^{(N)} = \|e^{(0)}\|_2 \cdot \|E^{(N)}\|_2. \tag{2.4.4}$$

**Corollary 2.4.7.** The error  $\epsilon^{(N)}$  converges to zero geometrically. More precisely,

$$\lim_{N \rightarrow \infty} \frac{\epsilon^{(N+1)}}{\epsilon^{(N)}} = \gamma,$$

where  $\gamma < 1$  is the modulus of the subdominant eigenvalue of  $P$ .

**Proof.** By Theorem 2.4.4. □

The above results will be extended, in Chapter VI, to the case of a periodic Markov chain. Also, the rate of convergence for the deviation matrix  $A^\#$  will be investigated in Chapter III, in the aperiodic case, when solving the evaluation equations of dynamic programming.

## Chapter III

### Discrete time Markov decision processes

In this chapter, we use the decomposition results of the previous chapters to provide new, simple and direct derivations of many important results of dynamic programming. Also, some new algorithms are described, to compute the gain, the bias and the higher order terms required in the policy improvement methods of dynamic programming.

With a given stationary policy  $\delta$ , a Markov decision process evolves according to a Markov chain with stochastic transition matrix  $P^\delta$ . Hence the singular matrix  $A^\delta = I - P^\delta$  can be decomposed as in Equation (1.3.1).

In Section 3.1, the decomposition of  $A^\delta$  is used to give a direct derivation of the Laurent expansion of the resolvent [see Veinott (1969) and (1974)], and of the discounted value function, when the interest rate is small. The boundedness of the powers of the transition matrix  $P^\delta$  is not used in this derivation. This property is crucial, however, for the probabilistic interpretation of the terms of the series expansion, especially the gain and the bias terms.

Then in Section 3.2, we formulate the problem of finding the terms of a truncated series as a finite system of linear equations. Using matrix decomposition again, we show that the packed vector (of the terms of the truncated series) can be solved using the Drazin inverse of the packed matrix of coefficients.

Next, the computation of the terms of the series expansion is considered in Section 3.3. We show how the matrix decomposition algorithms of Chapter I can be used to compute the gain, the bias and the higher order terms.

In Section 3.4, we show that when the transition matrix is aperiodic, the gain and the bias can be computed iteratively using the power method, and that the convergence is geometric. We also show that the error on the approximate gain and bias is related to the residual error by a system of equations of the same type as the evaluation equations.

Finally, the application of the Drazin inverse approach to policy improvement algorithms is explored in Section 3.4. We provide a new proof of the result of Veinott (1974) that at most  $n - m$  terms are needed to compare the value of two policies, with respect to the discount optimality criterion, where  $n$  is the number of states and  $m$  is the number of recurrent classes.

### 3.1. Markov reward processes

Consider a homogeneous Markov chain  $\{X(t) : t = 0, 1, 2, \dots\}$  with finite state space  $N = \{1, \dots, n\}$  and stochastic transition matrix  $P$ , where

$$p_{ij} = \text{Prob}\{X(t+1) = j \mid X(t) = i\}, \quad i, j = 1, \dots, n.$$

Suppose also that at the end of every period  $t$ , the system earns a reward  $r_i$ , where  $i = X(t)$ . We say that  $r$  is the vector of *immediate rewards*. Such a Markov chain with rewards is called a *Markov reward process*, and is denoted by the triplet  $(N, r, P)$ .

Now suppose an arbitrary random variable  $Y$  is defined on the Markov chain  $\{X(t)\}$ . We define the conditional expectation operator  $E_i$  as

$$E_i(Y) = E\{Y \mid X(0) = i\}, \quad \text{for } i = 1, \dots, n.$$

Let  $\rho > 0$  be the interest rate for one period. Then  $\beta = (1 + \rho)^{-1}$  is the one period discount factor, with  $0 < \beta < 1$ . We are interested in the present value, at time  $t = 0$ , of the total reward earned over an infinite horizon. More precisely, we define the expected total discounted reward function  $v(\rho)$  such that

$$v_i(\rho) = E_i\left(\sum_{k=0}^{\infty} \beta^{k+1} r_{X(k)}\right), \quad i = 1, \dots, n.$$

The expectation can be expressed directly using matrix notation, and we get

$$v(\rho) = \beta \sum_{k=0}^{\infty} \beta^k P^k r = \beta(I - \beta P)^{-1} r. \quad (3.1.1)$$

The infinite series in Equation (3.1.1) converges by Lemma 2.3.6, because

$$\lim_{k \rightarrow \infty} \beta^k P^k = 0,$$

with  $0 < \beta < 1$ .

We are interested in the case when  $\beta$  is in the neighbourhood of 1. Following Veinott (1969) and (1974), we replace the discount factor  $\beta$  by the interest rate  $\rho$ , so that

$$\begin{aligned} v(\rho) &= (1 + \rho)^{-1} \{I - (1 + \rho)^{-1} P\}^{-1} r \\ &= \{(1 + \rho)I - P\}^{-1} r \\ &= \{\rho I + A\}^{-1} r \end{aligned} \quad (3.1.2)$$

where  $A = I - P$ . The matrix  $R_\rho = (\rho I + A)^{-1}$  is called the *resolvent* of  $A$ . We now provide a direct derivation of the Laurent expansion for the resolvent [Theorem 2 of Veinott (1969) and Theorem 3 of Veinott (1974)] using algebraic properties of the matrix  $A$  and its Drazin inverse  $A^\#$ .

**Theorem 3.1.1.** Let  $\lambda$  be the non-zero eigenvalue of  $A$  with smallest modulus. If  $0 < \rho < |\lambda|$  then

$$R_\rho = \rho^{-1} P^* + \sum_{i=0}^{\infty} (-\rho)^i (A^\#)^{i+1}. \quad (3.1.3)$$

**Proof.** By definition,

$$\begin{aligned} R_\rho &= (\rho I + A)^{-1} = S^{-1} \begin{pmatrix} \rho I + B & 0 \\ 0 & \rho I \end{pmatrix}^{-1} S \\ &= S^{-1} \begin{pmatrix} (\rho I + B)^{-1} & 0 \\ 0 & \rho^{-1} I \end{pmatrix} S \\ &= \rho^{-1} S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S + S^{-1} \begin{pmatrix} (\rho I + B)^{-1} & 0 \\ 0 & 0 \end{pmatrix} S. \end{aligned}$$

The first term is  $\rho^{-1}P^*$ . Now consider the second term.

$$\begin{aligned} [\rho I + B]^{-1} &= [B(\rho B^{-1} + I)]^{-1} \\ &= [I + \rho B^{-1}]^{-1} B^{-1}. \end{aligned}$$

By hypothesis,  $\sigma(\rho B^{-1}) = \rho/|\lambda| < 1$ , so that

$$[I + \rho B^{-1}]^{-1} = \sum_{i=0}^{\infty} (-\rho)^i (B^{-1})^i,$$

by Lemma 2.3.6. But

$$S^{-1} \begin{pmatrix} (B^{-1})^i & 0 \\ 0 & 0 \end{pmatrix} S = (A^\#)^i$$

so the second term above equals

$$\sum_{i=0}^{\infty} (-\rho)^i (A^\#)^{i+1}$$

which gives the desired result. □

**Corollary 3.1.2.** Let  $w_{-1}, w_0, w_1, \dots$ , be vectors such that

$$w_{-1} = P^* r \tag{3.1.4}$$

and

$$w_i = (-1)^i (A^\#)^{i+1} r, \quad i = 0, 1, 2, \dots \tag{3.1.5}$$

Then

$$v(\rho) = \rho^{-1} w_{-1} + w_0 + \sum_{i=1}^{\infty} \rho^i w_i. \tag{3.1.6}$$

**Proof.** By substituting (3.1.3) into Equation (3.1.2). □

This Laurent expansion of the discounted reward function  $v(\rho)$  is fundamental for the notion of discount optimality that will be discussed in Section 3.5. Observe that our derivation uses only the property that  $\text{ind}(A) = 1$ , that is, that the matrix  $A$  can be decomposed as in Equation (1.3.1).

A special case of Theorem 3.1.1 is when the matrix  $A$  is nonsingular [i.e.,  $\text{ind}(A) = 0$ ]. This is the case, for example, if all the recurrent subchains of  $P$  are substochastic. Then Equation (3.1.3) becomes

$$R_\rho = \sum_{i=0}^{\infty} (-\rho)^i (A^{-1})^{i+1}.$$

Theorem 3.1.1 is itself a special case of a result of Rothblum (1981) [his Theorem 3.1], in which a matrix  $A$  with  $\text{ind}(A) = k \geq 0$  has terms in  $\rho^{-i}$ , for  $i = 0, \dots, k$ .

When  $P$  is a stochastic matrix, the terms  $w_{-1}$  and  $w_0$  have a probabilistic interpretation of particular interest. Define the symbols

$$g = w_{-1} = P^* r \tag{3.1.7}$$

and

$$h = w_0 = A^\# r. \tag{3.1.8}$$

The term  $g$  is called the *gain* of the process and  $h$  is the *bias*.

**Theorem 3.1.3.** Suppose  $P$  is a stochastic matrix. Then

$$g = \lim_{k \rightarrow \infty} P^k r (C, 1)$$

and

$$h = \sum_{k=0}^{\infty} (P^k r - g) (C, 1).$$

Moreover, if  $P$  is aperiodic then the result is valid using ordinary limits.

**Proof.** By Theorems 2.3.9 and 2.3.10. □

Note that in addition to the fact that  $\text{ind}(A) = 1$ , this result uses the fact that the powers of  $P$  are bounded matrices. Theorem 3.1.3 suggests the following interpretation for the gain and bias terms. The entry  $g_i$  is the expected reward earned by the system per time period, in the long run, given that the system starts in state  $i$ . The entry  $h_i$ , on the other hand, is the total difference between the expected earnings in one period and the long run average earning, given that the system starts in state  $i$ , summed over all periods.

We conclude this section with a very simple derivation of an equation which is well known for computing the gain and the bias [see Howard (1960) and Blackwell (1962)]. This equation is usually derived using a partial series expansion, but here it is based on the generalized inverse representation for  $P^*$ .

**Proposition 3.1.4.** Let  $g$  and  $h$  be defined by (3.1.7) and (3.1.8). Then  $g$  and  $h$  satisfy

$$h = r - g + Ph. \quad (3.1.9)$$

**Proof.** We use the fact that  $P^* = I - AA^\#$ . By (3.1.7), we have

$$g = (I - AA^\#)r = r - A(A^\#r) = r - (I - P)h,$$

where the last equality follows from (3.1.8). The result follows by rearranging terms.  $\square$

### 3.2. Singular systems of equations

Let  $A$  be an  $n \times n$  matrix of index 1 [i.e., decomposition (1.3.1) is valid], and suppose  $b$  is a given column vector. Recall that the projection matrix  $W = I - AA^\#$  is given by Equation (1.2.8). In the special case  $A = I - P$  with  $P$  a (stochastic) transition matrix, we have  $W = P^*$ , the limiting matrix.

**Lemma 3.2.1.** The (singular) system of equations  $Ax = b$  has a solution iff  $Wb = 0$ . Further, if  $Wb = 0$ , then for arbitrary  $y$

$$x = A^\#b + Wy \quad (3.2.1)$$

is a solution of  $Ax = b$ .

**Proof.** Let

$$x = S^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, b = S^{-1} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \text{ and } y = S^{-1} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

Then

$$Ax = S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} SS^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = S^{-1} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

which is equivalent to

$$\begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

This system has a solution iff  $b_2 = 0$ , which is equivalent to  $Wb = 0$ , because

$$Wb = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} SS^{-1} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = S^{-1} \begin{pmatrix} 0 \\ b_2 \end{pmatrix}.$$

Now we assume that  $b_2 = 0$  and we take  $x = A^\#b + Wy$ , for some arbitrary vector  $y$ .

We have

$$\begin{aligned} Ax &= AA^\#b + AWy \\ &= S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ 0 \end{pmatrix} + S^{-1} \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ &= S^{-1} \begin{pmatrix} b_1 \\ 0 \end{pmatrix} + 0 = b. \end{aligned}$$

□

**Theorem 3.2.2.** Let  $\{b_i, i = -1, 0, 1, \dots\}$  be a sequence of  $n$ -vectors and  $W = I - AA^\#$ .

Then the system of equations

$$Ax_{-1} = b_{-1} \tag{3.2.2}$$

$$x_{i-1} + Ax_i = b_i \quad i = 0, 1, 2, \dots \tag{3.2.3}$$

has a solution iff

$$Wb_{-1} = 0. \tag{3.2.4}$$

Furthermore, if (3.2.4) holds, the solution is unique and is given by

$$x_i = A^\#(b_i - x_{i-1}) + Wb_{i+1}, \quad i = -1, 0, 1, 2, \dots \tag{3.2.5}$$

where  $x_{-2} = 0$ .

**Proof.** By Lemma 3.2.1, Equation (3.2.2) has a solution iff  $Wb_{-1} = 0$ . Now assume (3.2.2) has a solution. We show by induction on  $i$  that under (3.2.4), a solution exists and (3.2.5) gives its unique value. The induction hypothesis is that  $x_{k-1}$  is unique and that (3.2.3) has a solution for  $i = k$ , for some  $k \geq -1$ . This is trivially true for  $k = -1$  with  $x_{-2} = 0$ , because (3.2.2) has a solution, by hypothesis. Rewrite (3.2.3) with  $i=k$  as

$$Ax_k = b_k - x_{k-1}.$$

By Lemma 3.2.1, one solution is

$$x_k = A^\#(b_k - x_{k-1}) + Wy_k \tag{3.2.6}$$

where  $y_k$  is an arbitrary vector. Now apply Lemma 3.2.1 to Equation (3.2.3) with  $i = k + 1$ . It has a solution iff

$$W(b_{k+1} - x_k) = 0.$$

Substituting (3.2.6), we have

$$W(b_{k+1} - A^\#(b_k - x_{k-1}) - Wy_k) = 0.$$

But  $WA^\# = 0$  (see Corollary 2.3.2) and  $W^2 = W$ , so we must have

$$Wy_k = Wb_{k+1}$$

and hence  $x_k$  is uniquely determined by (3.2.5). □

We now apply this to the Markov reward process  $(N, r, P)$ .

**Corollary 3.2.3.** Let  $A = I - P$ . Then the terms  $g, h$  and  $w_i, i = 1, 2, \dots$ , of the series (3.1.6) are the unique solution of the recurrence

$$Ag = 0 \tag{3.2.7}$$

$$g + Ah = r \tag{3.2.8}$$

$$w_{i-1} + Aw_i = 0 \quad i = 1, 2, \dots \tag{3.2.9}$$

with  $g = w_{-1}$  and  $h = w_0$ .

**Proof.** Apply Theorem 3.2.2 with  $b_{-1} = 0, b_0 = r$  and  $b_i = 0$ , for  $i = 1, 2, \dots$ . Here  $W = P^*$  so we get  $g = P^*r, h = A^\#r$  and assuming  $w_{i-1} = (-1)^{i-1}(A^\#)^i r$ , Equation (3.2.6) gives  $w_i = (-1)^i(A^\#)^{i+1} r$  as in Equation (3.1.5). □

This last result unifies some previous approaches to the problem. Howard (1960) derived (3.2.7) and (3.2.8) using  $z$ -transforms and made  $h$  unique by setting one component to zero in each recurrent class. Blackwell (1962) derived both equations using the series (3.1.1) and some limiting argument as  $\beta \nearrow 1$ . He also used the extra conditions

$$P^*g = P^*r$$

and

$$P^*h = 0.$$

We saw that these conditions are necessary and sufficient for (3.2.8) and (3.2.9) with  $i = 1$  to have a solution, respectively. Finally, Veinott (1969) derived (3.2.7), (3.2.8) and (3.2.9) using the series expansion (3.1.6).

As will be seen in the next section, truncated recursions as in (3.2.2) and (3.2.3), but with a finite number of terms, are also of interest. They form a finite system of linear equations. Let  $A$  be an  $n \times n$  complex matrix such that  $\text{ind}(A) \leq 1$  (i.e.,  $A$  is either non-singular or else it satisfies equation (1.3.1)). For  $k \geq 1$ , we define the matrix  $A(k)$  as the block Jordan matrix of order  $k$  that has  $A$  for its diagonal blocks. For example,  $A(1) = A$  and for  $k = 4$ , we have

$$A(4) = \begin{pmatrix} A & 0 & 0 & 0 \\ I & A & 0 & 0 \\ 0 & I & A & 0 \\ 0 & 0 & I & A \end{pmatrix}.$$

We are interested in solving the system of equations:

$$A(k)x(k) = b(k) \tag{3.2.10}$$

where  $x(k)$  and  $b(k)$  are partitioned so that they be conformable with  $A$ . For example, with  $k = 4$  we have

$$x(4) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad \text{and} \quad b(4) = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix},$$

where  $x_1, \dots, x_4$  and  $b_1, \dots, b_4$  are  $n$ -vectors. This system can also be written recursively as

$$Ax_1 = b_1 \tag{3.2.11}$$

and

$$x_{i-1} + Ax_i = b_i, \quad i = 2, \dots, k. \tag{3.2.12}$$

If the matrix  $A$  is non-singular, then (3.2.11) would have the unique solution  $x = A^{-1}b_1$  and (3.2.12), which can be written as

$$Ax_i = b_i - x_{i-1}, \quad (3.2.13)$$

would have the unique solution

$$x_i = A^{-1}(b_i - x_{i-1}), \quad i = 2, \dots, k.$$

Expanding this iterative solution, we can easily verify that it can be expressed as

$$x(k) = A(k)^{-1}b(k)$$

where  $A(k)^{-1}$  is a block lower triangular matrix such that for  $i \geq j$ , block  $(i, j)$  is given by

$$(-1)^{i-j}(A^{-1})^{i-j+1}. \quad (3.2.14)$$

For example, with  $k = 4$  we have

$$A(4)^{-1} = \begin{pmatrix} A^{-1} & 0 & 0 & 0 \\ -A^{-2} & A^{-1} & 0 & 0 \\ A^{-3} & -A^{-2} & A^{-1} & 0 \\ -A^{-4} & A^{-3} & -A^{-2} & A^{-1} \end{pmatrix}.$$

We now extend these results to the case when  $A$  is singular. We assume that  $\text{ind}(A) = 1$ , so that  $A^\#$  exists, and we define  $W = I - AA^\#$  as before. We show below that an analogous representation for  $A(k)^D$  with  $A^\#$  replacing  $A^{-1}$  in (3.2.14) holds, but that it does not give a solution to the system of equations (3.2.10).

**Theorem 3.2.4.** Suppose  $\text{ind}(A) = 1$ . Then the system (3.2.10) has a solution iff  $Wb_1 = 0$ . In this case,  $x_1, \dots, x_{k-1}$  are uniquely determined and are given by

$$x_1 = A^\#b_1 + Wb_2 \quad (3.2.15)$$

and

$$x_i = A^\#(b_i - x_{i-1}) + Wb_{i+1}, \quad i = 2, \dots, k-1 \quad (3.2.16)$$

while the last component is

$$x_k = A^\#(b_k - x_{k-1}) + Wy \quad (3.2.17)$$

for an arbitrary vector  $y$ .

**Proof.** Apply Lemma 3.2.1 as in Theorem 3.2.2 to obtain existence and uniqueness up to  $i = k - 1$ . For  $i = k$ , there is no next equation to satisfy, so the vector  $y$  is arbitrary.  $\square$

This result is readily applied to the Laurent expansion for Markov reward processes. Let us take  $A = I - P$ , where  $P$  is, as before, the transition matrix of a Markov chain with immediate reward vector  $r$ .

**Corollary 3.2.5.** The terms  $w_j$ ,  $j = -1, 0, \dots, i - 1$  of the series (3.1.6) are uniquely determined by a system of  $i + 2$  equations.

**Proof.** To obtain the terms  $w_{-1}, w_0, \dots, w_i$ , let us define

$$x_{j+2} = w_j, \quad j = -1, 0, \dots, i,$$

and

$$b_{j+2} = \begin{cases} 0 & \text{if } j = -1 \\ r & \text{if } j = 0 \\ 0 & \text{if } j = 1, 2, \dots, i. \end{cases}$$

Using equations (3.2.7), (3.2.8) and (3.2.9), we form the system (3.2.10) with  $k=i+2$ .

For example, for  $i=2$ , we have  $k = 4$  and

$$\begin{pmatrix} A & 0 & 0 & 0 \\ I & A & 0 & 0 \\ 0 & I & A & 0 \\ 0 & 0 & I & A \end{pmatrix} \begin{pmatrix} w_{-1} \\ w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ r \\ 0 \\ 0 \end{pmatrix}.$$

Clearly, the first equation has a solution (because  $b_1 = 0$ ) and so Theorem 3.2.4 applies. Hence  $w_{-1}, \dots, w_{i-1}$  are uniquely determined and are given by (3.2.15) and (3.2.16). The last component  $w_i$  is not unique and is given by (3.2.17).  $\square$

We will now see how the solution of (3.2.10) can be expressed in terms of the Drazin inverse of the matrix  $A(k)$ , thus extending equation (3.2.14) to the singular case.

**Lemma 3.2.6.** If  $A$  is singular but satisfies (1.3.1), then  $A(k)$ ,  $k \geq 1$ , has index  $k$  and its Drazin inverse  $A(k)^D$  is a block lower triangular matrix such that its block  $(i, j)$ ,

$i \geq j$ , is given by

$$(-1)^{i-j}(A^\#)^{i-j+1}. \quad (3.2.18)$$

**Proof.** We use the fact that  $A$  can be decomposed as in (1.3.1), where  $B$  is non-singular. Let  $S(k)$  be the block diagonal matrix that has all diagonal blocks equal to the matrix  $S$  of Equation (1.3.1). We define a new matrix  $C(k)$  as

$$C(k) = S(k)A(k)S(k)^{-1}$$

so that

$$A(k) = S(k)^{-1}C(k)S(k).$$

We argue that there is a permutation matrix  $Q(k)$  such that

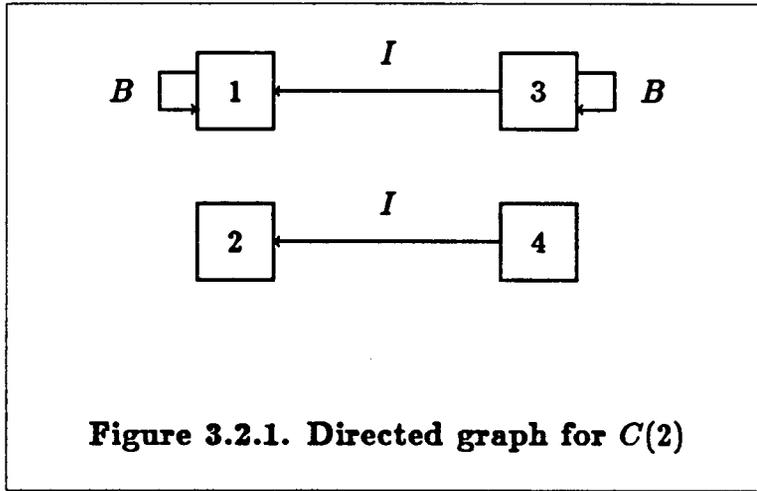
$$\begin{aligned} D(k) &= Q(k)C(k)Q(k)^{-1} \\ &= \begin{pmatrix} B(k) & 0 \\ 0 & J(k) \end{pmatrix} \end{aligned} \quad (3.2.19)$$

where  $B(k)$  and  $J(k)$  are both block Jordan matrices of order  $k$ , with their diagonal blocks respectively equal to  $B$  and  $0$ . We show how to produce such a permutation in the simple case  $k = 2$ . It will be clear that the same is true for  $k > 2$ . We have

$$C(2) = \begin{pmatrix} B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & B & 0 \\ 0 & I & 0 & 0 \end{pmatrix}.$$

Let us now consider a directed graph that has one node associated with each row of  $C(2)$ , and with an arc going from node  $i$  to node  $j$  whenever block  $(i, j)$  is non-zero. The graph of  $C(2)$  is shown in Figure 3.2.1.

In general, the graph of  $C(k)$  has the following structure: every node  $i$  such that  $3 \leq i \leq 2k$  has an arc going to node  $i - 2$  with label  $I$ , and every odd-numbered node  $i$  has an arc going to itself, with label  $B$ . Hence we see that the nodes of the graph can be grouped into two subsets: the odd numbered nodes and the even numbered nodes. The desired permutation is simply to reorder the nodes as  $(1, 3, \dots, 2k - 1, 2, 4, \dots, 2k)$ , because the subgraph associated with the odd-numbered nodes is the graph of  $B(k)$ ,



and the subgraph associated with the even-numbered nodes is the graph of  $J(k)$ . So we have, for  $k = 2$ :

$$D(2) = \begin{pmatrix} B & 0 & 0 & 0 \\ I & B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \end{pmatrix}.$$

which is indeed of the form (3.2.19). Now it is easy to verify that

$$J(k)^k = 0$$

and that  $J(k)$  is of index  $k$ . Then

$$D(k)^D = \begin{pmatrix} B(k)^{-1} & 0 \\ 0 & 0 \end{pmatrix}$$

by definition, as in Lemma 1.2.6. Now  $B(k)^{-1}$  is given by Equation (3.2.14), and by a graph argument similar to the above we see that  $A(k)^D$  has the required shape as in equation (3.2.18). □

It is straightforward to verify that  $A(k)A(k)^D$  is a block diagonal matrix with all

diagonal blocks equal to  $AA^\#$ . For  $k = 4$ , we have

$$\begin{aligned} A(4)A(4)^D &= \begin{pmatrix} A & 0 & 0 & 0 \\ I & A & 0 & 0 \\ 0 & I & A & 0 \\ 0 & 0 & I & A \end{pmatrix} \begin{pmatrix} A^\# & 0 & 0 & 0 \\ -(A^\#)^2 & A^\# & 0 & 0 \\ (A^\#)^3 & -(A^\#)^2 & A^\# & 0 \\ -(A^\#)^4 & (A^\#)^3 & -(A^\#)^2 & A^\# \end{pmatrix} \\ &= \begin{pmatrix} AA^\# & 0 & 0 & 0 \\ 0 & AA^\# & 0 & 0 \\ 0 & 0 & AA^\# & 0 \\ 0 & 0 & 0 & AA^\# \end{pmatrix}. \end{aligned}$$

The consequence is that  $y(k) = A(k)^D b(k)$  is not a solution of (3.2.10) in general, because  $AA^\# b_i \neq b_i$  for  $i \geq 2$ . Nonetheless, we can use  $A(k)^D$  to construct a solution. Define a new matrix  $R(k)$  whose only non-zero blocks are on the superdiagonal and are all equal to  $W = I - AA^\#$ . For example,  $R(1) = 0$  and with  $k = 4$  we have

$$R(4) = \begin{pmatrix} 0 & W & 0 & 0 \\ 0 & 0 & W & 0 \\ 0 & 0 & 0 & W \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

We show the following result.

**Theorem 3.2.7.** If the system  $Ax_1 = b_1$  has a solution, then the system  $A(k)x(k) = b(k)$  has a solution given by

$$x(k) = [A(k)^D + R(k)]b(k). \quad (3.2.20)$$

Moreover,  $x_1, x_2, \dots, x_{k-1}$  are uniquely determined while  $x_k + Wy$  is also a solution, for any vector  $y$ .

**Proof.** Expanding the matrix product in (3.2.20) gives precisely the solution of Theorem 3.2.4. □

**Corollary 3.2.8.** The terms  $w_{-1}, \dots, w_i$  of the series (3.1.6) are given by  $[A(k)^D + R(k)]b(k)$ , where  $k = i + 2$  and  $b_k$  is defined as in Corollary 3.2.5.

### 3.3. Computataion of gain, bias and other terms

A consequence of Corollary 3.2.3 is that the terms  $g$ ,  $h$  and  $w_i$ , for  $i = 1, 2, \dots$ , of the series (3.1.6) can be computed recursively, by solving successive systems of linear equations, all with the same singular matrix of coefficients.

One such algorithm is described in Veinott (1969), Page 1651. It requires parsing the transition matrix  $P$  into its recurrent subchains and transient states. Gaussian elimination is used to compute the solutions.

Another approach is to use the matrix decomposition algorithms of Chapter I. Applying the fast decomposition algorithm of Figure 1.3.5 to the matrix  $A = I - P$ , both the gain  $g$  and the bias  $h$  can be computed directly using the fast solution algorithm of Figure 1.4.3 with the right hand side  $b = r$ , where  $r$  is the vector of immediate rewards.

The terms  $w_i$ , for  $i = 1, 2, \dots$ , can be computed as follows. Observe that at Step 3.2 of the fast solution algorithm, the bias is given by

$$h = PL \begin{pmatrix} z_1 \\ 0 \end{pmatrix},$$

where  $z_1$  was computed at Step 2.3. Let  $z_1^{(0)} = z_1$ .

**Lemma 3.3.1.** Let  $z_1^{(i+1)}$  be given by

$$z_1^{(i+1)} = -B_{11}^{-1} z_1^{(i)}, \quad (3.3.1)$$

where the product by  $B_{11}^{-1}$  is computed as in Figure 1.4.2. Then

$$w_i = PL \begin{pmatrix} z_1^{(i)} \\ 0 \end{pmatrix}, \quad i = 0, 1, 2, \dots, \quad (3.3.2)$$

where  $P$  and  $L$  were obtained with the fast decomposition algorithm.

**Proof.** Recall that the fast decomposition algorithm of Figure 1.3.5 factorized the matrix  $A$  as  $A = PLUQ$ . Then the matrix  $B$ , although never explicitly computed, was defined as  $B = UQPL$ , so that

$$A = (PL) \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} (PL)^{-1}.$$

Using Equation (1.3.8), we have

$$A^\# = (PL) \begin{pmatrix} B_{11}^{-1} & B_{11}^{-1} B_{11}^{-1} B_{12} \\ 0 & 0 \end{pmatrix} (PL)^{-1}. \quad (3.3.3)$$

We show by induction that Equation (3.3.2) holds. We know it holds for  $i = 0$ , the bias term. By Equation (3.1.5), we have

$$w_{i+1} = (-A^\#)((-1)^i (A^\#)^{i+1} r) = -A^\# w_i, \quad i = 0, 1, 2, \dots$$

Hence

$$\begin{aligned} w_{i+1} &= -(PL) \begin{pmatrix} B_{11}^{-1} & B_{11}^{-1} B_{11}^{-1} B_{12} \\ 0 & 0 \end{pmatrix} (PL)^{-1} (PL) \begin{pmatrix} z_1^{(i)} \\ 0 \end{pmatrix} \\ &= PL \begin{pmatrix} -B_{11}^{-1} z_1^{(i)} \\ 0 \end{pmatrix}, \end{aligned}$$

and the result follows from Equation (3.3.1).  $\square$

The amount of arithmetic required is about the same for both the method of Veinott (1969) and our fast solution algorithm (when the number of recurrent classes is small), provided that only a small number of terms are required (less than  $n$ ). We have no computational experience from which to compare their relative performance, however.

### 3.4. Iterative computation of gain and bias

When only the gain and bias terms are required [e.g., when using the average reward criterion as in Howard (1960) and Blackwell (1962)], and when the transition matrix  $P$  is large and sparse, it may be appropriate to use an iterative method to compute  $g$  and  $h$ .

We describe such an algorithm based on the powers of the transition matrix, and we show that its rate of convergence is geometric when  $P$  is aperiodic, using the results of Chapter II. Also, a stopping criterion is proposed, based on a sensitivity analysis of the evaluation equations.

The iterative algorithm is described in Figure 3.4.1. We call it the power method because, as we will see, it produces the same sequence of iterates for the gain as the standard power method for computing an eigenvector.

Step 0. Set  $g^{(0)} = r$ ,  $x^{(1)} = r$  and  $N = 1$ .  
 Step 1. Compute  $x^{(N+1)} = r + Px^{(N)}$ .  
 Step 2. Compute  $g^{(N)} = x^{(N+1)} - x^{(N)}$ .  
 Step 3. Compute  $h^{(N)} = x^{(N)} - Ng^{(N)}$ .  
 Step 4. If  $N \cdot \|g^{(N)} - g^{(N-1)}\| \leq \delta$  then stop,  
 else add 1 to  $N$  and go to Step 1.

**Figure 3.4.1. Power method for gain and bias**

**Theorem 3.4.1.** Suppose  $P$  is aperiodic, and let the sequences  $g^{(N)}$  and  $h^{(N)}$  be defined as in Figure 3.4.1. Then

$$\lim_{N \rightarrow \infty} g^{(N)} = g \quad \text{and} \quad \lim_{N \rightarrow \infty} h^{(N)} = h,$$

where  $g = P^*r$  and  $h = A^\#r$ .

**Proof.** From Step 1, we have

$$x^{(N+1)} = \sum_{i=0}^N P^i r,$$

and hence

$$g^{(N)} = P^N r, \tag{3.4.1}$$

which converges when  $P$  is aperiodic, by Theorem 2.3.9. Further, we have

$$\begin{aligned} h^{(N)} &= \sum_{i=0}^{N-1} P^i r - NP^N r \\ &= \sum_{i=0}^{N-1} (P^i - P^*)r - N(P^N - P^*)r. \end{aligned} \tag{3.4.2}$$

The first term converges to  $h = A^\#r$ , by Theorem 2.3.10, and the second term goes to zero. □

**Lemma 3.4.2.** Suppose  $P$  is aperiodic, and let  $E^{(N)} = P^N - P^*$ , as in Equation (2.4.1).

Then

$$e_1^{(N)} = g^{(N)} - g = E^{(N)}r \quad (3.4.3)$$

and

$$e_2^{(N)} = h^{(N)} - h = -(A^\# + NI)E^{(N)}r. \quad (3.4.4)$$

**Proof.** Equation (3.4.3) is immediate from (3.4.1). To show (3.4.4), recall that

$$A^\# = \sum_{i=0}^{\infty} (P^i - P^*),$$

by Theorem 2.3.10, when  $P$  is aperiodic. Hence from (3.4.2), we have

$$\begin{aligned} h^{(N)} - h &= \sum_{i=0}^{N-1} (P^i - P^*)r - \sum_{i=0}^{\infty} (P^i - P^*)r - NE^{(N)}r \\ &= -\sum_{i=N}^{\infty} (P^i - P^*)r - NE^{(N)}r \\ &= -\left(\sum_{i=0}^{\infty} (P^i - P^*)\right)(P^N - P^*)r - NE^{(N)}r \\ &= -A^\#E^{(N)}r - NE^{(N)}r, \end{aligned}$$

and the result follows. □

To show that the convergence is geometric, let us measure the error with

$$\epsilon_1^{(N)} = \|E^{(N)}\|_2 \cdot \|r\|_2 \quad (3.4.5)$$

and

$$\epsilon_2^{(N)} = (\|A^\#\|_2 + N)\epsilon_1^{(N)}. \quad (3.4.6)$$

Then

$$\|e_1^{(N)}\|_2 \leq \epsilon_1^{(N)}$$

and

$$\|e_2^{(N)}\|_2 \leq \epsilon_2^{(N)}.$$

**Corollary 3.4.3.** Suppose  $P$  is aperiodic. Then

$$\lim_{N \rightarrow \infty} \frac{\epsilon_1^{(N+1)}}{\epsilon_1^{(N)}} = \gamma \quad \text{and} \quad \lim_{N \rightarrow \infty} \frac{\epsilon_2^{(N+1)}}{\epsilon_2^{(N)}} = \gamma,$$

where  $\gamma < 1$  is the modulus of the subdominant eigenvalue of  $P$ .

**Proof.** By Theorem 2.4.4. □

Although this result indicates that the convergence is asymptotically geometric, it is to be observed that the factor  $\|A^\#\|_2 + N$  in Equation (3.4.6) implies that the error on the bias is always greater than the error on the gain.

Now that the rate of convergence of the algorithm has been determined, we still have to justify the stopping rule given at Step 4 of Figure 3.4.1, and to specify some guidelines for the choice of the parameter  $\delta$ .

We proceed as follows, using residual errors. Let

$$\delta_1^{(N)} = Ag^{(N)} \tag{3.4.7}$$

and

$$\delta_2^{(N)} = g^{(N)} + Ah^{(N)} - r. \tag{3.4.8}$$

These can be computed easily from  $g^{(N)}$  and  $h^{(N)}$ , although we will see that their computation is not necessary. Now since  $g$  and  $h$  satisfy the evaluation equations, the (unknown) errors  $e_1^{(N)}$  and  $e_2^{(N)}$  must satisfy

$$Ae_1^{(N)} = \delta_1^{(N)} \tag{3.4.9}$$

and

$$e_1^{(N)} + Ae_2^{(N)} = \delta_2^{(N)}. \tag{3.4.10}$$

**Theorem 3.4.4.** The gain and bias errors are related to the residual errors by

$$e_1^{(N)} = A^\#\delta_1^{(N)}$$

and

$$e_2^{(N)} = A^\#\delta_2^{(N)}.$$

**Proof.** We have that  $\delta_1^{(N)}$  is compatible with  $A$  by construction, from Equation (3.4.7). Hence Theorem 3.2.4 gives

$$e_1^{(N)} = A^\#\delta_1^{(N)} + P^*\delta_2^{(N)} \tag{3.4.11}$$

and

$$e_2^{(N)} = A^\# \delta_2^{(N)} + P^* y, \quad (3.4.12)$$

where  $y$  is arbitrary. Now

$$P^* \delta_2^{(N)} = P^* g^{(N)} + P^* A h^{(N)} - P^* r = 0$$

because  $g = P^* g^{(N)} = P^* r$  and  $P^* A = 0$ . Also, Equation (3.4.4) implies that

$$P^* e_2^{(N)} = -P^* (A^\# + NI) E^{(N)} r = 0,$$

because  $P^* A^\# = 0$  and  $P^* E^{(N)} = 0$ . Hence  $P^* y = 0$ , and the result follows.  $\square$

The consequence of this result is that

$$\|e_1^{(N)}\|_2 \leq \|A^\#\|_2 \cdot \|\delta_1^{(N)}\|_2, \quad (3.4.13)$$

so that the error on the gain is bounded by a fixed factor  $\|A^\#\|_2$  times the residual error. This factor indicates the sensitivity (or conditioning) of the Markov chain with transition matrix  $P$  [see Meyer (1980)] to changes in the data.

Now observe that, at Step 4 of Figure 4.3.1, we have

$$g^{(N)} - g^{(N-1)} = (P - I)g^{(N-1)} = -A g^{(N-1)} = -\delta_1^{(N-1)}.$$

Hence our stopping criterion simply ensures that the residual error is small enough. By choosing  $\delta$  such that

$$\|A^\#\|_2 \cdot \delta \leq \epsilon \text{ say,}$$

we see from Equations (3.4.5) and (3.4.6) that both  $\epsilon_1^{(N)} \leq \epsilon$  and  $\epsilon_2^{(N)} \leq \epsilon$ , assuming that the problem is reasonably well conditioned (i.e., that  $\|A^\#\|_2$  is not too big).

### 3.5. Markov decision processes

Now consider a system in which an action must be taken before a transition occurs. Let  $N = \{1, \dots, n\}$  be the (finite) state space, and let  $A_i$  be the finite set of actions that can be taken when the system is in state  $i$ , for  $i = 1, \dots, n$ .

The system evolves in time according to the process  $\{X(t) : t = 0, 1, 2, \dots\}$ , where  $X(t)$  is the state of the system at time  $t$ . Let  $\{a(t) : t = 0, 1, 2, \dots\}$  be the sequence of selected actions, with  $a(t) \in A_{X(t)}$ .

We assume that the system satisfies the Markov property and that the transition probabilities do not depend explicitly on time. Hence we define

$$p_{ij}^k = \text{Prob}\{X(t+1) = j \mid X(t) = i, a(t) = k\},$$

for  $k \in A_i$  and  $i, j = 1, \dots, n$ . We assume that the system satisfies the Markov property, that is

$$\text{Prob}\{X(t+1) = j \mid X(0), a(0), \dots, X(t), a(t)\} = \text{Prob}\{X(t+1) = j \mid X(t), a(t)\}.$$

Immediately before a transition occurs, the system earns an (expected) reward  $r_i^k$ , where  $k = a(t)$  and  $i = X(t)$ .

The action  $a(t)$  is selected according to a function  $\pi$ , called a *policy*, such that

$$a(t) = \pi(t, X(0), a(0), \dots, X(t-1), a(t-1), X(t)), \text{ for } t = 0, 1, 2, \dots$$

Now let us denote by  $P$  the transition probability function, by  $r$  the reward function and by  $\Pi$  the set of all policies. Then the above model is called a *Markov decision process* and is denoted by the quadruple  $(N, r, P, \Pi)$ .

For a given policy  $\pi \in \Pi$  and interest rate  $\rho > 0$ , we define the expected total discounted reward function  $v_i^\pi(\rho)$  such that

$$v_i^\pi(\rho) = E_i^\pi \left( \sum_{k=0}^{\infty} \beta^{k+1} r_{X(k)}^{a(k)} \right), \quad i = 1, \dots, n,$$

where  $\beta = (1 + \rho)^{-1}$  and  $E_i^\pi$  is the conditional expectation operator under policy  $\pi$ . That is, for an arbitrary random variable  $Y$  defined from  $\{X(t)\}$  and  $\{a(t)\}$ , we have

$$E_i^\pi(Y) = E^\pi\{Y \mid X(0) = i\}, \quad i = 1, \dots, n,$$

where the expectation is taken under policy  $\pi$ .

The problem is to find a policy  $\pi \in \Pi$  that maximizes  $v_i^\pi(\rho)$  in some sense, for  $i = 1, \dots, n$ . Now define a function  $\delta$  to be a *decision rule* if  $\delta_i \in A_i$ , for  $i = 1, \dots, n$ , and let  $\Delta = A_1 \times A_2 \times \dots \times A_n$  be the set of all decision rules. Then a policy  $\pi$  is said to be a *Markov policy* if

$$\pi(t, X(0), a(0), \dots, X(t)) = \delta_i(t),$$

where  $i = X(t)$  and  $\delta(t) \in \Delta$ . Moreover, a Markov policy  $\pi$  is said to be *stationary* if  $\delta(0) = \delta(1) = \dots$

For problems with an infinite time horizon, it has been shown that there is a stationary policy that is optimal [see e.g., Heyman and Sobel (1984), Theorems 4-3 and 4-4] for the criteria we will consider. Hence there is no loss of generality in restricting the problem to stationary policies. For simplicity of notation, we will denote a stationary policy by its decision rule  $\delta$ , and the set of stationary policies by  $\Delta$ .

For a fixed, stationary policy  $\delta \in \Delta$ , the system is a Markov chain with rewards and we obtain the Markov reward process  $(N, r^\delta, P^\delta)$ , as in Section 3.1, with transition matrix  $P = P^\delta$  and reward vector  $r = r^\delta$ . For a given interest rate  $\rho > 0$  and discount factor  $\beta = (1 + \rho)^{-1}$ , the expected total discounted reward  $v^\delta(\rho)$  is given by

$$v^\delta(\rho) = \sum_{i=0}^{\infty} \beta^{i+1} (P^\delta)^i r^\delta$$

as in Section 3.1. Following Miller and Veinott (1969), we say that a policy  $\delta$  is  *$\rho$ -optimal* if

$$v^\delta(\rho) \geq v^\gamma(\rho) \quad \forall \gamma \in \Delta. \quad (3.5.1)$$

It was shown by Blackwell (1962) that there is a policy  $\delta \in \Delta$  that is  $\rho$ -optimal for every small enough  $\rho > 0$ . Such a policy is said to be *Blackwell optimal*. Motivated by the series (3.1.6), Veinott (1969) and (1974) defined a policy  $\delta$  to be  *$k$ -discount optimal* if

$$\lim_{\rho \searrow 0} \rho^{-k} [v^\delta(\rho) - v^\gamma(\rho)] \geq 0 \quad \forall \gamma \in \Delta. \quad (3.5.2)$$

(Veinott's definition also extends to the case of  $\rho < 0$ , but we will stick to the positive case, which Veinott(1969) calls  $k^+$  discount optimality). As special cases, a  $-1$ -discount optimal policy is also said to be *gain optimal*, while a  $0$ -discount optimal policy is also said to be *bias optimal* [Denardo (1970)] or *nearly optimal* [Blackwell (1962)].

The policy iteration method of Howard (1960) and Blackwell (1962) produces a policy  $\delta$  that is gain optimal. This method has been extended by Veinott (1966) to find a bias optimal policy and Miller and Veinott (1969) to find a Blackwell optimal policy. Veinott (1969) modified it to find a  $k$ -discount optimal policy, for any  $k \geq -1$ .

More precisely, let  $w_i^\delta$ ,  $i = -1, 0, 1, \dots$ , be the terms of the Laurent series (3.1.6), for a policy  $\delta \in \Delta$ . Then (3.5.2) implies that  $\delta$  is  $k$ -discount optimal if

$$\sum_{i=-1}^k \rho^i w_i^\delta \geq \sum_{i=-1}^k \rho^i w_i^\gamma, \quad \forall \gamma \in \Delta \quad (3.5.3)$$

for all small enough  $\rho > 0$ . The extended algorithm evaluates  $w_{-1}^\delta, w_0^\delta, \dots, w_{k+1}^\delta$  to show that no other policy is lexicographically better, up to  $w_k$ .

Now let  $n$  be the number of states of the process. Miller and Veinott (1969) showed that a policy is Blackwell optimal if and only if it is  $n$ -discount optimal. This result was extended by Denardo (1971) who showed that an  $(n-1)$ -discount optimal policy is Blackwell optimal. Then Veinott (1974), using the fact that  $\text{rank}(A) = n - m$ , where  $A = I - P$  and  $m$  is the number of recurrent classes of  $P$ , showed that any  $(n-m)$ -discount optimal policy is Blackwell optimal. We give a new proof of the latter result, using our matrix decomposition approach. As Miller and Veinott (1969), we need the following lemma from Gantmacher (1959), which we state without proof.

**Lemma 3.5.1.** Let  $M$  be a  $k \times k$  matrix and  $L$  a linear subspace of  $R^k$ . If  $M^i x \in L$  for  $i = 0, 1, \dots, k-1$ , then  $M^i x \in L$  for  $i = k, k+1, \dots$

(The idea is that for  $i \geq k$ ,  $M^i x$  is a linear combination of  $x, Mx, \dots, M^{k-1}x$ ). This implies the following.

**Theorem 3.5.2.** Suppose that some policy  $\delta \in \Delta$  has  $m$  recurrent classes (i.e.,

$1 \leq m \leq n$ ). Suppose also that some policy  $\gamma \in \Delta$  is such that  $w_i^\gamma = w_i^\delta$ , for  $i = -1, 0, \dots, n - m$ . Then  $w_i^\gamma = w_i^\delta$  for  $i > n - m$ .

**Proof.** First, observe that in the special case  $m = n$ , the result is trivially true because  $g^\gamma = g^\delta$ ,  $A^\delta = I - P^\delta = 0$  and  $(A^\delta)^\# = 0$ , so that  $h^\delta = w_0^\delta = w_1^\delta = \dots = 0$ . This implies that  $h^\gamma = 0$ . Now by equations (3.5.4) and (3.5.6),

$$w_i^\gamma = (-1)^i ((A^\gamma)^\#)^i h^\gamma = 0 = w_i^\delta, \quad i \geq 0.$$

In the case  $m < n$ , both policies have the same gain and bias, i.e.  $w_{-1}^\delta = w_{-1}^\gamma = g$  and  $w_0^\delta = w_0^\gamma = h$ . Equations (3.5.4) and (3.5.6) imply that

$$w_i^\delta = (-1)^i ((A^\delta)^\#)^i h, \quad i \geq 0. \quad (3.5.4)$$

Decompose  $A^\delta$  as in equation (1.3.1), and define

$$y_i = \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} = S w_i^\delta$$

where  $y_{i,1}$  and  $y_{i,2}$  have dimensions  $(n - m)$  and  $m$ , respectively. Define also

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = S h = S (A^\delta)^\# r^\delta = \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} S r^\delta.$$

Hence  $z_2 = 0$ . Multiply Equation (3.5.4) by  $S$  to get

$$\begin{aligned} \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} &= S w_i^\delta = (-1)^i S S^{-1} \begin{pmatrix} (B^{-1})^i & 0 \\ 0 & 0 \end{pmatrix} S h \\ &= (-1)^i \begin{pmatrix} (B^{-1})^i & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \end{aligned}$$

so that

$$\begin{aligned} y_{i,1} &= (-1)^i (B^{-1})^i z_1, \\ y_{i,2} &= 0. \end{aligned} \quad i \geq 0. \quad (3.5.5)$$

Now apply the same transformation  $S$  to  $A^\gamma$ , giving

$$A^\gamma = S^{-1} \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} S \quad (3.5.6)$$

and define

$$x_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \end{pmatrix} = S w_i^\gamma.$$

Equation (3.2.9), with  $i=i+1$  and  $A = A^\gamma$  together with (3.5.6), gives

$$\begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \begin{pmatrix} x_{i+1,1} \\ x_{i+1,2} \end{pmatrix} = - \begin{pmatrix} x_{i,1} \\ x_{i,2} \end{pmatrix}, \quad i \geq 0. \quad (3.5.7)$$

By hypothesis,  $x_{i,1} = y_{i,1}$  and  $x_{i,2} = 0$ , for  $i = 0, \dots, n - m$  so

$$\begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \begin{pmatrix} y_{i+1,1} \\ 0 \end{pmatrix} = - \begin{pmatrix} y_{i,1} \\ 0 \end{pmatrix}, \quad i = 0, \dots, n - m - 1.$$

Using (3.5.5), we get

$$(X_{11}B^{-1} - I)(B^{-1})^i z_1 = 0 \quad (3.5.8)$$

and

$$X_{21}(B^{-1})^{i+1} z_1 = 0. \quad (3.5.9)$$

Now recall that  $z_1$  is a vector of dimension  $(n - m)$ . Define a linear subspace  $L$  of  $\mathbb{R}^{n-m}$  such that  $x \in L$  iff  $(X_{11}B^{-1} - I)x = 0$ . Then Equation (3.5.8) implies that  $(B^{-1})^i z_1 \in L$ , for  $i = 0, \dots, n - m - 1$ . By Lemma 3.5.1, it is true also for  $i \geq n - m$  and hence Equation (3.5.8) is also valid for  $i \geq n - m$ . The same argument applies to (3.5.9) as well.

We still have to show that it implies that  $x_{i,1} = y_{i,1}$  and  $x_{i,2} = y_{i,2} = 0$ , for  $i > n - m$ . By Corollary 3.2.3, the recurrence (3.5.7) has a unique solution. Equations (3.5.8) and (3.5.9) imply that  $x_{i,1} = y_{i,1} = (-1)^i (B^{-1})^i z_1$  and  $x_{i,2} = y_{i,2} = 0$  is that solution for every  $i \geq 0$ .  $\square$

The following example shows that  $\delta$  and  $\gamma$  in Theorem 3.5.2 need not have the same number of recurrent classes. Let policy  $\delta$  have the transition matrix  $P^\delta$  and reward vector  $r^\delta$  given by

$$P^\delta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{and} \quad r^\delta = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 3 \end{pmatrix}.$$

Note  $n = 4$  and  $m = 2$ . With  $A = I - P^\delta$  we have

$$A^\# = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & -1/3 \\ 0 & -1/3 & 1/3 & 0 \\ 0 & 0 & -1/3 & 1/3 \end{pmatrix}$$

so that

$$g^\delta = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}, h^\delta = \begin{pmatrix} 0 \\ -2/3 \\ 1/3 \\ 1/3 \end{pmatrix}, w_1^\delta = \begin{pmatrix} 0 \\ 1/3 \\ -1/3 \\ 0 \end{pmatrix} \quad \text{and} \quad w_2^\delta = \begin{pmatrix} 0 \\ -1/9 \\ 2/9 \\ -1/9 \end{pmatrix}.$$

Let  $\gamma$  be a policy with a transition matrix given by

$$P^\gamma = \begin{pmatrix} 1 - 3x & x & x & x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

with  $0 \leq x \leq 1/3$ , and  $r^\gamma = r^\delta$ . Then  $g^\gamma = g^\delta$ ,  $h^\gamma = h^\delta$ ,  $w_1^\gamma = w_1^\delta$  and  $w_2^\gamma = w_2^\delta$ . Thus  $\delta$  and  $\gamma$  have the same series expansion, but  $\gamma$  has only one recurrent class if  $x > 0$ .

We now apply the above result to dynamic programming.

**Theorem 3.5.3.** A policy with  $m$  recurrent classes is Blackwell optimal iff it is  $(n - m)$ -discount optimal.

**Proof.** By applying Theorem 3.5.2 to Equation (3.5.3). □

It implies, in general, that a policy is Blackwell optimal if and only if it is  $(n - 1)$ -discount optimal (because  $m \geq 1$ , always). We will now use this result and the notation of the previous section to formulate a policy iteration algorithm in a unified way. We first recall that with a fixed interest rate  $\rho$ , the policy iteration method for finding a  $\rho$ -optimal policy can be formulated as in Figure 3.5.1, with  $\beta = (1 + \rho)^{-1}$ .

Using the notation of Section 3.2, we can now formulate the policy iteration method for finding a  $k$  discount optimal policy as follows. Let  $\kappa = k + 3$  (we need to evaluate  $\kappa$  terms  $w_{-1}, w_0, \dots, w_{k+1}$ ). The method is shown in Figure 3.5.2.

The operator  $A^\delta(\kappa)^D + R^\delta(\kappa)$  of Step 1, derived from Drazin inverse theory, provides an explicit solution to the policy evaluation problem. It is to be seen whether the latter algorithm could be formulated as a specialization of the method of Newton for solving systems of nonlinear equations, as the former was by Puterman and Brumelle (1979) using the ordinary inverse  $(A^\delta)^{-1}$ .

**Step 0.** Select an arbitrary policy  $\delta \in \Delta$ .

**Step 1.** Policy evaluation.

Compute  $v^\delta = (A^\delta)^{-1}r^\delta$ , where  $A^\delta = I - \beta P^\delta$ .

**Step 2.** Policy Improvement.

Find  $\gamma \in \Delta$  s.t.  $r^\gamma + \beta P^\gamma v^\delta = \max_{\eta \in \Delta} \{r^\eta + \beta P^\eta v^\delta\}$ .

(Choose  $\gamma = \delta$  if possible).

**Step 3.** If  $\gamma = \delta$  then stop

else set  $\delta = \gamma$  and go to Step 1.

**Figure 3.5.1. Policy iteration method for  $\rho$ -optimal policy**

**Step 0.** Select an arbitrary policy  $\delta \in \Delta$ .

**Step 1.** Policy evaluation.

Compute  $x^\delta(\kappa) = [A^\delta(\kappa)^D + R^\delta(\kappa)]b^\delta(\kappa)$ ,

as in Corollary 3.2.5.

**Step 2.** Policy improvement.

Find  $\gamma \in \Delta$  s.t.  $b^\gamma(\kappa) + P^\gamma(\kappa)x^\delta(\kappa) = \max_{\eta \in \Delta} \{b^\eta(\kappa) + P^\eta(\kappa)x^\delta(\kappa)\}$

where  $P^\gamma(\kappa) = I(\kappa) - A^\gamma(\kappa)$

and the maximization is done lexicographically.

(Choose  $\gamma = \delta$  if possible).

**Step 3.** If  $\gamma = \delta$  then stop

else set  $\delta = \gamma$  and go to Step 1.

**Figure 3.5.2. Policy iteration method for  $k$  discount optimal policy**

It is important to note, however, that the formulation of Figure 3.5.2 is not the most

appropriate for computations. Indeed, we saw in Section 3.2 that the terms  $w_i^\delta$ , for  $i = -1, \dots, k$ , can be evaluated successively. Hence it is more efficient to combine Steps 1 and 2 in such a way that only those terms which are really required be computed.

There are two ways to do it. One is to successively evaluate  $x^\delta(i) = w_{i-3}$  and produce  $\Delta_i$  such that

$$\Delta_i = \{\gamma \in \Delta_{i-1} : b^\gamma(i) + P^\gamma x^\delta(i) \geq b^\eta(i) + P^\eta x^\delta(i), \forall \eta \in \Delta_{i-1}\}$$

where  $\Delta_0 = \Delta$ , and stop at the smallest  $i \leq k$  for which  $\Delta_i$  has only one element. This element is the policy  $\gamma$  of Step 2. (If  $\Delta_k$  has more than one element, choose any  $\gamma \in \Delta_k$ ).

Another way is to stop at the smallest  $i \leq k$  for which  $\Delta_i$  does not contain  $\delta$ , and choose  $\gamma \in \Delta_i$  arbitrarily. Then  $\gamma$  is a better policy than  $\delta$ , although it is not necessarily the best improvement. This latter approach might require fewer terms to be evaluated than the former.

## Chapter IV

### Continuous time Markov decision processes

This chapter is mainly a review of known results on continuous time Markov reward processes and decision processes. Its purpose is to show the equivalence between continuous time and discrete time Markov decision processes, and to introduce some of the notation that will be used in the next chapter.

In Section 4.1, we define the continuous time Markov reward process from the continuous time Markov chain of Çinlar (1975). Then we show, in Section 4.2, that the resolvent has the same algebraic structure as in the discrete time case. Our proof is more detailed than that of Howard (1960) and uses the same approach as in Chapter II. This result is used, in Section 4.3, to obtain a Laurent expansion of the resolvent. The gain and bias terms are also interpreted as limits of the reward functions.

Finally, in Section 4.4 the equivalence between continuous time and discrete time processes is reviewed, using a data transformation considered by Howard (1960), Veinott (1969), Schweitzer (1971) and Serfozo (1979).

#### 4.1. Continuous time Markov reward processes

Let  $\{X(t) : t \geq 0\}$  be a homogeneous, continuous time Markov chain with state space  $N = \{1, \dots, n\}$ . Let  $S_k$  be the epoch of the  $k$ -th transition, with  $S_0 = 0$ . Then we have

$$\text{Prob}\{S_{k+1} - S_k > t \mid X(S_k) = i\} = e^{-\lambda_i t}, \quad t \geq 0, \quad (4.1.1)$$

where  $0 \leq \lambda_i < \infty$ , for  $i = 1, \dots, n$ .

The transition rates are finite ( $\lambda_i < \infty$ ) because the state space is finite [see Çinlar (1975), page 244]. We will also assume that they are positive ( $\lambda_i > 0$ ). Then  $\tau_i = 1/\lambda_i$  is the expected holding time in state  $i$ , for  $i = 1, \dots, n$ , and the time matrix  $T = \text{diag}(\tau_i)$  is nonsingular.

The transition probabilities are defined as

$$p_{ij} = \text{Prob}\{X(S_{k+1}) = j \mid X(S_k) = i\}, \quad i, j = 1, \dots, n, \quad (4.1.2)$$

with  $p_{ii} = 0$  and  $\sum_{j=1}^n p_{ij} = 1$ , for  $i = 1, \dots, n$ , so that the matrix  $P = (p_{ij})$  is stochastic. For  $k = 0, 1, 2, \dots$ , let  $Y(k) = X(S_k)$ . Then the process  $\{Y(k) : k = 0, 1, 2, \dots\}$  is a discrete time Markov chain with transition matrix  $P$ , embedded at the transition epochs.

The continuous time Markov chain  $\{X(t) : t \geq 0\}$  is completely determined by the matrix

$$Q = T^{-1}(P - I), \quad (4.1.3)$$

which is called its *infinitesimal generator*.

Assume that  $X(t)$  gives the state of a dynamic system. Further, define the earning rate vector  $r = (r_i)$  as follows. Assume that when the system is in state  $i$ , it earns a reward at a rate of  $u_{ii}$  dollars per unit of time. Further, just before making a transition to state  $j$ , a reward of  $u_{ij}$  dollars is earned instantly, for  $i, j = 1, \dots, n$ . From this, the expected earning rates are defined as

$$r_i = u_{ii} + \lambda_i \sum_{j \neq i} p_{ij} u_{ij} \quad (4.1.4)$$

[see Howard (1960), Equation (8.16)].

Such a model is a *continuous time Markov reward process* and is denoted by the triplet  $(N, r, Q)$ .

Define  $Z(t)$  as the total reward earned until time  $t$ . With  $E_i$  as the conditional expectation operator of Section 3.1, we define the vector  $V(t)$  of expected undiscounted

rewards by

$$V_i(t) = E_i\{Z(t)\}, \text{ for } i = 1, \dots, n.$$

The function  $V_i(t)$  is known to be differentiable, and its derivative  $u_i(t)$  has a limit, as  $t \rightarrow \infty$ , for  $i = 1, \dots, n$ . Indeed, define the matrix  $\Pi(t)$  such that

$$\pi_{ij}(t) = \text{Prob}\{X(t) = j \mid X(0) = i\}, \quad t \geq 0.$$

Then  $\pi_{ij}(t)$  has a limit as  $t \rightarrow \infty$  [see Heyman and Sobel (1982), Proposition 8-8], and the expected earning rate at time  $t$  is

$$u_i(t) = \sum_{j=1}^n \pi_{ij}(t) r_j, \quad i = 1, \dots, n.$$

This implies that  $V(t) = O(t)$ , because

$$\lim_{t \rightarrow \infty} \frac{V_i(t)}{t} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t u_i(x) dx$$

is bounded.

Let  $s > 0$  be the instantaneous interest rate, and let  $v(s)$  be the vector of expected total discounted rewards. That is,  $v_i(s)$  is the expected total discounted reward of the process over an infinite horizon, when the process starts in state  $i$ , for  $i = 1, \dots, n$ .

Using the conditional expectation operator  $E_i$ , we have

$$\begin{aligned} v_i(s) &= E_i \left\{ \int_0^{\infty} e^{-st} dZ(t) \right\} \\ &= \int_0^{\infty} e^{-st} dV_i(t), \quad i = 1, \dots, n. \end{aligned}$$

The expectation and integration operators can be permuted since the result is finite, because  $V(t) = O(t)$ . Conditioning on the first transition, we get

$$v_i(s) = \int_0^{\infty} \left[ \int_0^t e^{-sx} u_{ii} dx + e^{-st} \sum_{j \neq i} p_{ij} (u_{ij} + v_j(s)) \right] \lambda_i e^{-\lambda_i t} dt,$$

which can be integrated directly, giving

$$\begin{aligned} v_i(s) &= \frac{1}{\lambda_i + s} \left[ r_i + \lambda_i \sum_{j \neq i} p_{ij} v_j(s) \right] \\ &= \frac{1}{1 + s\tau_i} \left[ \tau_i r_i + \sum_{j \neq i} p_{ij} v_j(s) \right]. \end{aligned} \tag{4.1.5}$$

Multiplying both sides of Equation (4.1.5) by  $(1 + s\tau_i)$ , and using matrix notation, we get

$$(I + sT)v(s) = Tr + Pv(s).$$

Collecting terms and premultiplying both sides by  $T^{-1}$ , we finally obtain the resolvent equation

$$(sI + T^{-1}A)v(s) = r, \quad (4.1.6)$$

where  $A = I - P$ .

It is not obvious whether the matrix  $(sI + T^{-1}A)$  is nonsingular for any small enough interest rate  $s$ . We will show, in the next section, that it is indeed the case, and that the resolvent  $R_s = (sI + T^{-1}A)^{-1}$  has exactly the same properties as the resolvent  $R_p$  of a discrete time process. Hence all the results of discrete time dynamic programming that are based on the algebraic structure of  $R_p$  are also valid for continuous time Markov decision processes.

## 4.2. Algebraic structure of the resolvent

All the results of Chapter III, for discrete time decision processes, are based on the fact that  $\text{ind}(A) = 1$ , where  $A = I - P$ . We show that  $\text{ind}(T^{-1}A) = 1$ , where  $T$  is the diagonal matrix of expected transition times. Our argument is similar to that of Howard (1960), page 98, but we emphasize the algebraic structure.

The result is not true for a general matrix  $A$  of index 1. For example, let

$$A = \begin{pmatrix} 3/2 & 1/2 \\ -3/2 & -1/2 \end{pmatrix} \text{ and } T^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}.$$

We have  $\text{ind}(A) = 1$ , because

$$A = X^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X, \text{ with } X = \begin{pmatrix} -3 & -1 \\ 1 & 1 \end{pmatrix} \text{ and } X^{-1} = \begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 3/2 \end{pmatrix}.$$

We have, however, that  $\text{ind}(T^{-1}A) = 2$ , because

$$T^{-1}A = X^{-1} \begin{pmatrix} 0 & -3 \\ 1 & 4 \end{pmatrix} X X^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} X = X^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} X,$$

so that  $T^{-1}A$  is similar to  $J_2(0)$  which has index 2, by Lemma 1.2.1.

We now proceed to show that the result is true when  $A = I - P$ , with  $P$  a stochastic matrix.

**Lemma 4.2.1.** Suppose  $P$  is an irreducible, stochastic matrix, where  $A = I - P$ . Then  $T^{-1}A$  has the eigenvalue 0 with multiplicity 1.

**Proof.** We proceed as in the proof of Theorem 2.2.4. We know that the column vector  $e$  is a right hand eigenvector of  $P$  with eigenvalue 1, and that there is a positive row vector  $w$  that is the corresponding left hand eigenvector of  $P$ . This implies that  $e$  is also a right hand eigenvector of  $T^{-1}A$  with eigenvalue 0, and that  $wT$  is the corresponding left hand eigenvector of  $T^{-1}A$ .

We show that the corresponding Jordan block of  $T^{-1}A$  has dimension 1, by proving that Equation (1.1.6) has no solution. That is, we have to show that  $T^{-1}Ay = e$  has no solution  $y$ . Suppose  $y$  is a solution. Then

$$(wT)(T^{-1}A)y = (wT)e > 0,$$

because both  $wT$  and  $e$  are positive vectors. But  $wT$  is an eigenvector of  $T^{-1}A$  with eigenvalue 0, so that  $(wT)(T^{-1}A) = 0$ , which is a contradiction. Hence the result follows.  $\square$

**Theorem 4.2.2.** Suppose  $P$  is a stochastic matrix. Then  $\text{ind}(T^{-1}A) = 1$ , where  $A = I - P$ .

**Proof.** Let  $m$  be the number of recurrent classes of  $P$ . Then by Theorem 2.1.3, there is a permutation matrix  $S$  such that

$$A = S^{-1} \begin{pmatrix} A_{00} & A_{01} & A_{02} & \dots & A_{0m} \\ 0 & A_{11} & 0 & \dots & 0 \\ \vdots & \ddots & A_{22} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & A_{mm} \end{pmatrix} S.$$

Here  $A_{00} = I - P_{00}$  is the nonsingular submatrix corresponding to the transient states of  $P$ . Now, applying  $S$  to the diagonal matrix  $T^{-1}$ , we have

$$T^{-1}A = S^{-1} \begin{pmatrix} T_{00}^{-1}A_{00} & T_{00}^{-1}A_{01} & T_{00}^{-1}A_{02} & \dots & T_{00}^{-1}A_{0m} \\ 0 & T_{11}^{-1}A_{11} & 0 & \dots & 0 \\ \vdots & \ddots & T_{22}^{-1}A_{22} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & T_{mm}^{-1}A_{mm} \end{pmatrix} S,$$

where  $T_{ii}$  is the diagonal submatrix of  $T$  corresponding to the  $i$ -th subset, for  $i = 0, \dots, m$ .

The matrix  $T_{00}^{-1}A_{00}$  is nonsingular, hence all its eigenvalues are nonzero. Moreover, by Lemma 4.2.1, each  $T_{ii}^{-1}A_{ii}$ , for  $i = 1, \dots, m$ , has the eigenvalue 0 with multiplicity 1. This implies that the Jordan canonical form of  $T^{-1}A$  has exactly  $m$  Jordan submatrices with eigenvalue 0, each of dimension 1. Hence  $\text{ind}(T^{-1}A) = 1$ .  $\square$

The consequence of Theorem 4.2.2 is that the resolvent  $R_s$  of a continuous time Markov process exists and has a Laurent expansion analogous to that of the resolvent  $R_p$  of a discrete time process.

### 4.3. Laurent expansion of the discounted rewards

Let  $C = T^{-1}(I - P) = -Q$ . Then the resolvent of the continuous time process is  $R_s = (sI + C)^{-1}$ , and the vector of total discounted rewards is given by  $v(s) = R_s u$ . Because  $\text{ind}(C) = 1$ , we know that  $C^\#$  exists. Moreover, let  $W = I - CC^\#$ . The following results follow immediately from the results of Chapter III.

**Theorem 4.3.1.** Let  $\lambda$  be the nonzero eigenvalue of  $A$  with smallest modulus. If  $0 < s < |\lambda|$ , then

$$R_s = s^{-1}W + \sum_{i=0}^{\infty} (-s)^i (C^\#)^{i+1}. \quad (4.3.1)$$

**Proof.** Same as Theorem 3.1.1.  $\square$

Defining  $g = Wr$ ,  $h = C^\#r$  and  $w_i = (-s)^i (C^\#)^{i+1}r$ , for  $i = 0, 1, \dots$ , we have

$$v(s) = s^{-1}g + h + \sum_{i=1}^{\infty} s^i w_i. \quad (4.3.2)$$

**Theorem 4.3.2.** The terms  $g$ ,  $h$  and  $w_i$ , for  $i = 1, 2, \dots$ , of the series (4.3.2) are the unique solution of the recurrence

$$Cg = 0 \quad (4.3.3)$$

$$g + Ch = r \quad (4.3.4)$$

$$w_{i-1} + Cw_i = 0, \quad i = 1, 2, \dots, \quad (4.3.5)$$

where  $w_0 = h$ .

**Proof.** Same as Corollary 3.2.3. □

This implies that the policy iteration method of Chapter III can be applied directly to the continuous time Markov decision processes, using the matrix  $C$  instead of  $A$ . Further, the terms  $g$  and  $h$  of the series (4.3.2) can still be interpreted as *gain* and *bias*, respectively. We do this as follows.

Recall from Section 4.1 that the expected total discounted reward  $v_i(s)$  is the Laplace transform of the expected earning rate function  $u_i(t)$ , that is,

$$v_i(s) = \int_0^{\infty} e^{-st} u_i(t) dt, \quad i = 1, \dots, n.$$

We need the following Tauberian theorem.

**Theorem 4.3.3.** [Heyman and Sobel (1982), Property A-8]. Suppose  $\tilde{f}(s)$  is the Laplace transform of  $f(t)$ . Then

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \searrow 0} s\tilde{f}(s),$$

provided both limits exist.

**Corollary 4.3.4.** The vector  $g$  of Equation (4.3.2) is the limiting earning vector, that is,

$$\lim_{t \rightarrow \infty} u_i(t) = g_i, \quad i = 1, \dots, n.$$

**Proof.** We know from Section 4.1 that  $u_i(t)$  has a limit as  $t \rightarrow \infty$ . By Equation (4.3.2), we also have that

$$\lim_{s \searrow 0} sv_i(s) = g_i.$$

The result follows from the theorem. □

**Corollary 4.3.5.** The vector  $h$  of Equation (4.3.2) is the bias vector, that is,

$$\lim_{t \rightarrow \infty} (V_i(t) - g_i t) = h_i, \quad i = 1, \dots, n.$$

**Proof.** Let  $f_i(t) = V_i(t) - g_i t$ . Then

$$\begin{aligned} \tilde{f}_i(s) &= \int_0^{\infty} e^{-st} V_i(t) dt \\ &= \int_0^{\infty} e^{-st} V_i(t) dt - \int_0^{\infty} e^{-st} g_i t dt \\ &= \frac{v_i(s)}{s} - \frac{g_i}{s^2}, \end{aligned}$$

where the last equality follows from elementary properties of the Laplace transforms.

Hence

$$\lim_{s \searrow 0} s \tilde{f}_i(s) = h_i,$$

by Equation (4.3.2). Also,  $f_i(t)$  has a limit as  $t \rightarrow \infty$  because it is bounded and  $u_i(t)$  converges. The result follows from the theorem.  $\square$

We note that it is sufficient to use ordinary limits. In the discrete time case, it was necessary to use Cesàro convergence when the transition matrix was periodic. It is not necessary in the continuous time case, because the exponential distribution of the transition times automatically does the time averaging.

#### 4.4. Equivalent Markov decision processes

In this section, the continuous time Markov decision process is defined. By examining the reward processes associated to stationary policies, a whole family of discrete time Markov decision processes are shown to be equivalent to a given continuous time process.

A continuous time Markov decision process is a system that evolves according to a process  $\{X(t) : t \geq 0\}$  in which an action is selected every time a transition occurs, and which earns a reward that depends both on the state of the system and the action selected.

More precisely, suppose the  $k$ -th transition occurs at time  $t = S_k$  and that the system moves to state  $i = Y_k = X(S_k)$ . Then an action  $a = a_k \in \Delta_i$  is selected and the next transition will occur according to the following probability function:

$$\text{Prob}\{Y_{k+1} = j, S_{k+1} > t + x \mid Y_k = i, S_k = t, a_k = a\} = p_{ij}^a e^{-\lambda_i^a x}, \quad (4.4.1)$$

where  $0 < \lambda_i^a < \infty$ ,  $p_{ij}^a \geq 0$ ,  $p_{ii}^a = 0$  and

$$\sum_{j=1}^n p_{ij}^a = 0, \text{ for } a \in \Delta_i, i = 1, \dots, n.$$

During the interval  $(S_k, S_{k+1})$ , a reward is earned at a rate of  $u_{ii}^a$  dollars per unit of time and an additional amount of  $u_{ij}^a$  dollars is received at time  $S_{k+1}$ .

For a policy  $\pi$ , define  $Z^\pi(t)$  as the total reward earned until time  $t$ , and the vector  $V^\pi(t)$  such that

$$V_i^\pi(t) = E_i\{Z^\pi(t)\}, \quad i = 1, \dots, n.$$

Then with an interest rate  $s > 0$ , the expected total discounted reward vector  $v^\pi(s)$  is defined as

$$v_i^\pi(s) = \int_0^\infty e^{-st} dV_i^\pi(t), \quad i = 1, \dots, n.$$

The objective is to find a policy  $\pi$  to select the actions so that  $v_i^\pi(s)$  is maximized for all  $i = 1, \dots, n$ . It has been shown that there is a stationary policy that is optimal [see Veinott (1969)]. Hence it is sufficient to consider only the stationary policies.

Suppose a fixed stationary policy  $\delta \in \Delta$  is used, where  $\Delta = \Delta_1 \times \dots \times \Delta_n$ . Then the action  $a_k = \delta_i$  is selected if the  $k$ -th transition is to state  $i = X(S_k)$ . Hence the system is the continuous time Markov reward process  $(N, r^\delta, Q^\delta)$ , where

$$r_i^\delta = u_{ii}^{\delta_i} + \lambda_i^{\delta_i} \sum_{j \neq i} p_{ij}^{\delta_i} u_{ij}^{\delta_i}$$

and

$$q_{ij}^{\delta_i} = \begin{cases} -\lambda_i^{\delta_i} & \text{if } i = j \\ \lambda_i^{\delta_i} p_{ij}^{\delta_i} & \text{if } i \neq j. \end{cases}$$

As was pointed out in the previous section, the solution of a continuous time Markov reward process has the same structure as in the discrete time case, and can be computed using the same algorithms. Further, the discount optimality criteria can be defined in just the same manner, from the Laurent series, and a  $k$ -discount optimal policy can be found with the policy improvement methods.

This similarity between continuous and discrete time processes was first pointed out by Howard (1960) for the expected total discounted reward criteria as well as the average reward criterion. Howard (1960) also noted that a given continuous time Markov decision process is equivalent to a whole family of discrete time processes, in the sense that for any stationary policy  $\delta \in \Delta$ , all these processes have the same value.

This can be seen as follows. Consider the continuous time Markov reward process  $(N, r, Q)$ , where  $r = r^\delta$  and  $Q = Q^\delta$ , and let  $\alpha$  be a positive real constant. Then the reward process  $(N, \alpha r, \alpha Q)$  is equivalent to the original process, up to a scaling of the time unit.

Now by choosing  $\alpha$  appropriately, we can define a stochastic matrix  $\tilde{P}$  such that

$$\alpha Q = \tilde{P} - I.$$

But then the discrete time Markov reward process  $(N, \alpha r, \tilde{P})$  has the same value as the original continuous time process, because

$$\tilde{A} = I - \tilde{P} = -\alpha Q$$

so that the terms of the series for the discrete time process satisfy the same equations as for the continuous time process (compare Corollary 3.2.3 and Theorem 4.3.2, with  $C = -Q$ ).

If the scalar  $\alpha$  is chosen such that

$$0 < \alpha < 1/\lambda_i^\alpha, \quad \forall a \in \Delta_i, \quad i = 1, \dots, n,$$

then such a stochastic matrix  $\tilde{P}^\delta$  exists for all policies  $\delta \in \Delta$ . Hence such an  $\alpha$  defines a discrete time Markov decision process which is equivalent to the original continuous time process.

This equivalence was suggested by Howard (1960) for the expected total discounted reward criterion and the average reward criterion. Veinott (1969) extended the result to the discount optimality criteria.

The same device was suggested by Schweitzer (1971) as a data transformation to guarantee the convergence of the method of successive approximations to find a gain optimal policy (average reward criterion). To accomodate models with  $p_{ii}^a < 0$  (strictly speaking, those are semi-Markov decision processes), the scalar  $\alpha$  must be selected such that

$$0 < (1 - p_{ii}^a)\alpha < 1/\lambda_i^a, \quad \forall a \in \Delta_i, \quad i = 1, \dots, n.$$

Finally, the equivalence was extended to models with a countably infinite state space by Serfozo (1979).

## Chapter V

### Semi-Markov decision processes

In this chapter, we apply the matrix decomposition approach to semi-Markov decision processes. We follow the development of Denardo (1971), in which Laplace-Stieltjes transforms are used to derive a Laurent expansion of the expected total discounted rewards.

In Section 5.1, we define a semi-Markov reward process in a way that is compatible with the definition of a semi-Markov decision process in Heyman and Sobel (1984).

The material of Section 5.2 is completely new. The matrix decomposition approach is applied to the moments of the transition probability function, and a condition of Denardo (1971) about the entries of the first moment is interpreted algebraically. A new algorithm is also introduced to solve some singular systems of equations involving the first moment.

In Section 5.3, many results of Denardo (1971) are reviewed, and their proofs are formulated in terms of matrix decomposition, using the results of Section 5.2. The Laurent expansion of the expected total discounted rewards is obtained, emphasizing the matrix formulation.

In Section 5.4, the application of the above results to semi-Markov decision processes is reviewed, mainly following Denardo (1971).

Finally, in Section 5.5 some special cases of semi-Markov decision processes are reviewed. As pointed out in Denardo (1971) and Heyman and Sobel (1984), both

discrete time and continuous time Markov decision processes can be seen as special cases. Their formulation as semi-Markov decision processes is a good illustration of the model. Also, the equivalence between the semi-Markov and continuous time Markov decision processes is reviewed, under the average reward criterion. The equivalence was used in Ross (1970), but our approach stresses the fact that the processes are not equivalent with respect to other optimality criteria (e.g., bias optimality, etc.).

### 5.1. Semi-Markov reward processes

Let  $\{Y(k), S(k) : k = 0, 1, 2, \dots\}$  be a two-dimensional process such that  $Y(k) \in N = \{1, \dots, n\}$ ,  $S(0) = 0$  and, for all  $x \geq 0$  and  $k = 0, 1, 2, \dots$ ,

$$\begin{aligned} & \text{Prob}\{Y(k+1) = j, S(k+1) \leq x+t \mid Y(0), S(0), \dots, Y(k) = i, S(k) = x\} \\ &= \text{Prob}\{Y(k+1) = j, S(k+1) \leq x+t \mid Y(k) = i, S(k) = x\} \\ &= Q_{ij}(t), \quad t \geq 0, \quad i, j = 1, \dots, n. \end{aligned} \tag{5.1.1}$$

Then  $\{Y(k) : k = 0, 1, 2, \dots\}$  is a discrete time Markov chain with transition matrix  $P = Q(\infty)$ , and  $S(k)$  is the time of the  $k$ -th transition.

Let  $X(k)$  be the reward earned during the  $k$ -th stage of the process. The probability distribution of  $X(k)$  is assumed to depend on  $Y(k)$ ,  $Y(k+1)$  and  $\tau(k) = S(k+1) - S(k)$ . We define the conditional reward functions  $u_{ij}(x, y)$  as the expected value of the cumulative reward during  $[S(k), S(k) + y]$  if  $Y(k) = i$ ,  $Y(k+1) = j$  and  $\tau(k) = x$ , with  $0 \leq y < x$ , and

$$u_{ij}(x, x) = E\{X(k) \mid Y(k) = i, Y(k+1) = j, \tau(k) = x\}.$$

Unconditioning, we get the reward function  $R_i(t)$  such that

$$R_i(t) = \sum_{j=1}^n \left[ \int_{0^-}^{t^+} u_{ij}(x, x) dQ_{ij}(x) + \int_{t^+}^{\infty} u_{ij}(t, x) dQ_{ij}(x) \right] \tag{5.1.2}$$

is the expected reward earned during the interval

$$[S(k), \min\{S(k) + t, S(k+1)\}],$$

before the  $(k+1)$ -th transition. A sufficient condition for the integrals to exist in (5.1.2) is if  $u_{ij}(x, x)$  and  $u_{ij}(t, x)$  have bounded variation in  $[0, \infty]$  and  $[t, \infty]$ , respectively.

Such a model is called a *semi-Markov reward process* and is denoted by the triplet  $(N, R, Q)$ , where  $R(t)$  is the vector of expected reward functions of Equation (5.1.2), and  $Q(t)$  is the matrix of transition probability functions of Equation (5.1.1).

The discrete time and continuous time Markov reward processes of Sections 3.1 and 4.1 are special cases of the semi-Markov reward process.

In the discrete time case, we have directly

$$Q_{ij}(t) = \begin{cases} 0 & \text{if } t < 1, \\ p_{ij} & \text{if } t \geq 1, \end{cases}$$

and

$$R_i(t) = \begin{cases} 0 & \text{if } t < 1, \\ r_i & \text{if } t \geq 1. \end{cases}$$

In the continuous time case, we have

$$Q_{ij}(t) = p_{ij}(1 - e^{-\lambda_i t}), \text{ for } i \neq j,$$

and

$$u_{ij}(x, y) = \begin{cases} u_{ii}y & \text{if } y < x, \\ u_{ii}y + u_{ij} & \text{if } y = x, \end{cases}$$

so that by Equation (5.1.2),

$$\begin{aligned} R_i(t) &= \sum_{j=1}^n \left[ \int_0^t (u_{ii}x + u_{ij}) p_{ij} \lambda_i e^{-\lambda_i x} dx + \int_t^\infty u_{ii} t p_{ij} \lambda_i e^{-\lambda_i x} dx \right] \\ &= \left( \frac{u_{ii}}{\lambda_i} + \sum_{j \neq i} p_{ij} u_{ij} \right) (1 - e^{-\lambda_i t}) \\ &= \frac{r_i}{\lambda_i} (1 - e^{-\lambda_i t}), \end{aligned}$$

where  $r_i$  was defined in Equation (4.1.4). Hence  $Q_{ij}(\infty) = p_{ij}$  and  $R_i(\infty) = r_i/\lambda_i$  is the expected one-stage reward.

Now for  $t \geq 0$ , let  $Z(t)$  be the total reward earned by the system until time  $t$ . We are interested in the vector  $V(t)$  of expected total undiscounted rewards until time  $t$ , i.e.,

$$V_i(t) = E_i\{Z(t)\}, \quad i = 1, \dots, n.$$

For  $t > 0$ , the expectation is finite if the expected number of transitions in  $[0, t]$  is finite. A sufficient condition for this in terms of  $Q(t)$  is described in the Section 5.2..

Then with interest rate  $s > 0$ , we have the total discounted reward  $v_i(s)$  defined as

$$v_i(s) = \int_{0-}^{\infty} e^{-st} dV_i(t), \text{ for } i = 1, \dots, n. \quad (5.1.3)$$

Hence  $v(s)$  is precisely the Laplace-Stieltjes transform of  $V(t)$ .

Similarly, consider the transforms  $q(s)$  and  $r(s)$ , respectively of  $Q(t)$  and  $R(t)$ . We have

$$q_{ij}(s) = \int_{0-}^{\infty} e^{-st} dQ_{ij}(t) \quad (5.1.4)$$

and

$$r_i(s) = \int_{0-}^{\infty} e^{-st} dR_i(t), \quad (5.1.5)$$

for  $i, j = 1, \dots, n$ . We define the  $k$ -th normalized moments  $Q_k$  and  $R_k$  as

$$Q_k = \int_{0-}^{\infty} t^k dQ(t)/k! \quad (5.1.6)$$

and

$$R_k = \int_{0-}^{\infty} t^k dR(t)/k!, \quad (5.1.7)$$

for  $k = 0, 1, 2, \dots$ , whenever they exist. Note that  $Q_k$  is a matrix and  $R_k$  is a vector.

These moments are useful because of the following Abelian theorem.

**Theorem 5.1.1.** [See Heyman and Sobel (1982), Property A-9]. Suppose  $f(s)$  is the Laplace-Stieltjes transform of  $F(t)$ . Then

$$\int_{0-}^{\infty} e^{-st} t^k dF(t) = (-1)^k \frac{d^k}{ds^k} f(s). \quad (5.1.8)$$

**Corollary 5.1.2.** Suppose  $Q_0, Q_1, \dots, Q_k$  exist and are finite, for some  $k \geq 0$ . Then

$$q(s) = Q_0 - sQ_1 + \dots + (-s)^k Q_k + o(s^k). \quad (5.1.9)$$

**Proof.** Evaluate Equation (5.1.8) at  $s = 0$ , with  $f(s) = q_{ij}(s)$ . By (5.1.6), we have

$$\frac{d^\ell}{ds^\ell} q_{ij}(s) \Big|_{s=0} = (-1)^\ell \ell! (Q_\ell)_{ij} \quad \text{for } \ell = 0, \dots, k.$$

The result follows by taking a Taylor expansion of  $q_{ij}(s)$  in a positive neighbourhood of  $s = 0$ . □

**Corollary 5.1.3.** Suppose  $R_0, R_1, \dots, R_\ell$  exist and are finite, for some  $\ell \geq 0$ . Then

$$r(s) = R_0 - sR_1 + \dots + (-s)^\ell R_\ell + o(s^\ell). \quad (5.1.10)$$

**Proof.** Same as the previous corollary, but with  $f(s) = r_i(s)$ . □

A similar series expansion will be derived in Section 5.3 for the expected total discounted reward function  $v(s)$ . But first, some preliminary results on the algebraic structure of the first moment  $Q_1$  of the transition probability function are derived.

## 5.2. Decomposition of the first moment

Let us explore the algebraic properties of the moments  $Q_0$  and  $Q_1$  of the transition probability function  $Q(t)$ . These properties will be useful for expressing the solution of the singular equations which will be encountered when deriving the Laurent expansion of the discounted reward function  $v(s)$ .

The properties of the zeroth moment  $Q_0$  were explored in Chapters II and III, since

$$Q_0 = \int_0^\infty dQ(t) = Q(\infty) = P,$$

where  $P = (p_{ij})$  is the transition matrix of the Markov chain embedded at the transition epochs. For the rest of this section, we will denote  $Q_0$  by  $P$ . Also, we assume that  $P$  is a stochastic matrix.

Let also  $A = I - P$ . Then, as in Chapter III,  $A$  satisfies Equation (1.3.1), and hence  $A^\#$  exists and is given by (1.3.2). Again, we will need the limiting matrix  $P^* = I - AA^\#$ . Recall that

$$P = S^{-1} \begin{pmatrix} P_{11} & 0 \\ 0 & I \end{pmatrix} S \text{ and } P^* = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} S, \quad (5.2.1)$$

where the identity matrix is  $m \times m$ , with  $m$  the number of recurrent classes of  $P$ .

For the rest of this section, we denote the first moment  $Q_1$  of  $Q(t)$  by  $H$ , i.e.,  $Q_1 = H = (h_{ij})$ . By Equation (5.1.6), if  $p_{ij} > 0$  then  $h_{ij}/p_{ij}$  is the expected holding

time in state  $i$ , conditioned on the next state being  $j$ , for  $i, j = 1, \dots, n$ . The following results show that if the matrix  $H$  has at least one nonzero entry in each of the recurrent classes of the embedded Markov chain  $P$ , then the expected number of transitions in a finite time interval is finite.

**Lemma 5.2.1.** Suppose  $P$  is irreducible, and suppose that  $H \neq 0$ . Then

$$\text{rank}(P^*HP^*) = \text{rank}(P^*) = 1$$

**Proof.** Let  $\pi > 0$  be the row vector of the stationary probabilities of  $P$ . Then

$$P^* = \begin{pmatrix} \pi \\ \vdots \\ \pi \end{pmatrix},$$

and  $\text{rank}(P) = 1$ . Now

$$HP^* = \begin{pmatrix} \alpha_1 \pi \\ \vdots \\ \alpha_n \pi \end{pmatrix},$$

where  $\alpha_i = \sum_{j=1}^n h_{ij}$ , for  $i = 1, \dots, n$ . There is at least one  $i$  for which  $\alpha_i > 0$ , because  $H \neq 0$ . Then

$$P^*HP^* = \beta \begin{pmatrix} \pi \\ \vdots \\ \pi \end{pmatrix} = \beta P^*,$$

where  $\beta = \sum_{j=1}^n \pi_j \alpha_j \geq \pi_i \alpha_i > 0$ , and the result follows.  $\square$

**Lemma 5.2.2.** Suppose  $P$  has  $m$  recurrent classes, and suppose  $H$  has at least one nonzero entry in each recurrent class of  $P$ . Then

$$\text{rank}(P^*HP^*) = \text{rank}(P^*) = m$$

**Proof.** That  $\text{rank}(P^*) = m$  is clear from Equation (5.2.1). We prove the other equality in the case  $m = 2$ . There is no added difficulty with  $m > 2$ , except for heavier notation.

As in Theorem 2.1.3, let  $T$  be a permutation matrix such that

$$P = T \begin{pmatrix} P_{00} & P_{01} & P_{02} \\ 0 & P_{11} & 0 \\ 0 & 0 & P_{22} \end{pmatrix} T^{-1} \text{ and } H = T \begin{pmatrix} H_{00} & H_{01} & H_{02} \\ 0 & H_{11} & 0 \\ 0 & 0 & Q_{22} \end{pmatrix} T^{-1}.$$

Then

$$P^* = T \begin{pmatrix} 0 & P_{01}^* & P_{02}^* \\ 0 & P_{11}^* & 0 \\ 0 & 0 & P_{22}^* \end{pmatrix} T^{-1},$$

and it is easy to verify that

$$P^* H P^* = T \begin{pmatrix} 0 & P_{01}^* H_{11} P_{11}^* & P_{02}^* H_{22} P_{22}^* \\ 0 & P_{11}^* Q_{11} P_{11}^* & 0 \\ 0 & 0 & P_{22}^* H_{22} P_{22}^* \end{pmatrix} T^{-1}.$$

This implies that  $\text{rank}(P^* H P^*) \geq m$ , because  $\text{rank}(P_{ii}^* H_{ii} P_{ii}^*) = 1$ , for  $i = 1, \dots, m$ , by Lemma 5.2.1. The equality will be proved in the corollary.  $\square$

Now let us apply the similarity transformation of Equation (5.2.1) to the matrix  $Q$ , and write

$$H = S^{-1} \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} S, \quad (5.2.2)$$

where  $H_{22}$  is an  $m \times m$  matrix (note that  $H_{11}$  and  $H_{22}$  are not the same here as in the proof of Lemma 5.2.2).

**Corollary 5.2.3.**  $H_{22}$  is nonsingular.

**Proof.** Using Equations (5.2.1) and (5.2.2), we have that

$$P^* H P^* = S^{-1} \begin{pmatrix} 0 & 0 \\ 0 & H_{22} \end{pmatrix} S.$$

Hence  $\text{rank}(H_{22}) = \text{rank}(P^* H P^*) = m$ , by Lemma 5.2.2. The result follows because  $H_{22}$  is an  $m \times m$  matrix, thus of full rank.  $\square$

This corollary has a consequence that will be important for deriving (and evaluating) the Laurent expansion of  $v(s)$ . Recall from Chapter I that a singular system of equations is said to be *compatible* if it has a solution. The following theorem shows that we can select the solution  $x$  of a compatible system

$$Ax = b \quad (5.2.3)$$

so that the system

$$Ay = c - Hx \quad (5.2.4)$$

be compatible as well.

**Theorem 5.2.4.** Suppose  $A$  is decomposed as in Equation (1.3.1), with  $B = I - P_{11}$ , and  $H$  as in (5.2.2), and suppose

$$b = S^{-1} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = S^{-1} \begin{pmatrix} b_1 \\ 0 \end{pmatrix} \text{ and } c = S^{-1} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Then the system (5.2.3) is compatible and has a unique solution  $x$  such that (5.2.4) is compatible, given by

$$x = S^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = S^{-1} \begin{pmatrix} B^{-1}b_1 \\ H_{22}^{-1}(c_2 - H_{21}B^{-1}b_1) \end{pmatrix}. \quad (5.2.5)$$

Moreover, there are infinitely many solutions  $y$  to Equation (5.2.4), given by

$$y = S^{-1} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = S^{-1} \begin{pmatrix} B^{-1}(c_1 - H_{11}x_1 - H_{12}x_2) \\ y_2 \end{pmatrix}, \quad (5.2.6)$$

where  $y_2$  is arbitrary.

**Proof.** Recall from Section 1.4 that the system (5.2.3) is compatible iff  $b_2 = 0$ . Writing (5.2.3) and (5.2.4) in decomposed form, we have

$$Bx_1 = b_1, \quad (5.2.7)$$

$$By_1 = c_1 - H_{11}x_1 - H_{12}x_2 \quad (5.2.8)$$

and

$$0 = c_2 - H_{21}x_1 - H_{22}x_2. \quad (5.2.9)$$

Since  $B$  is nonsingular,  $x_1$  is uniquely determined by (5.2.7), and (5.2.4) is compatible iff  $x_2$  satisfies (5.2.9). But since  $H_{22}$  is nonsingular, by Corollary 5.2.3, we have that  $x_2$  is uniquely determined. Finally,  $y_1$  is uniquely determined by (5.2.8), and  $y_2$  is arbitrary since it appears in none of the equations.  $\square$

The decomposition approach is not only a convenient way to express the solution of the above equations, it also provides an efficient algorithm for its computation.

We saw, in Chapter I, how the matrix  $A$  can be efficiently decomposed in a way that is equivalent to Equation (1.3.1). Moreover, we also saw how to use the decomposition to compute the component  $x_1$  of the solution, satisfying (5.2.7).

We now show how to compute  $x_2$  efficiently, to satisfy (5.2.9). Let  $d_2 = c_2 - H_{21}x_1$ . Then  $x_2$  is the solution of the nonsingular system  $H_{22}x_2 = d_2$ . The following lemmas show how to use the decomposition (1.3.1) to compute  $H_{21}x_1$  and  $H_{22}$  efficiently.

First, let  $z_2 = H_{21}x_1$ , and write

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} H_{11} \\ H_{21} \end{pmatrix} x_1. \quad (5.2.11)$$

Clearly, if we can compute  $z$  efficiently, then we have  $d_2 = c_2 - z_2$  immediately.

**Lemma 5.2.5.**

$$z = SHS^{-1} \begin{pmatrix} x_1 \\ 0 \end{pmatrix}.$$

**Proof.** We have

$$SHS^{-1} \begin{pmatrix} x_1 \\ 0 \end{pmatrix} = SS^{-1} \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} SS^{-1} \begin{pmatrix} x_1 \\ 0 \end{pmatrix} = \begin{pmatrix} H_{11} \\ H_{21} \end{pmatrix} x_1 = z.$$

□

Hence  $z$  is obtained by premultiplying  $\begin{pmatrix} x_1 \\ 0 \end{pmatrix}$  by  $S^{-1}$ , then by the matrix  $H$  and finally, by  $S$ .

The system  $H_{22}x_2 = d_2$  can be solved efficiently using Gaussian elimination, provided that the matrix  $H_{22}$  can be computed efficiently. That this is the case can be seen as follows.

**Lemma 5.2.6.**  $H_{22} = (0 \ I)SHS^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix}$ , where  $I$  is the  $m \times m$  identity matrix.

**Proof.** Because

$$SHS^{-1} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix},$$

by Equation (5.2.2), so we get

$$(0 \ I)SHS^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix} = (0 \ I) \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} 0 \\ I \end{pmatrix} = H_{22}.$$

□

This implies that, in order to compute  $H_{22}$ , we simply apply the transformation  $S^{-1}$  to  $m$  vectors, then we apply  $H$ , then  $S$  and finally,  $m$  row vectors. The total number of flops is proportional to  $mn^2$ . In fact, this product and the PLU decomposition of  $H_{22}$  needs to be done once only, because the solution for several right hand vectors  $b$  and  $c$  can be computed from the same decomposition.

The above method does not require parsing the Markov chain into its recurrent subsets. The chain structure was used only to derive the properties of the decomposition of the matrix  $H$ . A different approach was used in Denardo (1971), where the solution vector  $x$  is expressed in terms of  $P^*$ ,  $A^\#$  and, within each subchain, in terms of the coefficient  $\beta$  of the proof of Lemma 5.2.1. Our method is more economical because it does not require  $P^*$  and  $A^\#$  to be explicitly computed.

Another algorithm that does not require the parsing into its subchains is the method of Federgruen and Spreen (1980). It is based on a modified Gaussian elimination procedure similar to the PLUQ decomposition of Chapter I which we used in the fast decomposition algorithm.

Their formulation of the procedure does not include row interchanges, however, and fails to produce the desired elimination for some transition matrices. An example is

$$P = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \text{ with } A = I - P = \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}.$$

This matrix  $A$  cannot be made upper triangular by Gaussian elimination without row interchanges.

Fortunately, their results are still valid because it suffices to relabel the states in order to produce a transition matrix  $\tilde{P}$  for which the elimination is valid. In the above example, we get

$$\tilde{P} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } \tilde{A} = I - \tilde{P} = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix},$$

which is already upper triangular.

### 5.3. Series expansion of the discounted rewards

In this section, we return to the original notation  $Q_0, Q_1, \dots$ , for the moments of the transition probability function. At the root of the Laurent expansion of  $v(s)$  of Denardo (1971) is the following renewal equation for  $V(t)$ , which is obtained by conditioning on the time  $x$  of the first transition.

Recalling the definition of  $Q(t)$  and  $R(t)$ , we have

$$V_i(t) = R_i(t) + \sum_{j=1}^n \int_{0^-}^{t^+} dQ_{ij}(x) V_j(t-x),$$

where the integral includes the possible instantaneous rewards at times 0 and  $t$ . In matrix form, we have

$$V(t) = R(t) + \int_{0^-}^{t^+} dQ(x) V(t-x). \quad (5.3.1)$$

By the convolution theorem [see Heyman and Sobel (1982), Property A-12a], the Laplace-Stieltjes transforms satisfy

$$v(s) = r(s) + q(s)v(s),$$

which can be written as

$$[I - q(s)]v(s) = r(s). \quad (5.3.2)$$

This determines the function  $v(s)$  uniquely, for small  $s$ , because of the following lemma.

**Lemma 5.3.1.** Suppose  $Q_1$  has at least one nonzero entry per recurrent class of  $Q_0$ . Then the matrix  $I - q(s)$  is nonsingular.

**Proof.** By Equation (5.1.9), we have

$$I - q(s) = I - Q_0 + sQ_1 + o(s).$$

Decompose  $A = I - Q_0$  as in Equation (1.3.1) and write  $Q_1$  as in Equation (5.2.2).

We have

$$I - q(s) = S^{-1} \begin{pmatrix} B + sH_{11} + o(s) & sH_{12} + o(s) \\ sH_{21} + o(s) & sH_{22} + o(s) \end{pmatrix} S, \quad (5.3.3)$$

where  $B$  is nonsingular, and recall that  $H_{22}$  is also nonsingular, by Corollary 5.2.3. This implies that for  $s > 0$  small enough, both  $B + sH_{11} + o(s)$  and  $sH_{22} + o(s)$  are nonsingular, and the result follows.  $\square$

We now obtain the Laurent expansion of the discounted rewards  $v(s)$  in the special case when all moments exist. This is a special case of Theorem 1 of Denardo (1971).

**Theorem 5.3.2.** Suppose all moments of  $q(s)$  and  $r(s)$  exist and are finite. Then

$$v(s) = \frac{1}{s}V_{-1} + V_0 + sV_1 + s^2V_2 + \dots \quad (5.3.4)$$

where  $V_i$ , for  $i = -1, 0, 1, 2, \dots$ , are uniquely determined by

$$AV_{-1} = 0 \quad (5.3.5)$$

and

$$AV_i = (-1)^i R_i + \sum_{j=-1}^{i-1} (-1)^{i-j} Q_{i-j} V_j, \quad i = 0, 1, 2, \dots \quad (5.3.6)$$

**Proof.** Rewrite Equation (5.3.2) using the infinite expansions corresponding to (5.1.9), (5.1.10) and (5.3.4). We have

$$[I - Q_0 + sQ_1 - s^2Q_2 + \dots][s^{-1}V_{-1} + V_0 + sV_1 + \dots] = [R_0 - sR_1 + s^2R_2 - \dots].$$

Collecting terms, we get an infinite sequence of linear systems of equations, which can be written as follows, in block matrix form:

$$\begin{pmatrix} A & 0 & \dots & & & & \\ Q_1 & A & 0 & \dots & & & \\ -Q_2 & Q_1 & A & 0 & \dots & & \\ Q_3 & -Q_2 & Q_1 & A & 0 & \dots & \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ R_0 \\ -R_1 \\ R_2 \\ \vdots \end{pmatrix}, \quad (5.3.7)$$

where  $A = I - Q_0$ . The notation of (5.3.7) is equivalent to the sequence of (5.3.5) and (5.3.6). We still have to show that the  $V_i$  terms are uniquely determined.

We do this by induction on  $i$ . The induction hypothesis is that  $V_{-1}, \dots, V_{i-1}$  are uniquely determined by rows  $j = -1, \dots, i$ , for some  $i \geq -1$ . We have to show that there is a unique solution  $V_i$  for which row  $i + 1$  is a compatible system.

The hypothesis is trivially true for  $i = -1$ , because the system (5.3.5) is homogeneous. Now with  $V_{-1}, \dots, V_{i-1}$  fixed, we can write Equation (5.3.6) as

$$AV_i = b, \quad (5.3.8)$$

where  $b = (-1)^i R_i + \sum_{j=-1}^{i-1} (-1)^{i-j} Q_{i-j} V_j$ . The next equation, with  $V_{i+1}$ , can be written as

$$AV_{i+1} = c - Q_1 V_i, \quad (5.3.9)$$

where  $c = (-1)^{i+1} R_{i+1} + \sum_{j=-1}^i (-1)^{i+1-j} Q_{i+1-j} V_j$ . By Theorem 5.2.4, there is a unique  $V_i$  satisfying (5.3.8) for which (5.3.9) is compatible. This completes the proof.  $\square$

The result of Theorem 5.3.2 can also be extended to the case when only a finite number of moments are known. Then the terms of the series for  $v(s)$  are solution of a truncated system of equations. The following result is Lemma 2 of Denardo (1971). We give a new proof based on matrix decomposition.

**Lemma 5.3.3.** Suppose  $b(s) = o(s)$  and let  $x(s)$  be the solution of

$$[I - q(s)]x(s) = b(s). \quad (5.3.10)$$

Then  $x(s) = o(1)$ , i.e.,  $\lim_{s \rightarrow 0^+} x(s) = 0$ .

**Proof.** Write  $I - q(s)$  as in Equation (5.3.3) and let

$$Sx(s) = \begin{pmatrix} x_1(s) \\ x_2(s) \end{pmatrix} \text{ and } Sb(s) = \begin{pmatrix} b_1(s) \\ b_2(s) \end{pmatrix}.$$

Then Equation (5.3.10) becomes

$$\begin{pmatrix} A_{11}(s) & A_{12}(s) \\ A_{21}(s) & A_{22}(s) \end{pmatrix} \begin{pmatrix} x_1(s) \\ x_2(s) \end{pmatrix} = \begin{pmatrix} b_1(s) \\ b_2(s) \end{pmatrix}, \quad (5.3.11)$$

where

$$A_{11}(s) = B + sH_{11} + o(s)$$

$$A_{12}(s) = sH_{12} + o(s)$$

$$A_{21}(s) = sH_{21} + o(s)$$

$$A_{22}(s) = sH_{22} + o(s).$$

Hence  $A_{22}^{-1}(s) = \frac{1}{s}H_{22}^{-1} + o(s^{-1})$ . Now let  $C(s) = A_{11}(s) - A_{12}(s)A_{22}^{-1}(s)A_{21}(s)$ . Then  $C(s) = B + o(1)$ , so that  $C^{-1}(s) = B^{-1} + o(1)$ . We can write the solution of (5.3.1) directly as

$$x_1(s) = C^{-1}(s)[b_1(s) - A_{12}(s)A_{22}(s)b_2(s)] = o(s)$$

and

$$x_2(s) = A_{22}^{-1}(s)[b_2(s) - A_{21}(s)x_1(s)] = o(1).$$

This completes the proof. □

The following result is Theorem 1 of Denardo (1971), giving the truncated series of  $v(s)$ . Our proof is similar to that of Denardo (1971) but we emphasize the formulation with a block lower triangular matrix.

**Theorem 5.3.4.** Suppose  $Q_{k+2}$  and  $R_{k+1}$  are finite, for some  $k > 0$ . Then

$$v(s) = \frac{1}{s}V_{-1} + V_0 + sV_1 + \dots + s^kV_k + o(s^k), \quad (5.3.12)$$

where  $V_{-1}, \dots, V_k$  are uniquely determined by the finite system of equations

$$\begin{pmatrix} A & 0 & \dots & \dots & 0 \\ Q_1 & A & \ddots & & \vdots \\ -Q_2 & Q_1 & A & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ (-1)^{k+1}Q_{k+2} & \vdots & -Q_2 & Q_1 & A \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ \vdots \\ V_{k+1} \end{pmatrix} = \begin{pmatrix} 0 \\ R_0 \\ -R_1 \\ \vdots \\ (-1)^{k+1}R_{k+1} \end{pmatrix}, \quad (5.3.13)$$

and for which  $V_{k+1}$  is determined up to a term  $P^*y$ , where  $y$  is arbitrary.

**Proof.** Write  $v(s)$  as

$$v(s) = \frac{1}{s}V_{-1} + V_0 + sV_1 + \dots + s^kV_k + e_k(s).$$

Then, using (5.1.8) with  $k+1$  and (5.1.9) with  $l=k$ , Equation (5.3.2) becomes

$$\begin{aligned} & [A + sQ_1 - \dots - (-1)^{k+2}Q_{k+2} + o(s^{k+2})][s^{-1}V_{-1} + V_0 + sV_1 + \dots + s^kV_k + e_k(s)] \\ & = [R_0 - sR_1 + \dots + (-s)^{k+1}R_{k+1} + o(s^{k+1})]. \end{aligned} \quad (5.3.14)$$

Collecting terms, we get the system

$$\begin{pmatrix} A & 0 & \cdots & \cdots & 0 \\ Q_1 & A & \ddots & & \vdots \\ -Q_2 & Q_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A & 0 \\ (-1)^k Q_{k+1} & \ddots & -Q_2 & Q_1 & A \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ \vdots \\ V_k \end{pmatrix} = \begin{pmatrix} 0 \\ R_0 \\ -R_1 \\ \vdots \\ (-1)^k R_k \end{pmatrix} \quad (5.3.15)$$

together with

$$[I - q(s)]e_k(s) = s^{k+1} \left[ (-1)^{k+1} R_{k+1} + \sum_{j=-1}^k (-1)^{k+1-j} Q_{k+1-j} V_j \right] + o(s^{k+1}). \quad (5.3.16)$$

Dividing (5.3.16) by  $s^k$ , we get

$$[I - q(s)][e_k(s)/s^k] = o(s),$$

so that  $e_k(s)/s^k = o(1)$  by Lemma 5.3.3. Hence  $e_k(s) = o(s^k)$ . The extra equation in (5.3.13) ensures that  $V_k$  is uniquely determined and finite. The result on  $V_{k+1}$  follows because  $AP^* = 0$ .  $\square$

#### 5.4. Semi-Markov decision processes

A *semi-Markov decision process* is a model in which an action has to be selected each time the system makes a transition, and for which the transition probability function  $Q_{ij}^a(t)$  and the one-period expected reward function  $R_i^a(t)$  depend on the selected action  $a \in \Delta_i$ , for  $i, j = 1, \dots, n$ .

Without loss of optimality, and as in Denardo (1971), we consider only the policies  $\delta$  which are stationary. Again, we denote by  $\Delta = \Delta_1 \times \dots \times \Delta_n$  the set of stationary policies. Given a fixed policy  $\delta \in \Delta$ , the model reduces to the semi-Markov reward process  $(N, R^\delta, Q^\delta)$ , where  $N = \{1, \dots, n\}$ ,

$$R^\delta(t) = (R_i^\delta(t)) \quad i = 1, \dots, n,$$

and

$$Q^\delta(t) = (Q_{ij}^\delta(t)) \quad i, j = 1, \dots, n.$$

Correspondingly, the moments  $Q_k^\delta$  and  $R_k^\delta$ , for  $k = 0, 1, \dots$ , are defined by Equations (5.1.6) and (5.1.7), and the Laurent expansion of the expected total discounted rewards  $v(s)$  is given by Theorem 5.3.4, with moments  $V_{-1}^\delta, \dots, V_k^\delta$ , provided  $Q_{k+2}$  and  $R_{k+1}$  are finite.

Similarly to the discrete time case, a stationary policy  $\delta \in \Delta$  is said to be *s-optimal* if

$$v^\delta(s) \geq v^\gamma(s), \quad \forall \gamma \in \Delta.$$

Further, if  $\delta$  is *s-optimal* for all sufficiently small  $s > 0$ , then  $\delta$  is said to be *optimal*. Theorem 3 of Denardo (1971) gives a sufficient condition for an optimal policy to exist. We do not repeat it here.

Moreover, a policy  $\delta$  is said to be *k-discount optimal* if

$$\liminf_{s \rightarrow 0^+} s^{-k} [v^\delta(s) - v^\gamma(s)] \geq 0, \quad \forall \gamma \in \Delta.$$

As noted in Denardo (1971), with  $\Delta_{-2} = \Delta$  and

$$\Delta_k = \{\delta \in \Delta_{k-1} : V_k^\delta \geq V_k^\eta, \forall \eta \in \Delta_{k-1}\}, \quad k = -1, 0, \dots, \quad (5.4.1)$$

every policy in  $\Delta_k$  is *k-discount optimal* (provided that the relevant moments exist). Further, if there is an optimal policy  $\delta$ , then  $\delta \in \Delta_\infty$ .

Unlike in the discrete time case, there may be no finite  $k$  such that a *k-discount optimal* policy is optimal. Of course, if there is only one *k-discount optimal* policy (i.e., if  $|\Delta_k| = 1$ ), then it is optimal.

Policies in  $\Delta_{-1}$  and  $\Delta_0$  are called *gain optimal* and *bias optimal*, respectively. This is because  $V_{-1}$  and  $V_0$  can be interpreted as the gain and bias terms, since from Denardo (1971), page 483, we have

$$\lim_{t \rightarrow \infty} \frac{V(t)}{t} = V_{-1} \quad (5.4.2)$$

and

$$\lim_{t \rightarrow \infty} V(t) - tV_{-1} = V_0(C, 1), \quad (5.4.3)$$

where the Cesàro limit is defined as

$$\lim_{t \rightarrow \infty} f(t) = g(C, 1)$$

if

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t f(x) dx = g.$$

Hence  $V_{-1}$  is the average reward per unit of time, in the long run, and  $V_0$  is the total difference between the actual expected reward  $V(t)$  and the long run average  $V_{-1}t$ .

Provided the required moments exist for all policies, the policy iteration method of Figure 3.5.2 can be extended for finding a  $k$ -discount optimal policy of a semi-Markov decision process. But now, in the policy evaluation step (i.e., Step 1) we have to solve the system (5.3.13), as in Theorem 5.3.4. The terms  $V_{-1}, \dots, V_k$  can be evaluated as in Theorem 5.2.4 and Lemmas 5.2.5 and 5.2.6.

Moreover, in the policy improvement step (i.e., Step 2), we take the matrix  $P^\gamma(\kappa)$  as

$$P^\gamma(\kappa) = \begin{pmatrix} Q_0 & 0 & \dots & \dots & 0 \\ -Q_1 & Q_0 & \ddots & & \vdots \\ Q_2 & -Q_1 & Q_0 & \ddots & \vdots \\ \ddots & \ddots & \ddots & \ddots & 0 \\ (-1)^{k+2} Q_{k+2} & \ddots & Q_2 & -Q_1 & Q_0 \end{pmatrix}. \quad (5.4.4)$$

Hence the policy iteration method for discrete time Markov decision processes is a special case of the policy iteration method for semi-Markov decision processes, in which the matrix of moments is simpler.

### 5.5. Three special cases

In this section, we consider the discrete time and continuous time Markov decision processes as special cases of the semi-Markov decision process. This was already done in Denardo (1971), but we emphasize the similarity of the reward processes, using the matrix equations. We will see that the series expansion (5.3.4) is identical to (3.1.6), in the discrete time case, and to (4.3.1) in the continuous time case.

Moreover, we will also consider the equivalence between an arbitrary semi-Markov decision process and a corresponding continuous time Markov decision process, under the average reward criterion (i.e., gain optimality). This equivalence was also considered in Ross (1970), page 161. Again, we show the similarity of the reward processes using the matrix equations.

In a discrete time Markov reward process, the transition function  $Q_{ij}(t)$  is zero for  $0 \leq t < 1$ , and jumps to  $p_{ij}$  for  $t \geq 1$ , for  $i, j = 1, \dots, n$ . It is convenient to assume that a reward is earned just before a transition occurs, so that

$$R_i(t) = \begin{cases} 0 & \text{if } t < 1, \\ r_i & \text{if } t \geq 1, \end{cases} \quad i = 1, \dots, n.$$

The Laplace-Stieltjes transforms are obtained directly as

$$v(s) = e^{-s}P \text{ and } r(s) = e^{-s}r. \quad (5.5.1)$$

The discount factor  $\beta$  over one time period is precisely  $\beta = e^{-\rho}$ . As in Chapter III, let  $\beta = (1 + \rho)^{-1}$ , where  $\rho$  is the one period interest rate, giving

$$q(s) = (1 + \rho)^{-1}P \text{ and } r(s) = (1 + \rho)^{-1}r. \quad (5.5.2)$$

It is natural, in the discrete time case, to use  $z$ -transforms instead of Laplace-Stieltjes transforms, and (5.5.2) gives precisely the  $z$ -transforms of  $V(t)$  and  $R(t)$ , with parameter  $\rho$ . Let us denote the transforms by  $q(\rho)$  and  $r(\rho)$ .

With this notation, and using the fact that

$$(1 + \rho)^{-1} = 1 - \rho + \rho^2 - \dots,$$

Equation (5.3.2) becomes

$$[I - q(\rho)]v(\rho) = r(\rho), \quad (5.5.3)$$

where

$$q(\rho) = P - \rho P + \rho^2 P - \dots$$

and

$$r(\rho) = r - \rho r + \rho^2 r - \dots$$

Collecting terms, we get

$$v(\rho) = \rho^{-1}V_{-1} + V_0 + \rho V_1 + \dots \quad (5.5.4)$$

where  $V_{-1}, V_0, V_1, \dots$ , are a solution of

$$\begin{pmatrix} A & 0 & \dots & & & \\ P & A & 0 & \dots & & \\ -P & P & A & 0 & \dots & \\ P & -P & P & A & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ r \\ -r \\ r \\ \vdots \end{pmatrix}, \quad (5.5.5)$$

with  $A = I - P$ , as usual. Now, left multiplying by the matrix

$$\begin{pmatrix} I & 0 & \dots & & & \\ I & I & 0 & \dots & & \\ 0 & I & I & 0 & \dots & \\ 0 & 0 & I & I & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

we get

$$\begin{pmatrix} A & 0 & \dots & & & \\ I & A & 0 & \dots & & \\ 0 & I & A & 0 & \dots & \\ 0 & 0 & I & A & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ r \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \quad (5.5.6)$$

This is identical to the recurrence (3.2.7), (3.2.8) and (3.2.9), with  $w_i = V_i$ , for  $i \geq -1$ .

Hence  $v(\rho)$  has the same expansion as in Equation (3.1.6).

In the continuous time case, we have, as in Section 5.1,

$$Q_{ij}(t) = p_{ij}(1 - e^{-\lambda_i t})$$

and

$$R_i(t) = \frac{r_i}{\lambda_i}(1 - e^{-\lambda_i t}).$$

We obtain the Laplace transforms directly as

$$q_{ij}(s) = (1 + s\tau_i)^{-1} p_{ij} = (1 - s\tau_i + s^2\tau_i^2 - \dots)p_{ij}$$

and

$$r_i(s) = (1 + s\tau_i)^{-1} \tau_i r_i = (1 - s\tau_i + s^2\tau_i^2 - \dots)\tau_i r_i,$$

where  $\tau_i = 1/\lambda_i$ . Using matrix notation, with  $T = \text{diag}(\tau_i)$ , we have

$$q(s) = P - sTP + s^2T^2P - \dots \quad (5.5.7)$$

and

$$r(s) = Tr - sT^2r + s^2T^3r - \dots, \quad (5.5.8)$$

from which Equation (5.3.2) leads to

$$\begin{pmatrix} A & 0 & \dots & & & \\ TP & A & 0 & \dots & & \\ -T^2P & TP & A & 0 & \dots & \\ T^3P & -T^2P & TP & A & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ Tr \\ -T^2r \\ T^3r \\ \vdots \end{pmatrix}. \quad (5.5.9)$$

Left multiplying by the following matrix

$$\begin{pmatrix} I & 0 & \dots & & & \\ T & I & 0 & \dots & & \\ 0 & T & I & 0 & \dots & \\ 0 & 0 & T & I & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

we get

$$\begin{pmatrix} A & 0 & \dots & & & \\ T & A & 0 & \dots & & \\ 0 & T & A & 0 & \dots & \\ 0 & 0 & T & A & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} V_{-1} \\ V_0 \\ V_1 \\ V_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ Tr \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \quad (5.5.10)$$

By multiplying every row of the system (5.5.10) by  $T^{-1}$ , we obtain precisely the recurrence (4.3.3), (4.3.4) and (4.3.5), with  $w_i = V_i$ , for  $i \geq -1$ .

Consider now an arbitrary semi-Markov reward process. Let  $P = Q_0$  and  $H = Q_1$  be the zeroth and first moments of  $Q(t)$ , respectively, and let  $T$  be a diagonal matrix such that  $He = Te$ , where  $e$  is the column vector with all entries equal to 1. Then the diagonal entries of  $T$  are the expected transition times, that is,  $T = \text{diag}(\tau_i)$ .

**Lemma 5.5.1.** Suppose  $P$  is an irreducible, stochastic matrix. Then  $P^*HP^* = P^*TP^*$ .

**Proof.** Because  $P^* = e\pi$ , where  $\pi$  is the row vector of stationary probabilities of  $P$ .

Then

$$HP^* = He\pi = Te\pi = TP^*,$$

and the result follows. □

**Lemma 5.5.2.** Suppose  $P$  is a stochastic matrix. Then  $P^*HP^* = P^*TP^*$ .

**Proof.** Proceed as in the proof of Lemma 5.2.2, noting that  $P^*HP^*$  and  $P^*TP^*$  do not depend on the transient parts of  $H$  and  $T$ . □

Suppose  $\tau_i > 0$ , for  $i = 1, \dots, n$ . Then  $T$  is nonsingular. Consider a continuous time reward process with expected transition times  $\tau_i$  and reward rates  $r_i$ , for  $i = 1, \dots, n$ , where  $R_0 = T\mathbf{r}$  is the zeroth moment of  $R(t)$ .

**Theorem 5.5.3.** Let  $g_s$  be the gain of the semi-Markov process and let  $g_e$  be the gain of the continuous time process. Then  $g_s = g_e$ .

**Proof.** Let  $h_s$  and  $h_e$  be the respective bias vectors. Then Theorem 5.3.4 implies that

$$\begin{pmatrix} A & 0 \\ H & A \end{pmatrix} \begin{pmatrix} g_s \\ h_s \end{pmatrix} = \begin{pmatrix} 0 \\ T\mathbf{r} \end{pmatrix} \quad (5.5.11)$$

and

$$\begin{pmatrix} A & 0 \\ T & A \end{pmatrix} \begin{pmatrix} g_e \\ h_e \end{pmatrix} = \begin{pmatrix} 0 \\ T\mathbf{r} \end{pmatrix}. \quad (5.5.12)$$

Now decompose  $A$  as in Equation (1.3.1), and write

$$H = S^{-1} \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} S \text{ and } T = S^{-1} \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} S.$$

Lemma 5.5.2 implies that  $H_{22} = T_{22}$ . Both systems (5.5.11) and (5.5.12) have the same form as (5.2.3) and (5.2.4), with  $b = 0$ . This implies that  $x_1 = 0$ , by Equation (5.2.5), and that  $g_s$  does not depend on  $H_{21}$  (and  $g_e$  does not depend on  $T_{21}$ ). The result follows because  $H_{22} = T_{22}$ . □

**Corollary 5.5.4.** The higher order terms (e.g., bias  $h_s$  and  $h_e$ , etc.) are not equal in general.

**Proof.** Because they depend on  $H_{11}$ ,  $H_{12}$  and  $H_{21}$ , which are not necessarily equal to  $T_{11}$ ,  $T_{12}$  and  $T_{21}$ . Also, the higher moments of both  $Q(t)$  and  $R(t)$  may be different, as well. □

The significance of Theorem 5.5.3 is that when only the average reward per unit of time (i.e., the gain) of a semi-Markov reward process is required (like for finding a gain-optimal policy in a semi-Markov decision process), then it is not necessary to compute the decomposition of  $H$  as in Lemma 5.2.6. Rather, we can simply solve the continuous time problem, which can be done using the same algorithm as for a discrete time process.

## Chapter VI

### Stationary characteristics of tandem queues

Consider a queueing system with two service stations in series (i.e., in tandem). New jobs entering the system arrive at station 1. After a job has received service at station 1, it moves on to station 2. Jobs leave the system after completing service at station 2.

The waiting rooms are finite. There can be at most  $N_1$  jobs at station 1 and  $N_2$  at station 2 (including the jobs that are being served), with  $N_1 \geq 1$  and  $N_2 \geq 1$ . The blocking protocol is as follows. Let  $n_1$  be the number of jobs at station 1, and  $n_2$  at station 2. A new job arriving at station 1 when  $n_1 = N_1$  leaves the system immediately, without requiring any service. A job completing service at station 1 when  $n_2 = N_2$  returns to station 1 instantly, where it is treated as a new arrival.

All jobs are identical and enter the system according to a Poisson process with parameter  $\lambda$ . A job requests a random amount of service with mean  $\tau_1$  at station 1 and  $\tau_2$  at station 2. These service requirements are mutually independent. They are also independent of those of other jobs and of the arrival process.

Each station has a total service capacity (or service rate) of one unit of service per unit of time. The service discipline at station  $j$ , for  $j = 1, 2$ , is determined by the functions  $\gamma_j$  and  $\delta_j$  as follows [see Disney and Konig (1985), Section III.2].

Suppose there are  $n_j$  jobs at station  $j$ , with  $1 \leq n_j \leq N_j$ . Then  $\gamma_j(n_j, \ell)$  is the speed of service given to the job at position  $\ell$  at station  $j$ , with  $1 \leq \ell \leq n_j$ . When a job leaves station  $j$  from position  $\ell$ , the jobs in positions  $\ell + 1, \dots, n_j$  move to positions

$\ell, \dots, n_j - 1$ . When a job enters station  $j$  with  $n_j$  jobs already present, it moves into position  $\ell$  with probability  $\delta_j(n_j + 1, \ell)$ , with  $1 \leq \ell \leq n_j + 1$ , and the jobs in positions  $\ell, \dots, n_j$  move to  $\ell + 1, \dots, n_j + 1$ .

The functions satisfy  $\sum_{\ell=1}^{n_j} \gamma_j(n_j, \ell) = 1$  and  $\sum_{\ell=1}^{n_j} \delta_j(n_j, \ell) = 1$ . Three service disciplines of particular interest are First-In-First-Out (FIFO), Last-In-First-Out (LIFO) preemptive and processor sharing. Their functions can be expressed as follows.

$$\text{FIFO:} \quad \gamma_j(n_j, 1) = 1, \quad \delta_j(n_j, n_j) = 1;$$

$$\text{LIFO preemptive:} \quad \gamma_j(n_j, 1) = 1, \quad \delta_j(n_j, 1) = 1;$$

$$\text{Processor sharing:} \quad \gamma_j(n_j, \ell) = 1/n_j, \quad \delta_j(n_j, \ell) = 1/n_j, \quad \ell = 1, \dots, n_j.$$

When the service requirements are exponentially distributed, all three disciplines produce the same queue length process.

In the rest of this chapter, we will be interested in obtaining some stationary characteristics of tandem systems. First, in Section 6.1 we review the case in which the service requirements have an exponential distribution. In this case, we write the global balance equations using matrix notation.

Next, we use the results of Chapters I and II to analyze two numerical procedures for computing the stationary probabilities of the queue length process. In Sections 6.2 and 6.3, we consider the method of successive approximations, or power method [see e.g., Stewart (1978)], and the matrix decomposition method of Wong et al. (1977). Our original contribution is a proof of convergence of the former, taking into account the periodicity of the transition matrix, and a justification of the latter from the point of view of numerical stability.

In Section 6.4, we approach the solution of the stationary probabilities from the point of view of job-local-balance (JLB). The notion of local balance was studied by several authors, including Whittle (1968) and (1969), Kingman (1969), Baskett et al. (1975), Barbour (1976), Chandy et al. (1977), Schassberger (1978) and Hordijk and Van Dijk (1981), (1982), (1983a) and (1983b).

With the blocking protocol mentioned above, job-local-balance does not hold for our tandem system. Some modifications based on results in Hordijk and Van Dijk (1981) give two other queueing systems which satisfy JLB.

The stationary probabilities of the modified systems are then expressed by a simple, explicit formula which is easy to compute. It is the object of Section 6.5 to provide some new numerical evidence that the call congestion (i.e., the probability that a new job is rejected) of the modified systems provide a lower and an upper bound on the call congestion of the original system.

While a formal proof of the validity of the bounds, under certain conditions, can also be found in van Dijk and Lamond (1985), it was decided not to include it here, because the technique of proof bears little relationship with the methods discussed in this thesis.

### 6.1. Global balance equations

Let  $N_j(t)$  be the number of jobs at station  $j$  at time  $t$ , with  $j = 1, 2$  and  $t \geq 0$ . Then  $\{N_1(t), N_2(t) : t \geq 0\}$  is the *queue length process* of the system. Suppose that the service requirements have an exponential distribution, with parameters  $\mu_1 = (\tau_1)^{-1}$  and  $\mu_2 = (\tau_2)^{-1}$ .

We assume that the queue length process is *stationary*, that is, its distribution does not depend on  $t$ . Let  $s = (n_1, n_2)$ . If  $N_1(t) = n_1$  and  $N_2(t) = n_2$ , for some  $t \geq 0$ , then we say that the system is in state  $s$  at time  $t$ . The process  $\{N_1(t), N_2(t) : t \geq 0\}$  is a continuous time Markov chain [see Chapter IV] with state space  $S = \{s = (n_1, n_2) : 0 \leq n_i \leq N_i, i = 1, 2\}$ . Our model is the same as in Wong et al. (1977), with  $N_1$  and  $N_2 + 1$  seats.

The transition rate from a state  $r \in S$  to another state  $s \in S$ , denoted by  $R(r, s)$ , is given in Table 6.1.1. Let us also define  $B(s)$  as the total rate out of state  $s$ . Then for  $s = (n_1, n_2)$  we have

$$B(s) = \lambda a_0(s) + \mu_1 a_1(s) + \mu_2 a_2(s)$$

where  $a_k(s) = 0$ ,  $k = 0, 1, 2$ , except for

$$\begin{aligned} a_0(n_1, n_2) &= 1 & \text{if} & & n_1 < N_1, \\ a_1(n_1, n_2) &= 1 & \text{if} & & n_1 > 0 \text{ and } n_2 < N_2, \\ a_2(n_1, n_2) &= 1 & \text{if} & & n_2 > 0. \end{aligned}$$

to state	condition	rate
$(n_1 + 1, n_2)$	$(n_1 < N_1)$	$\lambda$
$(n_1 - 1, n_2 + 1)$	$(n_1 > 0) \text{ and } (n_2 < N_2)$	$\mu_1$
$(n_1, n_2 - 1)$	$(n_2 > 0)$	$\mu_2$

**Table 6.1.1. Transition rates out of state  $(n_1, n_2)$**

We are interested in the stationary probabilities  $p(s)$ ,  $s \in S$ , such that

$$p(s) = \lim_{t \rightarrow \infty} \text{Prob}\{\text{system is in state } s \text{ at time } t\}.$$

Those probabilities are the unique solution of the global balance equations

$$p(s)B(s) = \sum_{r \in S} p(r)R(r, s), \quad s \in S \quad (6.1.1)$$

together with the normalizing condition

$$\sum_{s \in S} p(s) = 1. \quad (6.1.2)$$

Now let  $N = (N_1 + 1) \times (N_2 + 1)$ . Then (6.1.1) and (6.1.2) give a system of  $N + 1$  linear equations in  $N$  unknowns. We will see, below, that one of the balance equations (6.1.1) is redundant, and that  $p(s)$  is uniquely determined by the remaining  $N$  linear equations.

But let us first look at the problem using matrix notation. Let us assume that the states  $s \in S$  are ordered in some way (e.g., lexicographically). Then  $p$  can be seen as

a row vector,  $B$  as a diagonal matrix with diagonal entries  $B(s)$ , and  $R$  as an  $N \times N$  matrix with entries  $R(r, s)$ . The global balance equations (6.1.1) can then be written as

$$pB = pR \quad (6.1.3)$$

and the normalizing condition (6.1.2) says that  $p$  is a probability vector. We can rewrite (6.1.3) as

$$p(B - R) = 0 \quad (6.1.4)$$

which can be transformed into

$$\pi(I - P) = 0 \quad (6.1.5)$$

where

$$\pi = pB / (pBe) \quad (6.1.6)$$

and

$$P = B^{-1}R.$$

This property was used in the proof of Lemma 4.2.1, where we used the time matrix  $T$  instead of the rate matrix  $B$ . Here  $e$  is just a column vector with all entries equal to 1, and the denominator of (6.1.6) ensures that  $\pi$  is also a probability vector. Note that  $B$  is a diagonal matrix, so that its inverse  $B^{-1}$  is easily computed. We will take advantage of this fact in formulating the power method, in the next section.

Observe also that  $Q = B - R$  is the infinitesimal generator of the queue length process. Moreover,  $P$  is the transition matrix of the embedded Markov chain [Ross (1970), chapter 5], for transitions out of a state, scaled to unit transition times.

## 6.2. Computation of the stationary probabilities

We review one method for solving Equation (6.1.5), the power method. In Section 6.3, another method for solving (6.1.4), the method of Wong et al. (1977), will be reviewed as well. Several other numerical methods for computing the stationary probabilities of Markov chains are described in Stewart (1978), Kaufman (1983) and Harrod and Plemmons (1984).

The principal advantage of the power method, for solving the tandem queue problem, is its simplicity. It is easier to implement than other, faster converging, iterative methods. The method of Wong et al. (1977), on the other hand, takes directly advantage of the special structure of the infinitesimal generator  $Q = B - R$  of Equation (6.1.4). It effectively reduces the original  $N \times N$  problem [with  $N = (N_1 + 1)(N_2 + 1)$ ] to a  $(2N_1 + 2) \times (2N_1 + 2)$  problem. Its implementation, however, is significantly more difficult than the power method, because it requires the computation of a Jordan canonical form. All numerical results in this chapter were obtained with the power method.

Let us analyze the solution of Equation (6.1.5) by the power method. We will proceed to compute  $\pi$ , from which we obtain  $p$  simply as

$$p = \pi B^{-1} / (\pi B^{-1} e).$$

The advantage of solving (6.1.5) instead of (6.1.4) is that  $\pi$  can be seen as the left hand eigenvector of  $P$  associated with the eigenvalue 1. As we will see, this eigenvector can be computed iteratively using the power method [see Wilkinson (1965)].

**Lemma 6.2.1.** The embedded Markov chain (i.e., with transition matrix  $P$ ) is ergodic (i.e.,  $P$  is irreducible).

**Proof.** We have to show that when the system is in any state  $r \in S$ , it can reach any other state  $s \in S$  with positive probability, in a finite number of steps. This is the case when  $\lambda$ ,  $\mu_1$  and  $\mu_2$  are positive (as we assume), because it is possible, with an appropriate order of arrivals and completions, to reach any state  $s = (n_1, n_2) \in S$  from state  $(0,0)$  with  $n_1 + n_2$  arrivals and  $n_2$  service completions at station 1. Similarly, the system can return to state  $(0,0)$  with  $n_1$  service completions at station 1 and  $n_1 + n_2$  service completions at station 2. □

**Theorem 6.2.2.** There is a unique probability vector that is a solution of equation (6.1.5).

**Proof.** Since the embedded chain is ergodic, this follows from Theorem 5.1.2 of Kemeny and Snell (1960). [Also from our Lemmas 2.2.1 and 2.2.5]. □

We know from Corollary 2.4.5 that the power method converges to the stationary probability vector when  $P$  is aperiodic. We proceed to show that for the tandem queue problem, the transition matrix  $P$  is periodic, with period 3. Let us classify the states into the following three sets:

$$S_m = \{s = (n_1, n_2) \in S : (n_1 + 2n_2) \bmod 3 = m\} \quad m = 0, 1, 2. \quad (6.2.1)$$

**Lemma 6.2.3.** The embedded Markov chain is periodic with period 3.

**Proof.** We show that all the possible transitions out of a state  $r \in S_k$  are into some states  $s \in S_m$ , where  $m = (k + 1) \bmod 3$ . Let  $r = (n_1, n_2)$ , then  $k = (n_1 + 2n_2) \bmod 3$ , according to (6.2.1). There are at most three possible transitions, as in Table 6.1.1:

$$\text{if } n_1 < N_1 \quad \text{then} \quad m = [(n_1 + 1) + 2n_2] \bmod 3;$$

$$\text{if } n_1 > 0 \quad \text{and} \quad n_2 < N_2 \quad \text{then} \quad m = [(n_1 - 1) + (2(n_2 + 1))] \bmod 3;$$

$$\text{if } n_2 > 0 \quad \text{then} \quad m = [n_1 + 2(n_2 - 1)] \bmod 3.$$

It is easy to verify that  $m = (k + 1) \bmod 3$  in all three cases. □

**Lemma 6.2.4.**

$$\sum_{s \in S_k} \pi(s) = \frac{1}{3} \quad k = 0, 1, 2.$$

**Proof.** . Because the embedded Markov chain spends exactly one third of the time in any class  $S_k$ . □

Following Varga (1962), let us assume that the states are ordered such that  $S_0$  comes first, then  $S_1$  and then  $S_2$ . We have

$$P = \begin{pmatrix} 0 & P_{01} & 0 \\ 0 & 0 & P_{12} \\ P_{20} & 0 & 0 \end{pmatrix}$$

and

$$\pi = (\pi_0 \quad \pi_1 \quad \pi_2).$$

Then

$$P^3 = \begin{pmatrix} T_0 & 0 & 0 \\ 0 & T_1 & 0 \\ 0 & 0 & T_2 \end{pmatrix}$$

where

$$T_0 = P_{01}P_{12}P_{20}, \quad \text{etc.}$$

**Lemma 6.2.5.** Each  $T_k$ ,  $k=0,1,2$ , is an irreducible, aperiodic transition matrix.

**Proof.** See e.g., Varga (1962), page 44, exercise 4. [See also the proof of our Theorem 2.2.9]. □

**Lemma 6.2.6.** For  $k=0,1,2$ , let  $q_k$  be the stationary probability vector of  $T_k$ . Then  $\pi_k = \frac{1}{3}q_k$ .

**Proof.** From equation (6.1.5), we have  $\pi = \pi P$ , and hence  $\pi = \pi P^3$ . But then  $\pi_k = \pi_k T_k$ , and  $\pi_k$  has weight  $\frac{1}{3}$ , by Lemma 6.2.4. □

**Theorem 6.2.7.** For  $k=0,1,2$ , let  $q_k^{(0)}$  be a probability vector, and  $q_k^{(n)} = q_k^{(0)} T_k^n$ . Then

$$q_k = \lim_{n \rightarrow \infty} q_k^{(0)} T_k^n$$

is the stationary distribution of  $T_k$ . [See Kemeny and Snell (1960), Theorem 4.1.6 for the proof, and also our Corollary 2.4.5].

**Corollary 6.2.8.** Let  $\pi^{(n)} = \pi^{(0)} P^n$ . Then

$$\pi = \lim_{n \rightarrow \infty} \pi^{(n)}$$

if and only if

$$\sum_{s \in S_k} \pi^{(0)}(s) = \frac{1}{3}, \quad k = 0, 1, 2. \quad (6.2.2)$$

**Proof.** Suppose that  $\sum_{s \in S_k} \pi^{(0)}(s) = w_k$ , for some arbitrary numbers  $w_k$ ,  $k=0,1,2$ .

Then

$$\lim_{n \rightarrow \infty} \begin{pmatrix} \pi^{(3n)} \\ \pi^{(3n+1)} \\ \pi^{(3n+2)} \end{pmatrix} = \begin{pmatrix} w_0 q_0 & w_1 q_1 & w_2 q_2 \\ w_2 q_0 & w_0 q_1 & w_1 q_2 \\ w_1 q_0 & w_2 q_1 & w_0 q_2 \end{pmatrix}$$

and the result follows from Lemma 6.2.6. □

The above results indicate that the stationary probabilities can be computed iteratively, using the power method [Wilkinson (1965), page 570], which we can formulate as in Figure 6.2.1.

Step 0: Take an arbitrary probability vector  $\pi^{(0)}$  satisfying (6.2.2),  
 set  $k = 0$  and specify  $\epsilon > 0$ .

Step 1: Compute  $p^{(k)} = \pi^{(k)} B^{-1}$ .

Step 2: Compute  $\pi^{(k+1)} = p^{(k)} R$ .

Step 3: If  $\|\pi^{(k+1)} - \pi^{(k)}\| \leq \epsilon$  for some appropriate norm (e.g., 1, 2,  $\infty$ )  
 then stop  
 else increment  $k$  and go to Step 1.

**Figure 6.2.1. The power method for stationary probabilities**

The sequence  $\{\pi^{(k)}, k = 0, 1, 2\}$  converges to the stationary probability vector  $\pi$ , by Corollary 6.2.8, and the algorithm is extremely easy to implement, using procedures like those shown in Figure 6.2.2. (Note that there are at most three multiplications for every state, in one iteration).

The performance of the power method, then, is determined by the speed at which the sequence  $\pi^{(n)}$  approaches its limit  $\pi$ . Let  $\gamma_k$  be the modulus of the subdominant eigenvalue of  $T_k$ , for  $k = 0, 1, 2$ . By Theorem 6.2.7 and Corollary 2.4.7, we see that each of the subsequences  $\pi^{(3n)}, \pi^{(3n+1)}$  and  $\pi^{(3n+2)}$  approaches  $\pi$  geometrically, with parameter  $\gamma_0, \gamma_1$  and  $\gamma_2$ , respectively.

Hence at each step of the power method, the error is reduced by a factor

$$\gamma = (\max\{\gamma_0, \gamma_1, \gamma_2\})^{1/3}$$

in average, when the number of iterations  $n$  is large.

Another important consideration, when using the power method, is how to select the parameter  $\epsilon$  for the stopping criterion, at Step 3 of Figure 6.2.1. We show, as in Section 3.4, that if  $\delta$  is the accuracy required on  $\pi$ , then we should select  $\epsilon \leq \delta/\|A^\#\|$ , where  $A^\#$  is the Drazin inverse of  $A = I - P$ .

```

CONST DIM1=100; DIM2=100; (* storage dimensions *)
TYPE STATE1=0..DIM1; STATE2=0..DIM2;
    DISTRIBUTION=ARRAY[STATE1,STATE2] OF REAL;
VAR N1:STATE1; N2:STATE2;    (* server capacities *)
    RATE0,RATE1,RATE2:REAL; (* arrival, service rates *)
    P:DISTRIBUTION          (* p vector *)

FUNCTION OUTRATE(I:STATE1;J:STATE2):REAL;
(* Total rate of transition out of a state *)
VAR RATE:REAL;
BEGIN (* OUTRATE *)
RATE:=0;
IF (I<N1) THEN RATE:=RATE+RATE0;
IF ((I>0) AND (J<N2)) THEN RATE:=RATE+RATE1;
IF (J>0) THEN RATE:=RATE+RATE2;
OUTRATE:=RATE
END (* OUTRATE *);

FUNCTION INFLOW(I:STATE1;J:STATE2;
                VAR P:DISTRIBUTION):REAL;
(* total flow of transitions into a state *)
VAR FLOW:REAL;
BEGIN (* INFLOW *)
FLOW:=0;
IF (I>0) THEN FLOW:=FLOW+RATE0*P[I-1,J];
IF ((I<N1) AND (J>0)) THEN FLOW:=FLOW+RATE1*P[I+1,J-1];
IF (J<N2) THEN FLOW:=FLOW+RATE2*P[I,J+1];
INFLOW:=FLOW
END (* INFLOW *);

```

**Figure 6.2.2. Pascal procedures for transition rates**

We see this as follows. Let  $\hat{\pi}$  be a computed solution, with error  $e = \hat{\pi} - \pi$ . Define the residual error  $r = \hat{\pi}(I - P)$ .

**Theorem 6.2.9.**  $e = A\#r$ .

**Proof.** Let  $u$  be a row vector such that  $u_i = 1/N$ . Then  $\pi$  is the unique solution of

the evaluation equations of dynamic programming:

$$\pi A = 0$$

$$\pi + hA = u,$$

where  $h$  stands as the "bias" vector. The result follows immediately from Theorem 3.4.4. □

Now by taking norms, we have

$$\|e\| \leq \|A^\# \| \cdot \|r\|.$$

Hence at Step 3, when we test for  $\|r\| \leq \epsilon$ , we have

$$\|e\| \leq \|A^\# \| \cdot \epsilon \leq \|A^\# \| \cdot \delta / \|A^\# \| = \delta.$$

It is our observation that  $\|A^\# \|_2 < 1000$  for a wide range of the parameters  $\lambda$ ,  $\mu_1$ ,  $\mu_2$ ,  $N_1$  and  $N_2$ . Interestingly, the rate of convergence varies quite strongly. With an accuracy  $\epsilon = 10^{-9}$ , the number of iterations varied from 50 to 3000 iterations, and it is not obvious how the parameters affect the rate of convergence. For most of the curves we produced, in which every point was evaluated using the latest  $\pi$  vector available, the average number of iterations was between 20 and 50. We did not observe that the buffer sizes  $N_1$  and  $N_2$  were a determinant factor of the speed of convergence (in terms of iterations required). This does not support an observation of Stewart (1978) that large waiting rooms imply slow convergence.

### 6.3. The matrix decomposition method

In this section, we describe a method due to Wong et al. (1977), which exploits the special structure of the infinitesimal generator. The method is related to the matrix decomposition approach of the previous chapters and uses the Jordan canonical form of a certain matrix.

We briefly review the method and formulate it in algorithmic form. Certain aspects of the method are clarified and we give a justification of the use of the Jordan canonical form, from the point of view of numerical stability. We have no computational experience to report, about its performance.

The method of Wong et al. (1977) produces a stationary probability vector by direct solution of a singular system of  $N_1 + 1$  equations. Let us use the two parameters  $\rho = \lambda/\mu_2$  and  $\alpha = \mu_1/\mu_2$ . Also, assume that the states are sorted in reverse lexicographic order [i.e.,  $p(0,0), p(1,0), \dots, p(N_1,0), p(0,1), \dots, p(N_1, N_2)$ ]. Note: the order shown on top of Page 5 of Wong et al. (1977) is incorrect. (The error has no effect on their development, however).

Then the matrix  $Q = B - R$  of Equation (6.1.4) has a block-tridiagonal structure.

We have

$$Q = \begin{pmatrix} -A_1 & \alpha L & & & \\ I & -A_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & -A_2 & \alpha L \\ & & & I & -A_3 \end{pmatrix}, \quad (6.3.1)$$

where

$$A_1 = \begin{pmatrix} \rho & -\rho & & & \\ & \rho + \alpha & \ddots & & \\ & & \ddots & \ddots & \\ & & & \rho + \alpha & -\rho \\ & & & & \alpha \end{pmatrix},$$

$A_2 = A_1 + I$  and  $A_3 = A_2 - \alpha H$ , with  $I$  the identity matrix,  $H$  a diagonal matrix with all ones on the main diagonal except  $h_{11} = 0$ , and  $L$  a matrix with all ones on the subdiagonal and all other entries zero.  $A_1, A_2, A_3, I, H$  and  $L$  are all  $(N_1 + 1) \times (N_1 + 1)$  matrices and  $Q$  is an  $N \times N$  matrix, where  $N = (N_1 + 1)(N_2 + 1)$ . [Hence  $Q$  has  $(N_2 + 1) \times (N_2 + 1)$  blocks].

Partition the stationary probability vector  $p$  as  $p = (p_0, p_1, \dots, p_{N_2})$ , where  $p_n = (p(0, n), \dots, p(N_1, n))$ , for  $n = 0, \dots, N_2$ . Let also  $x_n = (p_n, p_{n+1})$ , for  $n = 0, \dots, N_2 - 1$ . Then all the stationary probabilities can be expressed in terms of  $p_0$ , using the  $(2N_1 + 2) \times (2N_1 + 2)$  matrix  $B$  defined as

$$B = \begin{pmatrix} 0 & -\alpha L \\ I & A_2 \end{pmatrix}. \quad (6.3.2)$$

Indeed, Equation (6.1.4) with  $Q$  as in (6.3.1) implies that

$$p_1 = p_0 A_1, \quad (6.3.3)$$

and that

$$x_{n+1} = x_n B, \text{ for } n = 0, \dots, N_2 - 1, \quad (6.3.4)$$

hence

$$x_n = x_0 B^n, \text{ for } n = 0, \dots, N_2. \quad (6.3.5)$$

Now consider the  $(N_1 + 1) \times (N_1 + 1)$  matrix

$$C = (I \ A_1) B^{N_2} \begin{pmatrix} -\alpha H \\ I \end{pmatrix}. \quad (6.3.6)$$

The following result is shown in Wong et al. (1977).

**Theorem 6.3.1.** The equation  $p_0 C = 0$  uniquely determines all the entries of  $p_0$  in terms of  $p(0, 0)$ .

The significance of Theorem 6.3.1 is that it reduces the problem of solving the  $(N_1 + 1)(N_2 + 1)$  equations of (6.1.4) to the much smaller problem of solving only  $N_1 + 1$  equations. This suggests an efficient algorithm to compute  $p$ , as shown in Figure 6.3.1.

- Step 1. Let  $Y_0 = (I \ A_1)$ .
- Step 2. Compute  $Y_{n+1} = Y_n B$ , for  $n = 0, \dots, N_2 - 1$ .
- Step 3. Compute  $C = Y_{N_2} \begin{pmatrix} -\alpha H \\ I \end{pmatrix}$ .
- Step 4. Let  $p(0, 0) = 1$  and solve  $p_0 C = 0$ .
- Step 5. Compute  $p_1 = p_0 A_1$ .
- Step 6. Compute  $p_{n+2} = p_{n+1} A_2 - \alpha p_n L$ , for  $n = 0, \dots, N_2 - 2$ .
- Step 7. Scale  $p$  so that (6.1.2) is satisfied and stop.

**Figure 6.3.1. Unstable algorithm for stationary probabilities**

The matrix  $C$  of Equation (6.3.6) is produced at Steps 1, 2 and 3. Then Step 4 uses the result of Theorem 6.3.1. The vector  $p_0$  can be computed by either Gaussian

elimination or some other technique for singular equations (see Chapter I). Then the other components of the vector  $p$  are computed. Step 5 applies Equation (6.3.3) and Step 6 applies Equation (6.3.4). Finally, Step 7 obtains a probability vector.

The algorithm of Figure 6.3.1 is efficient because it can be implemented with only one array of dimension  $(N_1 + 1) \times 2(N_1 + 1)$  and it takes  $3(N_1 + 1)^2(N_2 + 1)$  flops to compute the matrix  $C$ . Then Step 4 takes  $\frac{1}{3}(N_1 + 1)^3$  flops if Gaussian elimination is used. The amount of arithmetic for Steps 5, 6 and 7 is negligible. By comparison, each iteration of the power method takes  $4(N_1 + 1)(N_2 + 1)$  flops. In the case  $N_1 = N_2$ , the algorithm is then equivalent to about  $N_1 + 1$  iterations of the power method. For  $N_1$  and  $N_2$  not too large (say up to 50), our experience with the power method indicates that the algorithm of Figure 6.3.1 would usually be much faster.

This algorithm is not recommended, however, because it is numerically unstable, as preliminary tests have shown. It appears that the iteration with the matrix  $B$  (both at Step 2 and at Step 6) is inaccurate. The method suggested by Wong et al. (1977) [Note: the numerical instability problem was not mentioned by these authors] works directly with the powers of  $B$ , in Equations (6.3.5) and (6.3.6). Those are obtained directly from the Jordan canonical form of  $B$ . The following result was obtained in Wong et al. (1977). and Disney and Chandramohan (1980). Note that their proof is similar to that of a related well known result concerning the eigenvalues of a symmetric tridiagonal matrix, based on the Sturm sequence property [see Wilkinson (1965)].

**Theorem 6.3.2.** There is a nonsingular matrix  $S$  such that

$$B = S \begin{pmatrix} J_M(0) & 0 \\ 0 & D \end{pmatrix} S^{-1}, \quad (6.3.7)$$

where  $J_M(0)$  is the Jordan submatrix of order  $M$  with eigenvalue 0, with  $M = \lfloor (N_1 + 2)/2 \rfloor$ , and  $D$  is an  $(N_1 - M) \times (N_1 - M)$  diagonal matrix with positive diagonal entries that are distinct of each other.

Assuming that the canonical form (6.3.7) is available, then the powers of  $B$  can

also be computed efficiently, so that Equations (6.3.5) and (6.3.6) become

$$x_n = x_0 S \begin{pmatrix} (J_M(0))^n & 0 \\ 0 & D^n \end{pmatrix} S^{-1} \quad (6.3.8)$$

and

$$C = (I \ A_1) S \begin{pmatrix} (J_M(0))^{N_2} & 0 \\ 0 & D^{N_2} \end{pmatrix} S^{-1}. \quad (6.3.9)$$

This leads to the algorithm of Figure 6.3.2.

- Step 1. Compute  $S$ ,  $D$ , and  $S^{-1}$  as in Equation (6.3.7).  
 Step 2. Compute  $C$  as in Equation (6.3.9).  
 Step 3. Let  $p(0,0) = 1$  and solve  $p_0 C = 0$ .  
 Step 4. Compute  $p_1 = p_0 A_1$  and let  $x_0 = (p_0 \ p_1)$ .  
 Step 5. Compute  $x_n$  as in Equation (6.3.8), for  $n = 1, \dots, N_2$ .  
 Step 6. Scale  $p$  so that Equation (6.1.2) is satisfied.

**Figure 6.3.2. Modified algorithm for stationary probabilities**

The difficult part of this algorithm is at Step 1. Although the theorem guarantees that the the eigenvalues are distinct, it does not indicate how well separated they are. Very close eigenvalues are known to cause numerical difficulties in computing the eigenvectors [see Golub and Wilkinson (1976)]. We have no computational experience to report, at the moment, concerning this problem.

#### 6.4. Job local balance and bounds

In this section, we review briefly the job-local-balance property (JLB) and show that it is not satisfied by the tandem queueing system under consideration. Certain modifications, based on results in Hordijk and Van Dijk (1981), are proposed and the

modified models are shown to satisfy JLB. It is argued intuitively that these models provide a lower and an upper bound on the call congestion of the original model.

If  $N_1 = N_2 = \infty$  then it is well known [Jackson (1957), (1963), Kingman (1969), Kelly (1979)] that for the exponential case and for any service discipline, the stationary probabilities are given by the product form (with  $c$  a normalizing constant):

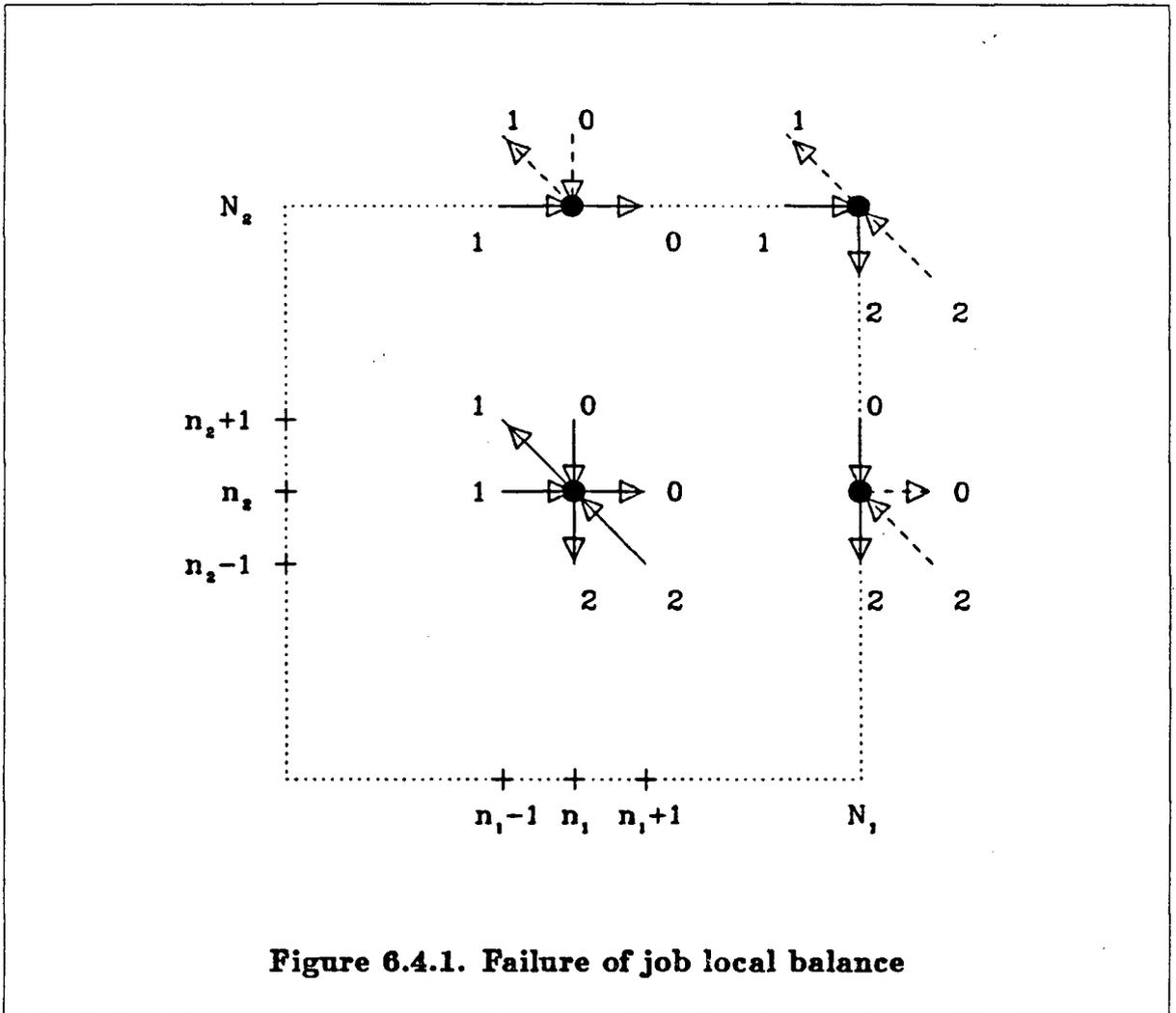
$$p(n_1, n_2) = c \left( \frac{\lambda}{\mu_1} \right)^{n_1} \left( \frac{\lambda}{\mu_2} \right)^{n_2} \quad n_1 \geq 0, n_2 \geq 0.$$

In addition, as shown by Barbour (1976), Schassberger (1978), Kelly (1979) and Hordijk and Van Dijk (1982), for a number of service disciplines such as computer sharing or a LIFO-preemptive discipline (but not a FIFO discipline), this product form solution remains valid for the non-exponential case, that is, with non-exponential service requirements. This phenomenon is known in the literature as “insensitivity”.

In different frameworks and terminology, Whittle (1968) and (1969), Kingman (1969), Baskett et al. (1975), Barbour (1976), Chandy et al. (1977), Schassberger (1978), Hordijk and Van Dijk (1982), (1983a) and (1983b), have demonstrated that product form solutions and insensitivity properties are related to a notion of partial balance. Roughly speaking, partial balance says that the global balance equations, which are necessary and sufficient for the stationarity of a probability distribution, are satisfied with a particular decomposition in local equations.

More precisely, as analyzed in Hordijk and Van Dijk (1983a), all these terminologies have the interpretation that for each job present in the system and for any system configuration, that job itself satisfies a notion of stationarity or more precisely, job-local-balance (JLB).

It is easily seen that with finite buffer sizes, JLB can not be satisfied (for instance, if the first node is saturated whereas the second node is not, then at the first node jobs can enter at a positive rate but only leave at a zero rate so that these rates can not balance). This can also be seen by looking at the transition diagram in Figure 6.4.1.



**Figure 6.4.1. Failure of job local balance**

The dashed arrows indicate the transitions which are not possible at the boundaries, thus destroying the job local balance. (An arrow is labelled  $i$  if it applies to a job at node  $i$ , for  $i = 1, 2$ , and 0 for a job outside the system).

The JLB notion, however, would guarantee the simple form of the stationary probabilities. Therefore, the purpose of this section is to find modifications of the finite tandem system which do satisfy JLB and which may provide approximations or bounds for stationary characteristics of the original model. More precisely, we concentrate on the stationary probability that an arriving job is blocked from entering the system. In communication engineering this is historically known as the "call congestion". The

modifications given are based on results given in Hordijk and Van Dijk (1981) and the intuition that they will provide a lower and an upper bound for the call congestion of the original tandem queue.

We define two modifications of the original model, from which we intuitively expect to obtain lower and upper bounds on the call congestion. These modifications, suggestively called the lower and upper bound models, are characterized as follows.

#### *Lower bound model*

A job is blocked from entering the system (rejected) only if the total number of jobs present in the system is equal to  $N_1 + N_2$ . A job which completes its service at node 1 is always allowed to enter node 2 instantly.

#### *Upper bound model*

In addition to the blocking protocols of the original model, a job is also blocked from entering the system (rejected) if  $n_2 = N_2$  and a job which completes its service at node 2 is prohibited from leaving the system if  $n_1 = N_1$ , in which case it must return in the queue for a new service at node 2.

It is intuitively clear that the lower bound model should have a lower blocking probability than the original model, since it makes a more efficient use of the  $N_1 + N_2$  available places, by sharing them among the two stations. Similarly, the upper bound model would be expected to have more congestion, since it has more stringent conditions for acceptance of a new job and it sometimes gives extra service to a completed job.

As shown in the above reference, job-local-balance is closely related to the algorithmic computation of the stationarity probabilities. In the special case of a standard Jackson type network, this algorithmic computation simply leads to the product form probabilities. By using a notion equivalent to JLB, it has been shown in Sections 7 and 8 of Hordijk and Van Dijk (1981) that the stationary probabilities of the lower and upper bound models, in the exponential case, are given by  $P_L$  and  $P_U$  as follows.

With  $c_1$  and  $c_2$  normalizing constants:

$$P_L(n_1, n_2) = \begin{cases} c_1(\lambda\tau_1)^{n_1}(\lambda\tau_2)^{n_2} & \text{if } 0 \leq n_1; 0 \leq n_2; n_1 + n_2 \leq N_1 + N_2 \\ 0 & \text{otherwise.} \end{cases} \quad (6.4.1)$$

$$P_U(n_1, n_2) = \begin{cases} c_2(\lambda\tau_1)^{n_1}(\lambda\tau_2)^{n_2} & \text{if } 0 \leq n_j \leq N_j; n_1 + n_2 < N_1 + N_2 \\ 0 & \text{otherwise.} \end{cases} \quad (6.4.2)$$

Although these probabilities exhibit a product form, it must be realized that the use of the expression "product form" is not fully justified because of the constraints of the admissible regions. Clearly, afterwards, the stationarity can just as well be verified by checking the global balance equations.

The JLB analysis has been instrumental, however, in the derivation of the above models as potential candidates. The lower bound model was derived by modifying the original model so that the dotted arrows of Figure 6.4.1 be included. On the other hand, the upper bound model was derived so that the transitions which are not locally balanced be excluded.

Furthermore, in addition to an algorithmic computation of the stationary probabilities, job-local-balance is associated with the insensitivity property [cf. Barbour (1976), Schassberger (1978) and Hordijk and Van Dijk (1982) and (1983b)]. Under this property, the stationary probabilities depend on the service distributions only through their means. Hence for service disciplines satisfying a job-local-balance condition [see the  $K_2$ -criterion in Hordijk and Van Dijk (1983a)] the probabilities given by (6.4.1) and (6.4.2) are valid as stationary probabilities regardless of the form of the service distributions. A computer sharing or LIFO-preemptive discipline satisfies such a condition. Hence, for such disciplines the bounding method may be expected to be insensitive as well, that is to provide bounds that hold for all service distributions.

From now on we use the subscript  $L$  ( $U$ ) to indicate a quantity for the lower (upper) bound model, and no subscript for the original model. Now, the stationary probabilities represent the fraction of time spent by the system in each state. Since Poisson arrivals see time averages [see Brumelle (1971) and Wolff (1982)], the call congestions are given

by

$$B = P(n_1 = N_1) \tag{6.4.3}$$

$$B_L = P_L(n_1 + n_2 = N_1 + N_2) \tag{6.4.4}$$

$$B_U = P_U(n_1 = N_1 \vee n_2 = N_2), \tag{6.4.5}$$

where  $P(\cdot)$  represents the stationary probability of an event. Based on the intuitive arguments at the beginning of this section, we may expect that

$$B_L \leq B \tag{6.4.6}$$

and

$$B \leq B_U \tag{6.4.7}$$

for the exponential case. Moreover, we would also expect the same results in the non-exponential case as well, under service disciplines satisfying JLB. Section 6 of van Dijk and Lamond (1985) contains a formal proof of (6.4.6) for the exponential case and under the condition that  $\lambda \geq \mu_2$ . It also briefly indicates a similar proof for (6.4.7) with  $\mu_1 \leq \mu_2$  and suggests a proof for the general case.

In addition to the validity of the bounds, a bounding method must also provide an indication of their accuracy. In Section 6.5, the values of  $B$ ,  $B_L$  and  $B_U$ , as given by (6.4.3), (6.4.4) and (6.4.5), will therefore be compared for a wide range of the system parameters. The numerical results also support the validity of (6.4.6) and (6.4.7), in general.

As a first illustration, let us briefly consider the case of  $\lambda = \mu_1 = \mu_2$ . Then, one easily verifies that:

$$B_L = 2/(N_1 + N_2 + 2)$$

$$B_U = (N_1 + N_2)/(N_1 N_2 + N_1 + N_2)$$

so that for  $N_1 = N_2 = N$  and with  $N$  large we observe that:  $B_U \sim 2B_L$ . Hence, very accurate bounds in percentage can not be expected. On the other hand, for  $N$  large

it holds that  $B_U \sim 0$  and the numerical results of section 6.5 will show that even on a percentage basis,  $B_L$  is then very accurate. A converse statement holds with small values of  $N$  for the upper bound. A more detailed investigation follows.

## 6.5. Performance results

### 6.5.1. Exponential case

In the case of exponential service requirements, the blocking probabilities were computed as described in Section 6.2 and compared with the lower and upper bounds, for a wide range of parameter values. Since only the proportions of  $\lambda$ ,  $\mu_1$  and  $\mu_2$  are relevant, we chose (without loss of generality)  $\lambda = 1$ . Below we will discuss our major observations as illustrated by Tables 6.5.1 and 6.5.2 and Figures 6.5.1 and 6.5.2.

**Table 6.5.1.** This tabulates the blocking probabilities for varying service speeds 0.1, 0.5 and 2.0, as well as buffer sizes (5,5), (5,10) and (10,5), in order to find out the accuracy of the bounds for large blocking probabilities, and simultaneously to observe the dominance of a bottleneck service speed and buffer size. As for the bounds, we found:

- For  $B \geq 0.9$ , the bounds are very accurate, within 4% of the correct value in the worst case.
- For  $B$  from 0.50 to 0.55, the worst case is accurate within 23% and the lower bound is always within 7% accuracy.
- For  $B \leq 0.02$ , the upper bound is less than 0.035.

Further, the following performance results appeared.

- The smallest service rate is strongly dominant; the size of the second rate is hardly of influence (e.g., cf.  $\mu_1 = 0.1, \mu_2 = 0.1$ ;  $\mu_1 = 0.1, \mu_2 = 0.5$ ).
- The smallest buffer size is strongly dominant; the size of the other buffer size is hardly of influence (e.g., cf.  $N_1 = 5, N_2 = 5$ ;  $N_1 = 5, N_2 = 10$ ).

Moreover, the different blocking levels were characterized by (recall that  $\lambda = 1$ ):

- $B \geq 0.9$  when the smallest rate is 0.1;
- $B \approx 0.5 - 0.55$  when the smallest rate is 0.5;
- $B$  drops to less than 0.02 when both rates are 2.0.

$N_1$	$N_2$	$\mu_1$	$\mu_2$	$B_L$	$B$	$B_U$
5	5	0.1	0.1	0.909184	0.916667	0.947369
			0.5	0.900000	0.900026	0.909357
		0.5	0.1	0.900000	0.900026	0.909357
			0.5	0.549973	0.584101	0.673684
		2.0	0.1	0.500366	0.508094	0.511811
			0.5	0.500366	0.500853	0.511811
			2.0	0.002694	0.017682	0.031250
			2.0	0.002694	0.017682	0.031250
5	10	0.1	0.1	0.906294	0.909091	0.947369
			0.5	0.900000	0.900001	0.909100
		0.5	0.1	0.900000	0.900000	0.909357
			0.5	0.533333	0.547311	0.670319
		2.0	0.1	0.500011	0.507937	0.508055
			0.5	0.500011	0.500018	0.504240
			2.0	0.000122	0.015905	0.016346
			2.0	0.000122	0.015905	0.016346
10	5	0.1	0.1	0.906294	0.916667	0.947369
			0.5	0.900000	0.900026	0.909357
		0.5	0.1	0.900000	0.900026	0.909100
			0.5	0.533333	0.583336	0.670319
		2.0	0.1	0.500011	0.500591	0.504240
			0.5	0.500011	0.500381	0.508055
			2.0	0.000122	0.000744	0.016346
			2.0	0.000122	0.000744	0.016346

**Table 6.5.1. Numerical values with  $\lambda = 1$ , varying  $\mu_1$  and  $\mu_2$**

**Table 6.5.2.** Throughout our computations it appeared that, as a function of the service speeds, the worst results occurred with equal parameters  $\lambda = \mu_1 = \mu_2$ . This makes sense since if one of them is significantly smaller, then that one will become the dominant bottleneck value. In Table 6.5.2, we can check the accuracy in this worst case for blocking probabilities varying from 0.6 to almost 0. We varied the buffer sizes from (1,1) up to (40,20). As for the bounds we now found:

- With  $N_1 = N_2$  and  $B \geq 0.10$ , the accuracy in both directions is within 30%.
- For  $N_1 \neq N_2$  and  $B \geq 0.10$ , the accuracy is maintained for the upper bound.

$N_1$	$N_2$	$B_L$	$B(N_1, N_2)$	$B(N_2, N_1)$	$B_U$
1	1	0.500000	0.600000	0.600000	0.666667
2	1	0.400000	0.533332	0.533334	0.600000
2	2	0.333333	0.422222	0.422222	0.500000
3	1	0.333333	0.512193	0.512196	0.571429
3	2	0.285714	0.379571	0.379579	0.454545
3	3	0.250000	0.325026	0.325026	0.400000
5	1	0.250000	0.501736	0.501743	0.545455
5	2	0.222222	0.347796	0.347809	0.411765
5	3	0.200000	0.280451	0.280480	0.347826
5	4	0.181818	0.244335	0.244373	0.310345
5	5	0.166667	0.222356	0.222356	0.285714
10	10	0.090909	0.124192	0.124192	0.166667
20	10	0.062500	0.084538	0.104327	0.130435
20	20	0.047619	0.064373	0.064373	0.090909
40	10	0.038462	0.046216	0.097846	0.111111
40	20	0.032258	0.034501	0.056293	0.069767

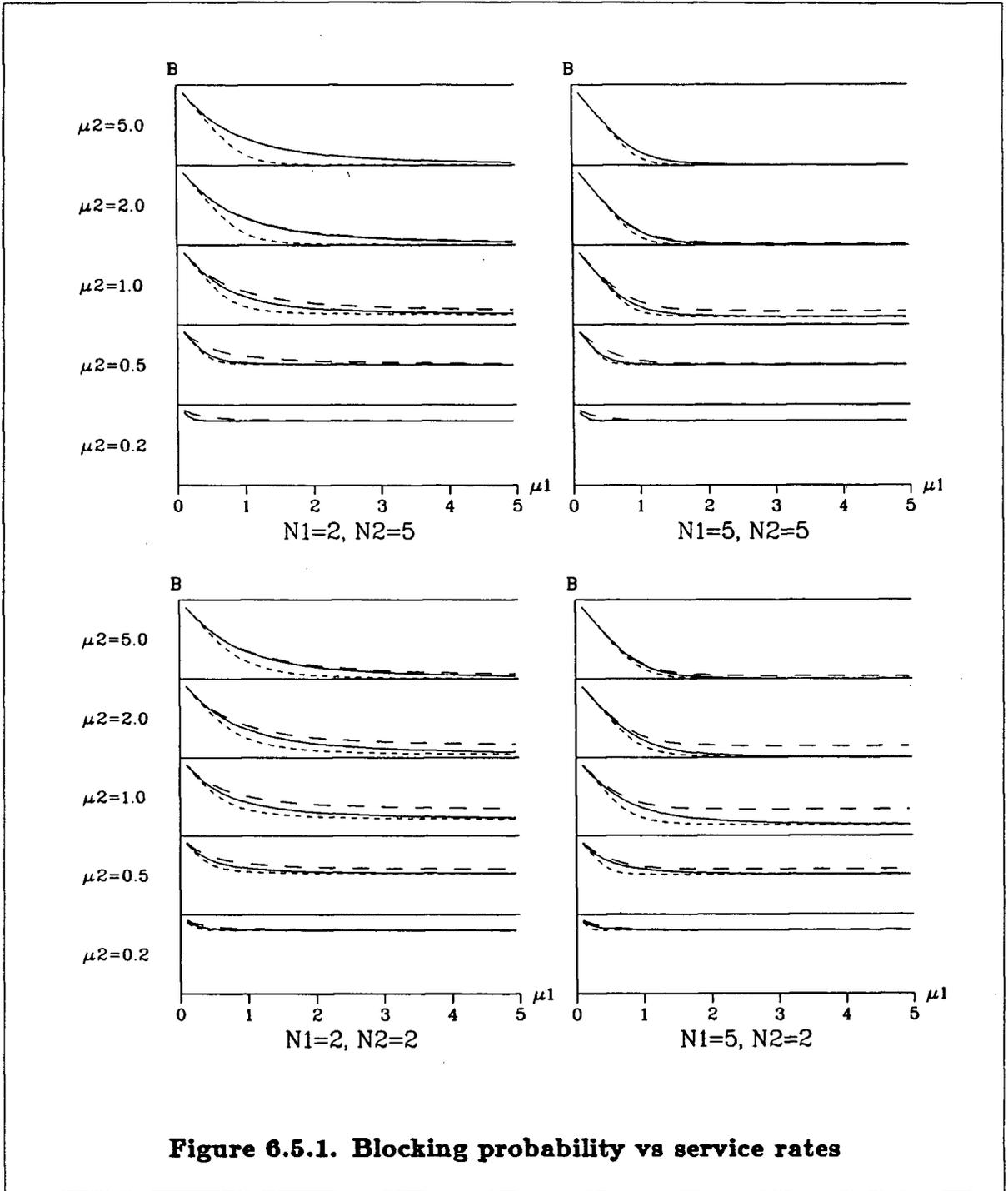
**Table 6.5.2. Numerical values with  $\lambda = \mu_1 = \mu_2 = 1$**

- For  $N_1 \neq N_2$  and  $B \leq 0.10$ , the accuracy is maintained in the bottleneck direction (that is, for the lower bound if  $N_1 > N_2$ ; for the upper bound if  $N_1 < N_2$ ).

As in Table 6.5.1, again the dominance of the smallest buffer can be observed [for instance, compare (2,1), (3,1) and (5,1)]. Further, as a particular result of the equality of  $\lambda$ ,  $\mu_1$  and  $\mu_2$ , the calculations could be restricted by using a symmetry property. This will be discussed below (see Lemma 6.5.1).

The blocking probabilities could be classified as by

- $B \leq 0.54$  if  $N_1, N_2 \geq 2$ .
- $B \leq 0.13$  if  $N_1, N_2 \geq 10$ .



**Figure 6.5.1. Blocking probability vs service rates**

**Figure 6.5.1.** Here the blocking probabilities are plotted against  $\mu_1$ , the service rate at station 1, for several values of  $\mu_2$ , the rate at station 2. The solid line represents

the blocking probability, the short dashed line is the lower bound and the dashed line is the upper bound. Those curves show the distance between the bounds and the true value. In accord with intuition, the lower bound is more accurate when the bottleneck is at station 2 (i.e., when  $N_2$  or  $\mu_2$  is small), and the upper bound is more accurate when the bottleneck is at station 1 (i.e., when  $N_1$  or  $\mu_1$  is small).

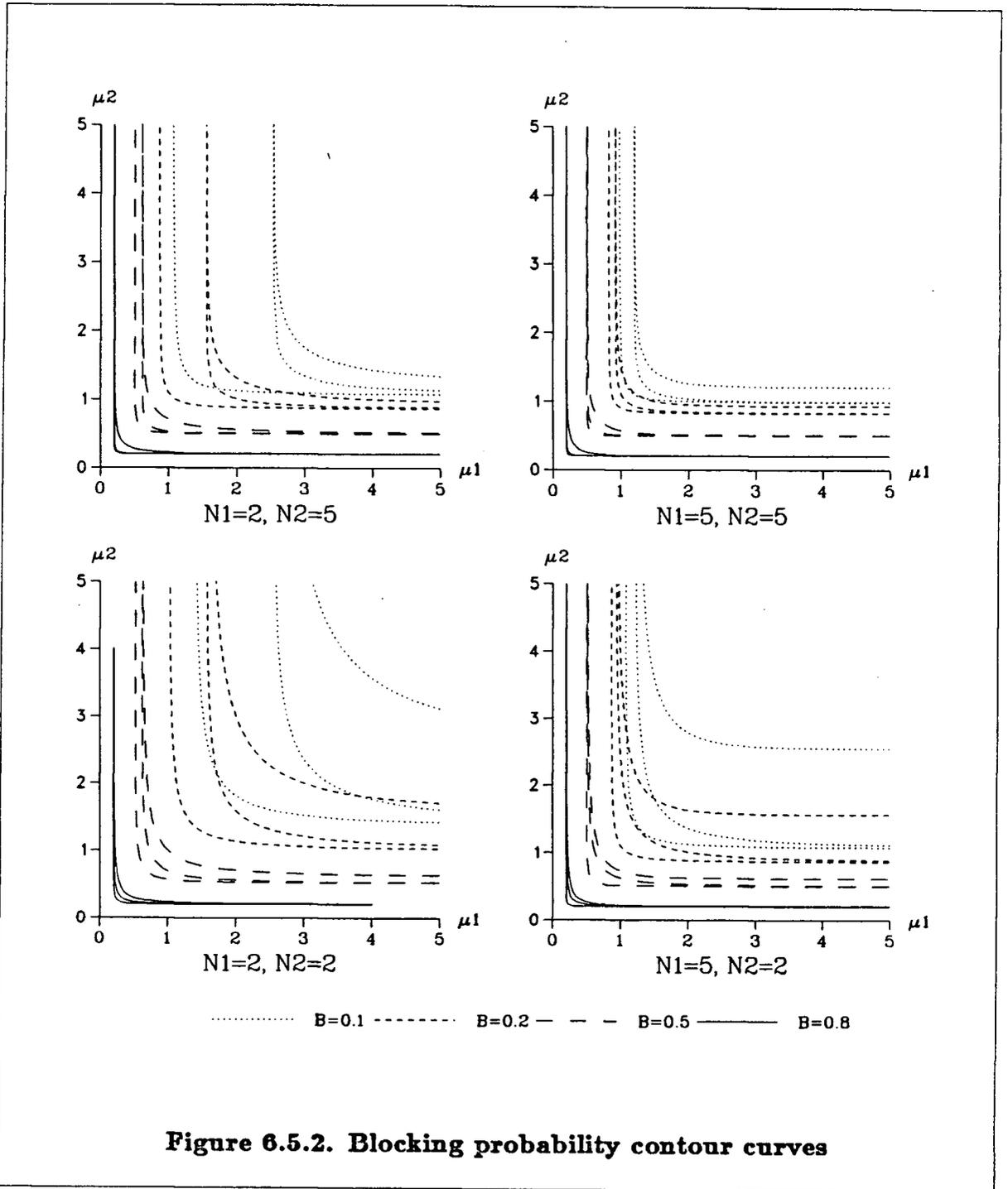
This can be argued by that in the lower bound model the blocking occurs when both stations are full, which is the case most of the time in the original model when the bottleneck is at station 2. When the bottleneck is at station 1, however, most blocking occurs when only the first station is full, because the second station is very seldom saturated, in both the original and upper bound models. Both bounds are most inaccurate when both rates are close to 1 (i.e., when  $\mu_1 \approx \mu_2 \approx \lambda$ ), where they are both in error by up to 30% approximately.

**Figure 6.5.2.** In this figure we give contour curves for fixed blocking probabilities as a function of  $\mu_1$  and  $\mu_2$ . For each value of the blocking probability, the upper curve represents the upper bound, the middle curve is the true value and the lower curve is for the lower bound. Each curve partitions the space of the service rates into two regions: points above the curve have a lower blocking probability, and points below the curve have a higher blocking probability. Hence they indicate how to select  $\mu_1$  and  $\mu_2$  when designing a system with constraints on the call congestion. Moreover, the bottleneck effect is shown more dramatically here than in Figure 6.4.1.

**Symmetry property.** In Table 6.5.2, the bounds are symmetric in  $N_1$  and  $N_2$ , because  $\mu_1 = \mu_2$ . Moreover, the blocking probabilities for the original model were obtained by evaluating the stationary probabilities only once, because of the following result. Let

$$p(n_1, n_2; N_1, N_2, \lambda, \mu_1, \mu_2)$$

be the stationary probability that the original model be in state  $(n_1, n_2)$ . We have the following symmetry.



**Lemma 6.5.1.**

$$p(n_1, n_2; N_1, N_2, \lambda, \mu_1, \mu_2) = p(N_2 - n_2, N_1 - n_1; N_2, N_1, \mu_2, \mu_1, \lambda) \quad (6.5.1)$$

**Proof.** Consider the system in which the empty seats (i.e., gaps) denote the states, instead of the number of jobs. Then whenever a job moves to the right, a gap moves to the left. Hence both systems have the same stationary probabilities.  $\square$

Table 6.5.1 shows some values with various service rates  $\mu_1$  and  $\mu_2$ , with blocking probabilities ranging from near 0 to near 1. While it seems natural to fix  $\lambda$  and vary  $\mu_1$  and  $\mu_2$ , Lemma 6.5.1 suggests that it would be more economical to fix  $\mu_1 = 1$  and vary  $\lambda$  and  $\mu_2$ . We did not, however, attempt to produce other tables using this symmetry. The idea of considering the gaps moving in the opposite direction appears to be related to the reversibility concept of Yamazaki et al. (1985), although our result is different.

### 6.5.2. Hyperexponential case

To support the conjectured insensitivity of the bounds, we also computed the blocking probabilities in the case of hyperexponential service distributions; that is, for service requirements of two successive exponential phases both with parameter  $\gamma_1$  at node 1 and similarly with parameter  $\gamma_2$  at node 2. Hence, the mean service requirements are  $\tau_i = 2\gamma_i^{-1}$ ,  $i = 1, 2$ .

Since the insensitivity of the product forms is only guaranteed for service disciplines satisfying a notion of JLB, we chose as natural candidate a computer sharing discipline with unit rate. Hence, if only one job is present at a node, it is served at rate 1, whereas if 2 jobs are present they are both served simultaneously each at a rate 0.5. Further, since the state description has to be extended, such as by listing the number of residual phases for each job, we restricted ourselves to the buffer sizes  $N_1 = N_2 = 2$ . (The number of states in this case is 36).

Similarly to the exponential case, as described in Section 6.2, the blocking probabilities were calculated using the power method. Essentially, only the transition matrix of the embedded Markov chain became more complex, particularly because of the required blocking protocol. Although the theory of JLB suggested insensitivity results only for a protocol which prescribes a repetition of a complete service at node 1 when

$\tau_1$	$\tau_2$	$B_L$	$B_W$	$B_R$	$B_E$	$B_U$
1	1	0.33333	0.38000	0.39857	0.42222	0.50000
2	2	0.62016	0.64220	0.65539	0.67580	0.72727
3	3	0.74040	0.75419	0.76477	0.77986	0.81818
4	4	0.80352	0.81376	0.82228	0.83404	0.86487
5	5	0.84209	0.85032	0.85737	0.86697	0.89286
2	5	0.80484	0.80957	0.81503	0.82102	0.84615
4	5	0.82684	0.83540	0.84322	0.85283	0.88048
2	10	0.90022	0.90111	0.90223	0.90325	0.91247
2	0.5	0.52381	0.57244	0.57345	0.57759	0.60000

Hyperexponential case

Numerical values with  $\lambda = 1$  and  $N_1 = N_2 = 2$

**Table 6.5.3. Comparison with hyperexponential service times**

blocking at node 2 occurs [see Hordijk and Van Dijk (1983a)], we also investigated the protocol where only the last exponential phase of the service needs to be repeated after a blocking.

Note that this latter protocol can also be interpreted as interruption of only the last phase. Hence, with a fixed mean service time  $\tau_1$  at node 1 and a service distribution of  $n$  successive phases, the expected time of the last phase is  $(n\tau_1)^{-1}$ , which tends to zero if  $n$  gets larger. Thus, this latter protocol approximately performs as a "waiting protocol" in which a job only has to wait at the preceding node exactly until the next node gets a vacancy. We will therefore refer to our original and this new blocking protocol as "returning", respectively "waiting", and use the notation:

$B_E$ : for the blocking probability in the exponential case (as by interruption);

$B_R$ : for the blocking probability under the returning blocking protocol;

$B_W$ : for the blocking probability under the waiting blocking protocol.

Even when the bounds are very close (within 1%) we observed (see Table 6.5.3) that

$$B_L < B_W < B_R < B_E < B_U. \quad (6.5.2)$$

This observation suggests the following conclusions (all w.r.t. computer sharing).

1. The bounds hold for phase type distributions under the returning protocol; that is, they are insensitive bounds, as was expected based upon JLB.
2. The bounds are also insensitive under the waiting protocol; this could have been expected for the upper bound as from (6.5.3) below. For the lower bound, we could give an intuitive argument similar to the exponential case.
3. The blocking protocols are related by

$$B_W < B_R < B_E. \quad (6.5.3)$$

This can be argued as follows:

$B_R < B_E$ : in the exponential case the returning protocol can be seen as an interruption of the total exponential service. Hence, a blocked job always still has to complete a full exponential service with mean  $\tau_1$ . For phase type distributions it may occur that during the first phase of a service the second node becomes saturated and then free again, without interrupting the service at all. The expected service time still needed by a job at node 1 after a saturation at node 2 will therefore be less than  $\tau_1$ . Clearly, smaller sojourn times at node 1 imply a less frequent congestion and therefore blocking at node 1.

$B_W < B_R$ : a repetition of only the last phase requires less time than the repetition of all phases. Hence, as above, we conclude less blocking.

## Conclusion

This research was intended to produce the following contributions to the fields of queueing and dynamic programming.

First, the matrix decomposition approach suggests a new way to look at the evaluation of the stationary characteristics of the models. It is hoped that this approach, which is mathematically simple and direct, will contribute to make the models easier to use and understand, and stimulate their development and application.

Second, the formulation of probabilistic models in the language of matrix algebra puts in evidence certain computational aspects of the solution. It is hoped that this work has contributed to point out some similarities between the computational problems of applied probability and the standard solution techniques of numerical algebra.

Finally, each one of the individual results is, of course, a contribution in its own right.

## Bibliography

- [1] Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1974). *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass.
- [2] Anstreicher, K.M. and Rothblum, U.G. (1985). Computing the index and Drazin inverse using the shuffle algorithm. Working Paper AR5384, School of Organization and Management, Yale University, New Haven, CT 06520.
- [3] Barbour, A.D. (1976). Networks of queues and the method of stages. *Adv. Appl. Prob.* **8**, pp. 584–591.
- [4] Baskett, F., Chandy, M., Muntz, R. and Palacios, J. (1975). Open, closed and mixed networks of queues with different classes of customers. *J.A.C.M.* **22**, pp. 248–260.
- [5] Berman, A. and Plemmons, R.J. (1979). *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York.
- [6] Blackwell, D. (1962). Discrete dynamic programming. *Ann. Math. Statist.* **33**, pp. 719–726.
- [7] Brumelle, S.L. (1971). On the relation between customer and time averages in queues. *J. Appl. Prob.* **8**, pp. 508–520.
- [8] Campbell, S.L. and Meyer, C.D. Jr. (1979). *Generalized Inverses of Linear Transformations*. Pitman, London.
- [9] Chandy, K.M., Howard, J.H. and Townsley, D.F. (1977). Product form and local balance in queueing networks. *J.A.C.M.* **24**, pp. 250–263.
- [10] Çinlar, E. (1975). *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, N.J.
- [11] Denardo, E.V. (1970). Computing a bias-optimal policy in a discrete-time Markov decision problem. *Operations Research* **18**, pp. 279–289.
- [12] Denardo, E.V. (1971). Markov renewal programs with small interest rates. *Ann. Math. Statist.* **42**, pp. 477–496.
- [13] van Dijk, N. and Lamond, B.F. (1985). Bounds for the call congestion in tandem queues. Working Paper #1078, The University of British Columbia, Faculty of Commerce and Business Administration, 2053 Main Mall, Vancouver, B.C., Canada V6T 1Y8.

- [14] Disney, R.L. and Chandramohan, J. (1980). A correction note on "Two finite M/M/1 queues in tandem: a matrix solution for the steady-state". *Opsearch* **17**, pp. 181–183.
- [15] Disney, R.L. and König, D. (1985). Queueing networks: a survey of their random processes. *SIAM Review* **27**, pp. 335–403.
- [16] Drazin, M.P. (1958). Pseudo-inverses in associative rings and semigroups. *Amer. Math. Monthly* **65**, pp. 506–514.
- [17] Faddeeva, V.N. (1959). *Computational Methods of Linear Algebra*. Dover, New York.
- [18] Federgruen, A. and Spreen, D. (1980). A new specification of the multichain policy iteration algorithm in undiscounted Markov renewal programs. *Management Science* **26**, pp. 1211–1217.
- [19] Funderlic, R.E. and Plemmons, R.J. (1981). LU decomposition of M-matrices by elimination without pivoting. *Lin. Alg. Appl.* **41**, pp. 99–110.
- [20] Gantmacher, F.R. (1959). *The Theory of Matrices 1* (English translation). Chelsea, New York.
- [21] Golub, G.H. and Van Loan, C.F. (1983). *Matrix Computations*. The Johns Hopkins University Press, Baltimore.
- [22] Golub, G.H. and Wilkinson, J.H. (1976). Ill-conditioned eigensystems and the computation of the Jordan canonical form. *SIAM Review* **18**, pp. 578–619.
- [23] Greville, T.N.E. (1973). The Souriau-Frame algorithm and the Drazin pseudoinverse. *Lin. Alg. Appl.* **6**, pp. 205–208.
- [24] Harrod, W.J. and Plemmons, R.J. (1984). Comparison of some direct methods for computing stationary distributions of Markov chains. *SIAM J. Stat. Comput.* **5**, pp. 453–469.
- [25] Hartwig, R.E. (1976). More on the Souriau-Frame algorithm and the Drazin inverse. *SIAM J. Appl. Math.* **31**, pp. 42–46.
- [26] Heyman, D.P. and Sobel, M.J. (1982). *Stochastic Models in Operations Research, Volume I*. McGraw-Hill, New York.
- [27] Heyman, D.P. and Sobel, M.J. (1984). *Stochastic Models in Operations Research, Volume II*. McGraw-Hill, New York.
- [28] Hordijk, A. and Van Dijk, N. (1981). Networks of queues with blocking. *Performance* **81** (ed. K.J. Kylstra). North-Holland, pp. 51–65.
- [29] Hordijk, A. and Van Dijk, N. (1982). Stationary probabilities for networks of queues. In *Applied Probability—Computer Science: The Interface* (eds. L. Disney and T.J. Ott), Birkhäuser, Vol. II, pp. 423–451.

- [30] Hordijk, A. and Van Dijk, N. (1983a). Networks of queues, part I: Job-local-balance and the adjoint process, part II: General routing and service characteristics. *Proceedings International Seminar on Modelling and Performance Evaluation Methodology*, INRIA, Vol. I, part I, pp. 79–104, part II, pp. 105–135. To appear in *Lecture Notes in Control and Information Sciences*, Springer Verlag, Berlin.
- [31] Hordijk, A. and Van Dijk, N. (1983b). Adjoint processes, job-local-balance and insensitivity for stochastic networks. To appear in *Proceedings 44th Session JSI*, Madrid, Spain.
- [32] Howard, R.A. (1960). *Dynamic Programming and Markov Processes*. Wiley, New York.
- [33] Jackson, J.R. (1957). Networks of waiting lines. *Operations Research* **5**, pp. 518–521.
- [34] Jackson, J.R. (1963). Jobshop-like queueing systems. *Management Science* **10**, pp. 131–142.
- [35] Johnston, R.L. (1982). *Numerical Methods: a Software Approach*. Wiley, New York.
- [36] Karlin, Samuel. (1969). *A First Course in Stochastic Processes*. Academic Press, New York.
- [37] Kaufman, L. (1983). Matrix methods for queueing problems. *SIAM J. Sci. Stat. Comput.* **4**, pp. 525–552.
- [38] Kelly, F.P. (1979). *Reversibility and Stochastic Networks*. Wiley.
- [39] Kemeny, J.G. and Snell, J.L. (1960). *Finite Markov Chains*. Van Nostrand, New York.
- [40] Kingman, J.F.C. (1969). Markov population processes. *J. Appl. Prob.* **6**, pp. 1–18.
- [41] Lamond, B.F. and Puterman, M.L. (1985). Generalized inverses in discrete time Markov decision processes. Working paper #1069, The University of British Columbia, Faculty of Commerce and Business Administration, 2053 Main Mall, Vancouver, B.C., Canada V6T 1Y8. Submitted, *Math. O.R.*
- [42] Meyer, C.D. Jr. (1980). The condition number of a finite Markov chain and perturbation bounds for the limiting probabilities. *SIAM J. Alg. Disc. Meth.* **1**, pp. 273–283.
- [43] Miller, B.L. and Veinott, A.F. Jr. (1969). Discrete dynamic programming with a small interest rate. *Ann. Math. Statist.* **40**, pp. 366–370.
- [44] Puterman, M.L. and Brumelle, S.L. (1979). On the convergence of policy iteration in stationary dynamic programming. *Math. O.R.* **4**, pp. 60–69.
- [45] Ross, S.M. (1970). *Applied Probability Models with Optimization Applications*. Holden-Day, San Francisco.
- [46] Rothblum, U.G. (1981). Resolvent expansions of matrices and applications. *Lin. Alg. Appl.* **38**, pp. 33–49.

- [47] Schassberger, R. (1978). The insensitivity of stationary probabilities in networks of queues. *Adv. Appl. Prob.* **10**, pp. 906–912.
- [48] Schweitzer, P.J. (1971). Iterative solution of the functional equations of undiscounted Markov renewal programming. *J. Math. Anal. Appl.* **34**, pp. 495–501.
- [49] Serfozo, R.F. (1979). An equivalence between continuous and discrete time Markov decision processes. *Operations Research* **27**, pp. 616–620.
- [50] Stewart, G.W. (1984). Rank degeneracy. *SIAM J. Sci. Stat. Comput.* **5**, pp. 403–413.
- [51] Stewart, W.J. (1978). A comparison of numerical techniques in Markov modelling. *Communications of the A.C.M.* **21**, pp. 144–152.
- [52] Varga, R.S. (1962). *Matrix Iterative Analysis*. Prentice-Hall, Englewood-Cliffs, N.J.
- [53] Veinott, A.F. Jr. (1966). On finding optimal policies in discrete dynamic programming with no discounting. *Ann. Math. Statist.* **37**, pp. 1284–1294.
- [54] Veinott, A.F. Jr. (1969). Discrete dynamic programming with sensitive discount optimality criteria. *Ann. Math. Statist.* **40**, pp. 1635–1660.
- [55] Veinott, A.F. Jr. (1974). Markov decision chains. In *Studies in Optimization* **10**, MAA Studies in Mathematics, edited by G.B. Dantzig and B.C. Eaves.
- [56] Whittle, P. (1968). Equilibrium distributions for an open migration model. *J. Appl. Prob.* **5**, pp. 567–571.
- [57] Whittle, P. (1969). Nonlinear migration processes. *Internal. Statist. Inst. Bull.* **42**, Bk. 1, pp. 642–647.
- [58] Wilkinson, J.H. (1965). *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford.
- [59] Wilkinson, J.H. (1982). Note on the practical significance of the Drazin inverse. In Campbell, S.L. (ed.), *Recent Applications of Generalized Inverses*. Pitman, Boston, pp. 82–99.
- [60] Wolff, R.W. (1982). Poisson arrivals see time averages. *Operations Research* **30**, pp. 223–231.
- [61] Wong, B., Giffin, W. and Disney, R.L. (1977). Two finite M/M/1 queues in tandem: a matrix solution for the steady-state. *Opsearch* **14**, pp. 1–8.
- [62] Yamazaki, G., Kawashima, T. and Sakasegawa, H. (1985). Reversibility of tandem blocking queueing systems. *Management Science* **31**, pp. 78–83.

BERNARD F. LAMOND

Publications:

- B.F. Lamond (1985). Matrix Methods in Queueing and Dynamic Programming. Ph.D. dissertation, The University of British Columbia.
- N. van Dijk and B.F. Lamond (1985). Bounds for the call congestion in tandem queues. Submitted, Operations Research.
- B.F. Lamond and M.L. Puterman (1985). Generalized inverses in discrete time Markov decision processes. Submitted, Mathematics of O.R.
- B. Lamond and N.F. Stewart (1981). Bregman's balancing method. Transportation Research, Vol. 15B, pp. 239-248.
- B. Lamond (1980). Balancement de matrices et optimisation d'entropie. Master's thesis, Université de Montréal.