Algorithms for Return Probabilities for Stochastic Fluid Flows

Nigel G. Bean, Małgorzata M. O'Reilly*and P. G. Taylor†

September 22, 2004

Abstract

We consider several known algorithms and introduce some new algorithms which can be used to calculate the probability of return to the initial level in the Markov stochastic fluid flow model. We give the physical interpretations of these algorithms within the fluid flow environment. The rates of convergence are explained in terms of the physical properties of the fluid flow processes. We compare these algorithms with respect to the number of iterations required and their complexity. The performance of the algorithms depends on the nature of the process considered in the analysis. We illustrate this with examples and give appropriate recommendations.

Keywords Markovian fluid model, return probabilities, fixed-point iterations, Newton's method, Asmussen's iteration, Latouche-Ramaswami method.

1 Introduction

Fluid flow models of the class considered in this paper were studied by, for example, Anick, Mitra and Sondhi [2], Asmussen [3], Ramaswami [21], Rogers [22], da Silva Soares and Latouche [9], and Ahn and Ramaswami [1]. In these models, the rate c_i at which the level of a fluid increases, or decreases, is governed by the state i of the underlying continuous-time Markov

*Applied Mathematics, University of Adelaide, SA 5005, Australia

†Department of Mathematics and Statistics, University of Melbourne, Vic 3010, Australia.

chain. The parameters c_i can be positive, negative or zero. Recently [6], we established expressions for several performance measures for this class of models. In this paper, we shall describe and analyze algorithms that can be used to calculate return probabilities to the initial level, from which all performance measures defined in [6] can be derived. We shall simplify the general fluid flow model by assuming c_i to be 1 or -1, because as shown in [3, 6, 22], general models can be transformed into such a model while preserving return probabilities. Such a simplified fluid flow model is a two-dimensional continuous-time uncountable-level finite-phase Markov process, denoted by $\{(X(t), \varphi(t)) : t \in \mathcal{R}^+\}$, where

- the level is denoted by $X(t) \in \mathcal{R}^+$,
- the phase is denoted by $\varphi(t) \in \mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$,
- the phase process $\{\varphi(t) : t \in \mathcal{R}^+\}$ is an irreducible Markov chain with infinitesimal generator $\mathcal{T}$, and
- the net rate of input c_i to the infinite fluid buffer, when $\varphi(t)$ is in state i, is equal to 1 for $i \in \mathcal{S}_1$ and -1 for $i \in \mathcal{S}_2$.

We partition the infinitesimal generator according to $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ so that

$$\mathcal{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$$

and let $s_1 = |\mathcal{S}_1|$, $s_2 = |\mathcal{S}_2|$.

Assume that the process $(X(t), \varphi(t))$ starts from level x in phase $i \in \mathcal{S}_1$ at time 0. Let $\theta(x) = \inf\{t > 0 : X(t) = x\}$ be the first passage time to level x. For $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, let

$$\Psi_{ij} = P[\theta(x) < \infty \ , \ \varphi(\theta(x)) = j \mid X(0) = x, \varphi(0) = i], \tag{1}$$

and $\Psi = [\Psi_{ij}]$. In physical terms, Ψ_{ij} is the probability that, starting from level x in phase $i \in \mathcal{S}_1$, the process $(X(t), \varphi(t))$ first returns to level x in finite time and does so in phase $j \in \mathcal{S}_2$, while avoiding levels below x. The matrix Ψ records return probabilities to the initial level for all such i, j. We have dropped the subscript x from the notation, since these probabilities do not depend on the starting level in this upwardly-homogenous process. Without loss of generality, we shall often assume that the process starts from level zero at time 0.

By [6], the matrix analogous to Ψ, defined for the general model in which rates c_i can take any value in $\mathcal{R}$, can be calculated from the corresponding matrix in this simplified model. The matrix Ψ appears in expressions for other measures of the process, see for example [6, 21, 9], and consequently, calculation of Ψ is an obvious first step which has become the focus of research. It has been determined [6, 22, 25] that the matrix Ψ is the minimal nonnegative solution of the nonsymmetric algebraic Riccati equation,

$$T_{11}\Psi + \Psi T_{22} + \Psi T_{21}\Psi + T_{12} = \mathbf{0}. \tag{2}$$

A number of methods that can be used to solve this equation have been reported by Guo [13]. Asmussen [3] established algorithms for Ψ by using an alternative approach. A further method was proposed by Ramaswami [21], who related the fluid process to a discrete-level quasi-birth-and-death process (QBD). This allows one to calculate Ψ using any algorithms available for QBDs, such as the Logarithmic Reduction algorithm of Latouche and Ramaswami [19] or the Cyclic Reduction algorithm of Bini and Meini [8]. In this paper we introduce two new algorithms, and analyze the above-mentioned algorithms with respect to their physical interpretation, complexity and convergence properties. The performance of the algorithms, which depends on the physical properties of the process, is illustrated with examples. We make recommendations in relation to which method is best in which circumstances.

This paper is organized as follows. Section 2 contains a brief description of the methods, which are then compared in Section 3 via physical interpretations. In Section 4 we show that the algorithms converge to Ψ and compare them with respect to the number of iterations required. Results on the rates of convergence are given in Section 5. In Section 6 we compare the numerical complexity of the algorithms. The examples in Section 7 focus on the overall performance of these algorithms. This is followed by concluding remarks in Section 8.

For $p \times q$ matrices A and B, we shall write $A < B$ if $A_{ij} < B_{ij}$ for all $i \in \{1, \ldots, p\}$, $j \in \{1, \ldots, q\}$, with similar definitions for $>$, $\leq$, $\geq$, and $=$. We shall write $A \neq B$ if $A = B$ is not satisfied. We shall use the notation A^t for the transpose of the matrix A and $\mathbf{0}$ for the zero matrix of an appropriate size. When necessary, we shall indicate that $\mathbf{0}$ is a square matrix of order s_1 or s_2 with the notation $\mathbf{0}_{11}$ or $\mathbf{0}_{22}$, respectively. Similarly, I_{11} and I_{22} shall be the identity matrices of order s_1 and s_2, respectively, and $\mathbf{e}$ a vector of ones of appropriate size.

2 Algorithms

Before we refer to known methods, we note that by the simple action of writing equation (2) in a different form and then introducing an iteration, several algorithms for Ψ can be obtained. For example, we can write (2) as

$$T_{11}\Psi + \Psi T_{22} = -T_{12} - \Psi T_{21}\Psi.$$

A corresponding iteration can be obtained by setting $\hat{\Psi}_0 = \mathbf{0}$, and then using

$$T_{11}\hat{\Psi}_{n+1} + \hat{\Psi}_{n+1}T_{22} = -T_{12} - \hat{\Psi}_n T_{21}\hat{\Psi}_n, \tag{3}$$

to calculate subsequent values of $\hat{\Psi}_n$. In a similar manner we can obtain three more iterations. In each, we start the iteration with the matrix $\mathbf{0}$. The $(n+1)$-st iteration is given, respectively, by the equations

$$T_{11}\acute{\Psi}_{n+1} + \acute{\Psi}_{n+1}(T_{22} + T_{21}\acute{\Psi}_n) = -T_{12}, \tag{4}$$

$$(T_{11} + \grave{\Psi}_n T_{21})\grave{\Psi}_{n+1} + \grave{\Psi}_{n+1}T_{22} = -T_{12}, \tag{5}$$

$$(T_{11} + \Psi_n T_{21})\Psi_{n+1} + \Psi_{n+1}(T_{22} + T_{21}\Psi_n) = -T_{12} + \Psi_n T_{21}\Psi_n. \tag{6}$$

The notation $\hat{\Psi}_{n+1}$, $\acute{\Psi}_{n+1}$, $\grave{\Psi}_{n+1}$ and Ψ_{n+1} reflects the fact that the equations (3)-(6) define different iterations. Alternative versions of the iterations (3)-(6), with a nonzero matrix in the 0-th iteration, could also be considered, but this is beyond the scope of this paper. All of (3)–(6) are equations of type $AX + XB = C$. These can be solved using the Bartels-Stewart algorithm [4] or the Hessenberg-Schur method [11].

Guo [13] analyzed several methods which can be used to find the minimal nonnegative solution of equation (2): fixed-point iterations, Newton's method and the Schur method. The Schur method, which is not an iterative method, but an approximation, is not considered here. We note however, that this approximation can be used to replace the 0-th iteration of the iterations (3)-(6) and can considerably reduce the computational cost, as pointed out by Guo [13] in relation to Newton's method. Guo's fixed-point iteration FP3 is the only fixed-point iteration worth considering, as it is the fastest in terms of the number of iterations [13]. Fixed-point iteration FP3 and Newton's method [13] are equivalent in our model to the iterations (3) and (6) respectively. We shall refer to the iterations (4) and (5) as the *First-Exit algorithm* and the *Last-Entrance algorithm* respectively. The reason for this choice of names will become apparent in Section 3, when we present their physical interpretations.

In [21], Ramaswami related the fluid process to a discrete time QBD, which allows for efficient calculation of Ψ by existing algorithms [8, 19]. Da Silva Soares and Latouche [9, Section 4] gave a probabilistic interpretation of this construction. We give a brief summary of this interpretation in Section 3. We shall refer to the method which consists of the QBD construction [21] and the Logarithmic Reduction algorithm [19] as the *Latouche-Ramaswami method*. The method which consists of the QBD construction [21] and the Cyclic Reduction algorithm [8] shall be referred to as the *Bini-Meini-Ramaswami method*. Since these two methods are similar, we shall discuss the physical interpretation of the Latouche-Ramaswami method only. However, numerical implementations of both the Latouche-Ramaswami and the Bini-Meini-Ramaswami method will be compared with the above mentioned algorithms in Section 7.

Asmussen [3, Section 3] proposed three iteration schemes which can be used to calculate Ψ and which, unlike Ramaswami's construction, operate within the stochastic fluid environment. However, he gave no numerical analysis of these schemes. We shall refer to the first scheme [3, Theorem 3.1] as *Asmussen's iteration*. In Section 3 we give the physical interpretation of this iteration. The second scheme [3, Theorem 3.2] can require many more iterations than the first, as pointed out by Asmussen and confirmed by our numerical experience. For this reason, we shall not discuss it here.

The third scheme [3, Theorem 3.3] requires calculation of the inverse of a Kronecker sum [12] at each step of the scheme, which is inefficient numerically. However, the equation in [3, Theorem 3.3], underlying this iteration scheme, can be shown to be equivalent to (4). This can be done in two steps. First, convert (4) into the equivalent form

$$vec\acute{\Psi}_{n+1} = -\Big[(T_{22} + T_{21}\acute{\Psi}_n)^t \oplus T_{11}\Big]^{-1} vecT_{12}, \tag{7}$$

where $A \oplus B$ is a Kronecker sum [12] of matrices A and B, and $vecA$ [12] is an ordered stack of the columns of A. Second, observe that the equation in [3, Theorem 3.3], (after correcting for the missing minus sign there) is equivalent to (7). In the First-Exit algorithm, the $(n+1)$-st iteration is obtained by solving equation (4), using the Bartels-Stewart algorithm [4]. We shall analyze this improved form here.

3 Physical Interpretations

We start by rewriting equations (3)-(6) in a form, which is more convenient for physical interpretation. This is given in the following lemma.

Lemma 1 *Equations (3) to (6) are equivalent to*

$$\hat{\Psi}_{n+1} = \int_0^\infty e^{T_{11}y}(T_{12} + \hat{\Psi}_n T_{21}\hat{\Psi}_n)e^{T_{22}y}dy, \tag{8}$$

$$\acute{\Psi}_{n+1} = \int_0^\infty e^{T_{11}y}T_{12}e^{(T_{22}+T_{21}\acute{\Psi}_n)y}dy, \tag{9}$$

$$\grave{\Psi}_{n+1} = \int_0^\infty e^{(T_{11}+\grave{\Psi}_n T_{21})y}T_{12}e^{T_{22}y}dy, \tag{10}$$

$$\Psi_{n+1} = \int_0^\infty e^{(T_{11}+\Psi_n T_{21})y}(T_{12} - \Psi_n T_{21}\Psi_n)e^{(T_{22}+T_{21}\Psi_n)y}dy. \tag{11}$$

Furthermore, for fixed $\hat{\Psi}_n$, $\acute{\Psi}_n$, $\grave{\Psi}_n$ and Ψ_n, each of the equations (3)-(6) has a unique solution.

Proof
In this proof we shall use the notation $\hat{\Psi}_{n+1}$, $\acute{\Psi}_{n+1}$, $\grave{\Psi}_{n+1}$ and Ψ_{n+1} to denote only the matrices defined in (8)-(11). The proof of the equivalence of (8)-(11) to (3)-(6) respectively follows by Theorem 9.2 in [7] and the fact that eigenvalues with the maximum real part of each of the matrices T_{11}, T_{22}, $(T_{22} + T_{21}\acute{\Psi}_n)$, $(T_{11} + \grave{\Psi}_n T_{21})$, $(T_{11} + \Psi_n T_{21})$ and $(T_{22} + T_{21}\Psi_n)$ are negative. This was established in [6, Lemma 2] for the first two matrices. The proof for the remaining matrices is analogous.

For example, by Theorems 1 and 2 below, we have $\acute{\Psi}_n < \Psi$ and so $(T_{22} + T_{21}\acute{\Psi}_n)\mathbf{e} \leq (T_{22} + T_{21}\Psi)\mathbf{e}$ with a strict inequality in at least one place. Hence, since $(T_{22} + T_{21}\Psi)\mathbf{e} \leq (T_{22} + T_{21})\mathbf{e} = \mathbf{0}$, we have $(T_{22} + T_{21}\acute{\Psi}_n)\mathbf{e} \leq \mathbf{0}$ with a strict inequality in at least one place. Next, repeat the argument in [6] to show that the eigenvalue of $(T_{22} + T_{21}\acute{\Psi}_n)$ with the maximum real part is negative. Hence, by Theorem 9.2 in [7], (9) is equivalent to (4).

Finally, by [12], each of the equations (3)-(6) has a unique solution. ■

In Theorem 2 below, we show that the matrices $\hat{\Psi}_n$, $\acute{\Psi}_n$, $\grave{\Psi}_n$ and Ψ_n converge to Ψ as $n \to \infty$. By taking limits as $n \to \infty$ in equations (8)-(11) we obtain three known expressions for Ψ [6, 9, 21] and the new expression

$$\Psi = \int_0^\infty e^{(T_{11}+\Psi T_{21})y}(T_{12} - \Psi T_{21}\Psi)e^{(T_{22}+T_{21}\Psi)y}dy.$$

We shall see that, at each iteration of the algorithms (8)-(11) (with a zero matrix assumed at the 0-th step) the matrix records the total probability mass of certain sample paths, but not others. We shall refer to the sets of sample paths that contribute to $\hat{\Psi}_n$, $\acute{\Psi}_n$, $\grave{\Psi}_n$ and Ψ_n, respectively as $\hat{\Omega}_n$, $\acute{\Omega}_n$, $\grave{\Omega}_n$ and Ω_n. When a sample path starts and finishes at some level w different to level zero, we shall call it a sample path in (say) $\hat{\Omega}_n$ *shifted to level* w. Our analysis shall lead to the determination of exactly which sample paths lie in each of $\hat{\Omega}_n$, $\acute{\Omega}_n$, $\grave{\Omega}_n$ and Ω_n via useful physical interpretations. Later, in Section 4, we shall use these interpretations to compare the algorithms with respect to the number of iterations required.

The physical interpretation of Newton's method [13] was previously unknown. Another novel aspect is our physical interpretation of the Latouche-Ramaswami method. Da Silva Soares and Latouche [9] gave a physical interpretation of the QBD constructed in [21]. Our contribution is the physical interpretation of the n-th iteration of the Logarithmic Reduction algorithm [19] applied to such a construction, in terms of a transformed fluid model.

The Cyclic Reduction algorithm [8] has a lower computational cost per iteration than the Logarithmic Reduction algorithm [19], as the former requires six matrix multiplications and one matrix inversion while the latter requires eight matrix multiplications and one matrix inversion per step [8]. The physical interpretations of the two algorithms, in the QBD environment, are similar. Both the Cyclic Reduction and the Logarithmic Reduction algorithm calculate the matrix G, from which Ψ can be determined.

With a "fair" counting of the number of iterations (the first iteration to include those sample paths that start at level 1 and hit level zero without reaching level 2), the k-th iteration in the Logarithmic Reduction algorithm is the approximation of the matrix G, incorporating all paths starting from level n that do not visit level $n + 2^k - 1$ or above, while the k-th iteration of the Cyclic Reduction algorithm is the approximation of the matrix G, incorporating all paths starting from level n that do not visit level $n + 2^{k-1}$ or above. Thus it is likely that the Cyclic Reduction algorithm will take one more iteration than the Logarithmic Reduction algorithm to reach a desired degree of accuracy. However, this extra iteration is likely to be compensated for in terms of the per-iteration efficiency of the Cyclic Reduction algorithm and we can conclude that this algorithm will use less CPU time in most cases. We shall expand on this in Section 6.

3.1 Fixed-Point Iteration FP3

Let the matrix $P(k)$ be such that $[P(k)]_{ij}$ is the probability that, starting from phase i, the process first returns to the initial level in finite time, does so in phase j, and there are exactly k transitions from $\mathcal{S}_1$ to $\mathcal{S}_2$ (or *peaks*) during this journey. We have

$$\Psi_{ij} = \sum_{k=1}^{\infty} [P(k)]_{ij}.$$

Conditioning analogous to the one outlined below, was introduced in [6] to derive the Laplace-Stieltjes transform of the time taken to return to the initial level, for a general model. By conditioning on the maximum fluid level reached and by (8), we have

$$P(1) = \hat{\Psi}_1 = \int_0^{\infty} e^{T_{11}y} T_{12} e^{T_{22}y} dy. \tag{12}$$

For $n \geq 1$, consider the paths in the set $\hat{\Omega}_{n+1}$. By (8),

$$\hat{\Psi}_{n+1} = \hat{\Psi}_1 + \int_0^{\infty} e^{T_{11}w} \hat{\Psi}_n T_{21} \hat{\Psi}_n e^{T_{22}w} dw, \tag{13}$$

and so $\hat{\Omega}_1 \subset \hat{\Omega}_{n+1}$. Moreover, with w defined as

$$w = \inf\{x : \text{ transition } \mathcal{S}_2 \to \mathcal{S}_1 \text{ occurs at level } x\}, \tag{14}$$

$\hat{\Omega}_{n+1}$ includes every path which has the following physical interpretation.

- Starting from level zero in phase i at time 0, the phase process remains in the set $\mathcal{S}_1$, until the fluid level reaches w in some phase $k \in \mathcal{S}_1$. Since the net input rates in $\mathcal{S}_1$ are 1, this happens at time w. The probability of this is $[e^{T_{11}w}]_{ik}$.
- Then, a sample path in $\hat{\Omega}_n$ shifted to level w, ending in some $k' \in \mathcal{S}_2$, occurs.
- Next, a transition from $k' \in \mathcal{S}_2$ to some $l' \in \mathcal{S}_1$ occurs at level w. This happens at a rate $[T_{21}]_{k'l'}$.
- Next, a sample path in $\hat{\Omega}_n$ shifted to level w, starting in l' and ending in l, occurs.

- Finally, the phase process remains in the set $\mathcal{S}_2$, until the fluid level first returns to level 0 and does so in phase $j \in \mathcal{S}_2$. Since the net input rates in $\mathcal{S}_2$ are -1, the duration of time spent in $\mathcal{S}_2$ during this stage of the process is w. The probability of this is $[e^{T_{22}w}]_{lj}$.

From this physical interpretation, we are able to deduce some useful information about the number of peaks in a sample path contributing to the n-th iteration of FP3. This is established in the form of the inequality below.

Lemma 2 *For $n \geq 1$, the matrix $\hat{\Psi}_n$ satisfies*

$$\sum_{k=1}^{n} P(k) \leq \hat{\Psi}_n \leq \sum_{k=1}^{2^{n-1}} P(k). \tag{15}$$

Proof
Let σ_n be the maximum number of peaks of a sample path in $\hat{\Omega}_n$, starting in some $i \in \mathcal{S}_1$ and finishing in some $j \in \mathcal{S}_2$. Clearly, $\sigma_{n+1} = 2\sigma_n$ for $n \geq 1$, and so by mathematical induction $\sigma_n = 2^{n-1}$ for $n \geq 1$. Hence, the right hand inequality of (15) holds.

We shall now show that, for all $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, any sample path contributing to $\sum_{k=1}^{n}[P(k)]_{ij}$ lies in $\hat{\Omega}_n$. By (12), this is true for $n = 1$. Assume that this is true for fixed $n \geq 1$. Consider a sample path contributing to $\sum_{k=1}^{n+1}[P(k)]_{ij}$. In this path, any return path to level w, defined in (14), must have no more than n peaks, and hence, by the inductive assumption, must lie in $\hat{\Omega}_n$. Consequently, from the physical interpretation of fixed-point iteration FP3 given above, a sample path contributing to $\sum_{k=1}^{n+1}[P(k)]_{ij}$ lies in $\hat{\Omega}_{n+1}$ and so $\sum_{k=1}^{n+1} P(k) \leq \hat{\Psi}_{n+1}$. Hence, the result follows by mathematical induction. ■

Remark 1 The inequalities in (15) cannot be replaced with strict ones. For example, consider a process with $\mathcal{S}_1 = \{1, 2\}$, $\mathcal{S}_2 = \{3, 4\}$, and generator

$$\mathcal{T} = \left[\begin{array}{cc|cc} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ \hline 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{array}\right].$$

In this process, the number of transitions $\mathcal{S}_1 \to \mathcal{S}_2$ required for the process to reach phase 3, starting from phase 1, must be odd. Thus, we have $[P(4)]_{13} = 0$, and so by (15), for $n = 3$ we have equality in the $(1, 3)$-th place on both sides of (15). ■

3.2 The First-Exit Algorithm

The physical interpretations of (9)-(11) require the definitions of "upward" and "downward" processes. In the First-Exit algorithm, the upward process has generator T_{11} and the downward process has generator $T_{22} + T_{21}\acute{\Psi}_n$.

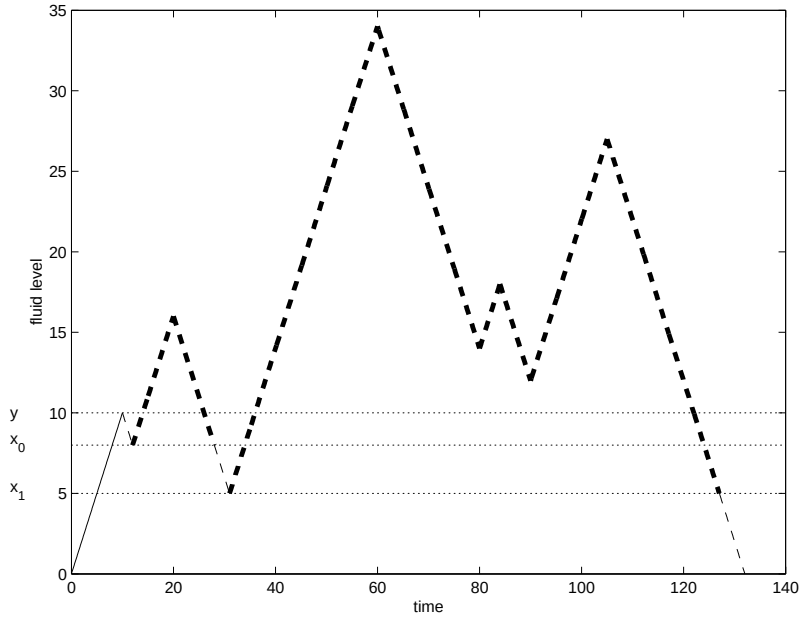

Figure 1: A path in $\acute{\Omega}_3$.

By (9) and (12) we have $\acute{\Psi}_1 = P(1)$. The physical interpretation of the entry $[\acute{\Psi}_{n+1}]_{ij}$ is that it is the probability that, starting from level zero in phase i at time 0,

- the upward process takes place until the fluid level first moves to some level y, and does so in some phase $k \in \mathcal{S}_1$. The probability of this is $[e^{T_{11}y}]_{ik}$.
- Next, a transition from $k \in \mathcal{S}_1$ to some phase $l \in \mathcal{S}_2$ occurs, at a rate $[T_{12}]_{kl}$.
- Finally, the downward process with generator $T_{22}+T_{21}\acute{\Psi}_n$ begins, which takes place until the fluid level first returns to level zero, and does so in phase j. The probability of this is $[e^{(T_{22}+T_{21}\acute{\Psi}_n)y}]_{lj}$. This process can

include a transition $\mathcal{S}_2 \to \mathcal{S}_1$, say at some level x, but the sample path between this point and the subsequent return to level x must be in $\acute{\Omega}_n$.

We illustrate this in Figure 1. Upward and downward processes are represented by solid and dashed lines, respectively. Level y is the level at which the phase process first leaves the set $\mathcal{S}_1$. Since the net input rates in $\mathcal{S}_1$ are 1, y is also a time at which this happens. In the downward process, transitions $\mathcal{S}_2 \to \mathcal{S}_1$ occur at x_0 and x_1, but the subsequent return paths to levels x_0 and x_1 are in $\acute{\Omega}_1$ shifted to level x_0, and $\acute{\Omega}_2$ shifted to level x_1, respectively. Both are indicated by thick dashed lines. The formula (9) is obtained by conditioning on y, the first time the process exits $\mathcal{S}_1$. This motivates the name of the algorithm. There is no restriction on the maximum number of peaks in a sample path contributing to $\acute{\Psi}_n$ for $n > 1$.

Note that conditioning analogous to the one outlined here, was introduced in [9] to derive the formula for Ψ in a simplified fluid flow model.

3.3 The Last-Entrance Algorithm

This algorithm is very similar to the First-Exit algorithm. In fact, time reversal in the physical interpretation for the n-th iteration of one algorithm, leads to the physical interpretation of the other. We omit the details. The formula (10) is obtained by conditioning on the level y at which the process last enters the set $\mathcal{S}_2$, motivating the name of the algorithm. As with the First-Exit algorithm, there is no restriction on the maximum number of peaks in a sample path contributing to $\grave{\Psi}_n$ for $n > 1$. Conditioning analogous to the one outlined here was introduced in [21] to derive the formula for Ψ in a simplified fluid flow model.

3.4 Newton's Method

Observe that, with $F(Z) = T_{11}Z + ZT_{22} + ZT_{21}Z + T_{12}$, equation (2) can be written in the form $F(\Psi) = \mathbf{0}$. Newton's method is a well known method (see [20], for example), which can be used to approximate a solution of an equation of type $F(X) = \mathbf{0}$, provided that it exists, with the iteration

$$X_{k+1} = X_k - [F^{'}(X_k)]^{-1}F(X_k).$$

In [13] Guo established the important fact that Newton's method for the solution of (2) is equivalent to the iteration (6). This fact has a significant

consequence. In Lemma 1 in Section 3 we show that (11) is equivalent to (6). Consequently, the physical interpretation of algorithm (11) given below is in fact the physical interpretation of Newton's method [13] in the fluid flow model. This was previously not known.

Newton's method resembles the First-Exit and Last-Entrance algorithms in the sense that, in a sample path contributing to the $(n+1)$-st iteration, exactly as in the two former algorithms, first, an upward process takes place, followed by a transition $\mathcal{S}_1 \to \mathcal{S}_2$, after which a downward process begins, which takes place until the fluid level returns to zero. However, in Newton's method, paths in Ω_{n+1} for $n \geq 1$ are more complex than paths in $\acute{\Omega}_{n+1}$ or $\grave{\Omega}_{n+1}$. Observe that, in the First Exit algorithm, only the downward process has a complex structure, while in the Last-Entrance algorithm, only the upward process does. In Newton's method both upward and downward processes have a complex structure.

By (11) and (12) we have $\Psi_1 = P(1)$. The physical interpretation of Ψ_{n+1} is that, for $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, $[\Psi_{n+1}]_{ij}$ is the probability mass of all *distinct* sample paths contributing to $[\Psi]_{ij}$ in which, starting from level zero in phase i, for some $y \geq 0$,

- first, the upward process takes place. The fluid level moves to some level y and does so in some phase $k \in \mathcal{S}_1$. The probability of this is $[e^{(T_{11}+\Psi_n T_{21})y}]_{ik}$.
- Next, a transition from $k \in \mathcal{S}_1$ to some $l \in \mathcal{S}_2$ occurs, at a rate $[T_{12}]_{kl}$.
- Finally, the downward process takes place, until the fluid returns to level zero in some phase $j \in \mathcal{S}_2$. The probability of this is $[e^{(T_{22}+T_{21}\Psi_n)y}]_{lj}$.

The probability mass of all such paths is the (i,j)-th entry of the matrix (11). We illustrate this in Figure 2. Level y is the level at which the upward process ends and the downward process begins. Observe that level y in general is *not* the maximum level reached in a sample path. Also note that the choice of y is not unique as we discuss below. The parts of the sample path, within the upward process, which are indicated by thick solid lines are in Ω_2 shifted to level x_0 and Ω_1 shifted to level x_2 respectively. The parts of the sample path, within the downward process, which are indicated by thick dashed lines are in Ω_1 shifted to level x_3 and Ω_2 shifted to level x_1 respectively.

By conditioning on y, without checking whether the sample paths are

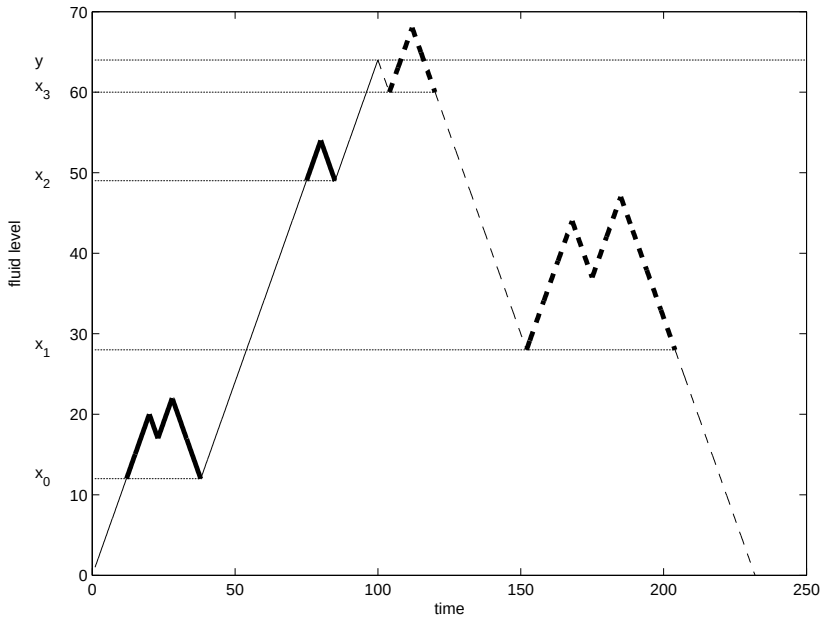

Figure 2: A sample path in Ω_3.

distinct, one might obtain the incorrect formula

$$\int_0^\infty e^{(T_{11}+\Psi_n T_{21})y} T_{12} e^{(T_{22}+T_{21}\Psi_n)y} dy. \tag{16}$$

This formula leads to multiple counting of the same sample path. For example, consider the sample path in Ω_2 illustrated in Figure 3. In this path, two possible candidates y_0 and y_1 for level y are indicated. Each choice of y is valid, as for each of them, the upward process and the downward processes conform to the definitions. To see this, observe that the parts indicated by thick solid and dashed lines are all in Ω_1 shifted to some level. Consequently, this sample path will contribute twice to the integral (16), for $n = 1$.

In contrast, the candidate for y illustrated in Figure 4 is invalid. This is because the part of the sample path indicated by the thick dashed line is not in Ω_1. Rather, it is in Ω_2.

The correct formula (11) is obtained from (16) by subtracting the matrix

$$\int_0^\infty e^{(T_{11}+\Psi_n T_{21})z} \Psi_n T_{21} \Psi_n e^{(T_{22}+T_{21}\Psi_n)z} dz. \tag{17}$$

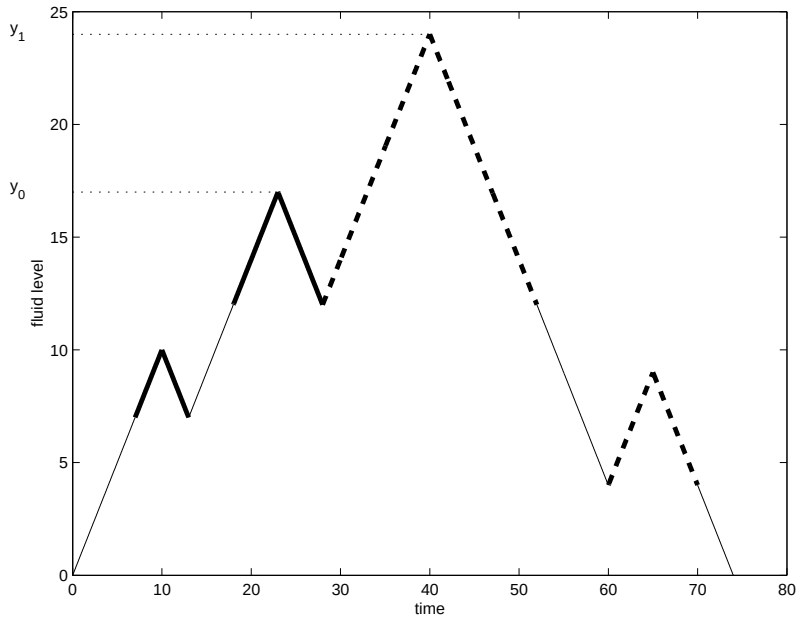

Figure 3: Possible candidates for y in the sample path in Ω_2.

This matrix records, for all $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, the total probability mass of all sample paths, in which, starting from level zero in phase i,

- first, the upward process takes place. The fluid level moves to z and does so in some phase $k \in \mathcal{S}_1$. The probability of this is $[e^{(T_{11}+\Psi_n T_{21})z}]_{ij}$.
- This is followed by a sample path in Ω_n shifted to level z, then a transition $\mathcal{S}_2 \to \mathcal{S}_1$, and again a sample path in Ω_n shifted to level z, ending in level z in some phase $l \in \mathcal{S}_2$. The probability of this is $[\Psi_n T_{21} \Psi_n]_{kl}$. For a given z, we shall use the notation $\zeta(z)$ for this part of a sample path.
- Finally, the downward process takes place, until the fluid level returns to zero, and does so in phase j. The probability of this is $[e^{(T_{22}+T_{21}\Psi_n)z}]_{lj}$.

We illustrate this in Figure 5 where $\zeta(z)$ is highlighted in bold. In the sample path in Figure 5, there is exactly one candidate for level z in the integral (17), for $n = 1$.

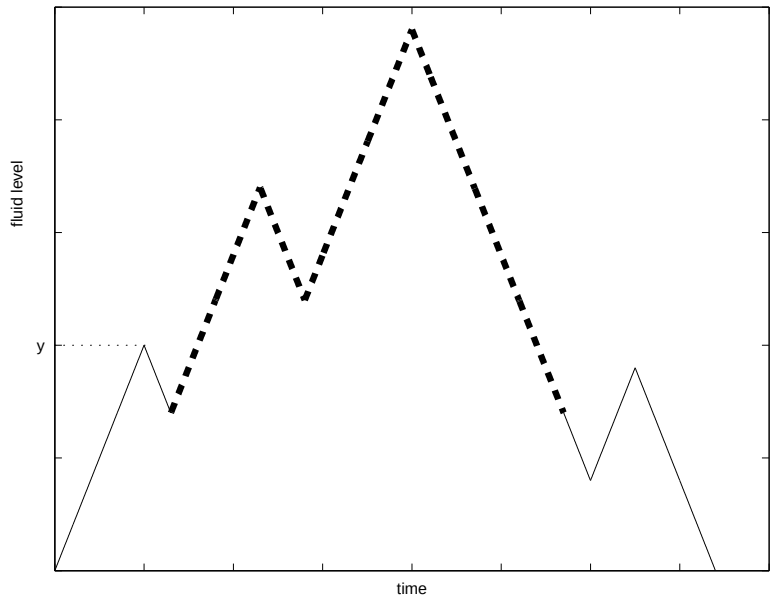

Figure 4: An invalid candidate for y in the sample path in Ω_2.

Note that the same sample path contributing to Ψ_2 has been illustrated in Figures 3, 4 and 5. Since this sample path is considered twice in (16) and once in (17), for $n = 1$, multiple counting of this sample path in the second iteration of Newton's method has been avoided.

This is true also for general n. Consider a sample path contributing to (17), for some n. Let ϱ be the set of all valid candidates for z in (17), with $z_1 = \min_{\varrho}(z)$. Note that, within the part $\zeta(z_1)$ of the sample path, all levels at which a transition $\mathcal{S}_2 \to \mathcal{S}_1$ occurs have a one to one correspondence to the elements of the set ϱ. Furthermore, within $\zeta(z_1)$, all levels at which a transition $\mathcal{S}_1 \to \mathcal{S}_2$ occurs have a one to one correspondence with all valid candidates for y in (16). Any other choice of y contradicts the definition of either the upward or downward process. Since, within $\zeta(z_1)$, the difference between the number of transitions $\mathcal{S}_1 \to \mathcal{S}_2$ and the number of transitions $\mathcal{S}_2 \to \mathcal{S}_1$ is equal to one, multiple counting of this sample path in the $(n+1)$-st iteration of Newton's method has been avoided.

Observe that there is no restriction on the maximum number of peaks in a sample path in Ω_n for $n > 1$. Compared to the First-Exit and Last-Entrance algorithms, Newton's method allows more flexibility in the shape of the sample path contributing to the n-th iteration. Furthermore, from

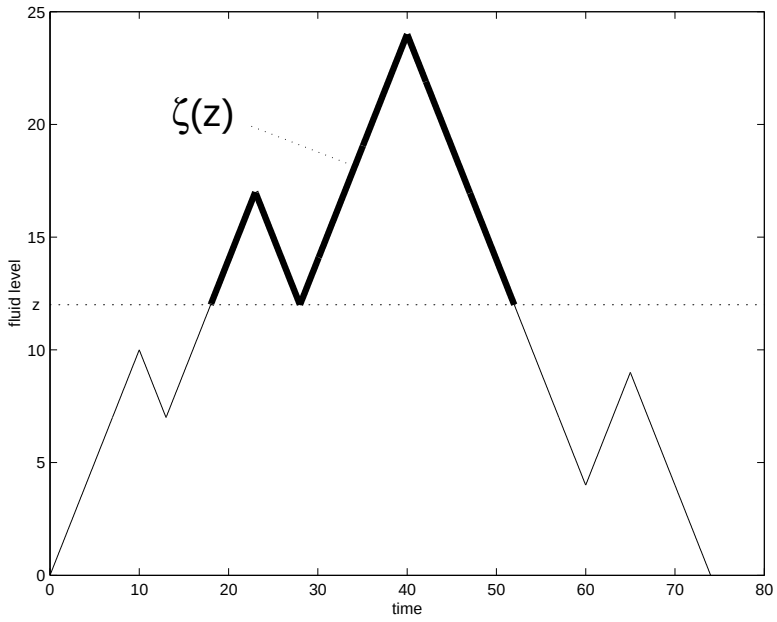

Figure 5: Level z in a sample path contributing to (17).

the analysis of the shape of the sample paths in the n-th iteration of these algorithms, we conclude that every path in $\hat{\Omega}_n$, $\acute{\Omega}_n$ or $\grave{\Omega}_n$ is also in Ω_n, and so the inequalities $\hat{\Psi}_n \leq \Psi_n$, $\acute{\Psi}_n \leq \Psi_n$, $\grave{\Psi}_n \leq \Psi_n$ must be satisfied. In Section 4 we shall show that these inequalities are, in fact, strict.

We emphasize that Newton's method applied to the fluid-flow model here differs from Newton's method applied to discrete-level QBDs discussed in [18]. In the fluid-flow model considered here, each iteration of Newton's method requires the solution of an equation of type $AX + XB = C$, for which efficient methods exist [4, 11]. We shall see later, in Section 6, that the complexity of this method is $\mathcal{O}(s^3)$, where s is the size of the set $\mathcal{S}$. In contrast, each iteration of Newton's method used to compute the $s \times s$ matrix G in [18] requires a solution of an equation of type $X - AXB = C$. The complexity of this task was reported in [18] to be $\mathcal{O}(s^6)$. An anonymous referee pointed out that the complexity of this task is $\mathcal{O}(s^3)$, when using the approach suggested in [10]. An interesting question is whether, in the context of approximating the matrix G in the QBDs, it would be more efficient to use Newton's method [18], implemented as described in [10], than the Logarithmic Reduction algorithm [19].

3.5 Asmussen's Iteration

The key to the physical interpretation of Asmussen's iteration [3, Theorem 3.1] is the first line of the expression for '$\alpha_{ij}^{(-+)}$' on page 30 of [3]. A simplified form of this expression was used to define a recursion for the calculation of $\alpha^{(-+)}(n+1)$ from $\alpha^{(-+)}(n)$. An analysis of all possible sample paths in $\alpha^{(-+)}(n)$, leads to the physical interpretation of the scheme. Note that the matrix $\alpha^{(-+)}$ is symmetrical to Ψ in the sense that, in the model which allows for negative fluid levels, $\alpha_{ij}^{(-+)}$ is the probability that the process starting from level zero in phase $i \in \mathcal{S}_2$ first returns to level zero and does so in phase $j \in \mathcal{S}_1$, while avoiding levels above zero. For the purpose of comparing this scheme with other algorithms, we give this interpretation in terms of Ψ. We shall use the notation Ψ_n^A and Ω_n^A in a manner analogous to our analysis of other algorithms.

The entry $[\Psi_1^A]_{ij}$ is the return probability of a journey in which, starting from level zero in phase i,

- the phase process remains in phase i, and so the fluid level increases, until the transition of the phase process $\mathcal{S}_1 \to \mathcal{S}_2$ occurs, at which point the fluid level begins to decrease,
- next, the phase process remains in the set $\mathcal{S}_2$ until the fluid level first reaches level zero and does so in phase j.

The entry $[\Psi_{n+1}^A]_{ij}$ is the return probability of a journey in which, starting from level zero in phase i, only two options are allowed. The first option is that

- the phase process remains in phase i, and so the fluid level increases, until the transition of the phase process $\mathcal{S}_1 \to \mathcal{S}_2$ occurs, at which point the fluid level begins to decrease, and then
- a process with generator $T_{22} + T_{21}\Psi_n^A$ begins, which takes place until the fluid level returns to 0 and does so in phase j.

The second option is that

- the phase process remains in phase i, and so the fluid level increases, until the transition of the phase process $\mathcal{S}_1 \to \mathcal{S}_1$ occurs, at say level x,
- next, a sample path in Ω_n^A shifted to level x occurs, and then

- a process with generator $T_{22} + T_{21}\Psi_n^A$ begins, which takes place until the fluid level first returns to 0 and does so in phase j.

In a sample path in Ω_n^A, in the initial excursion within the set $\mathcal{S}_1$, the phase process is allowed to visit at most n states in $\mathcal{S}_1$. In other words, at most $(n-1)$ transitions $\mathcal{S}_1 \to \mathcal{S}_1$ are allowed in the initial excursion within set $\mathcal{S}_1$. Further, in every subsequent such excursion, at most $(n-1)$ visits to states in $\mathcal{S}_1$ (or $(n-2)$ transitions $\mathcal{S}_1 \to \mathcal{S}_1$) are allowed (per excursion). This means that, an up-slope of every peak in Ω_n^A has a very restricted form, because of the restriction on the number of transitions within $\mathcal{S}_1$. Consequently, this iteration scheme might not be expected to perform well for processes in which many such transitions occur with high probability. In comparison, we have shown earlier that fixed-point iteration FP3 [13], which is restricted in terms of the maximum number of peaks in $\hat{\Psi}(n)$, has total flexibility on the up-slope of every peak. From the analysis of the shape of sample paths in the n-th iteration of the First-Exit algorithm and Asmussen's iteration [3, Theorem 3.1], we conclude that $\Omega_n^A \subseteq \acute{\Omega}_n$ and so $\Psi_n^A \leq \acute{\Psi}_n$. A rigorous proof of this fact is given in Section 4.

3.6 The Latouche-Ramaswami method

Below we give a brief summary of the physical interpretation of Ramaswami's construction [21], given by da Silva Soares and Latouche in [9]. The reader is referred to [9] for the details of this interpretation. Let $\vartheta \geq \max_{i\in\mathcal{S}} |\mathcal{T}_{ii}|$ and let the matrix $P = I + \frac{1}{\vartheta}\mathcal{T}$ be partitioned in a manner analogous to the partitioning of $\mathcal{T}$. Then, the entry $[\Psi]_{ij}$, $i \in \mathcal{S}_1$, $j \in \mathcal{S}_2$, is the first passage probability from state $(1, i)$ to state $(0, j)$, in the discrete time QBD with the transition matrices

$$A_0 = \begin{bmatrix} \frac{1}{2}I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad A_1 = \begin{bmatrix} \frac{1}{2}P_{11} & \mathbf{0} \\ P_{21} & \mathbf{0} \end{bmatrix}, \quad A_2 = \begin{bmatrix} \mathbf{0} & \frac{1}{2}P_{12} \\ \mathbf{0} & P_{22} \end{bmatrix}. \tag{18}$$

Furthermore, the G-matrix of this QBD (see, for example, Latouche and Ramaswami [19]) is

$$\begin{bmatrix} \mathbf{0} & \Psi \\ \mathbf{0} & P_{22} + P_{21}\Psi \end{bmatrix}.$$

In the Latouche-Ramaswami method, the quadratically convergent Logarithmic Reduction algorithm [19] is applied to this QBD. For the physical

interpretation of the Logarithmic Reduction algorithm, within the QBD environment, see [19]. As pointed out in [9], the QBD construction involves a disconnection from any reference to the fluid flow model. Nevertheless, we are able to make useful predictions about the performance of this method, using the fact that the Logarithmic Reduction algorithm is quadratically convergent with respect to the maximum level reached in a sample path contributing to the G-matrix in the QBD. We observe that the maximum level reached in the constructed QBD [21] is directly related to the number of transitions $\mathcal{S}_1 \to \mathcal{S}_1$ in the fluid flow process, in the following way.

As $\vartheta \to \infty$, A_0 remains constant, while the nonzero entries in A_1 and in A_2 decrease, with

$$A_1 \to \begin{bmatrix} \frac{1}{2}I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad A_2 \to \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}.$$

In practical terms this means that the constructed QBD can reach higher levels with a larger probability. Consequently, if ϑ is large, then

- the constructed QBD [21] will be a process with a large amount of probability mass associated with sample paths which reach very high levels,

and consequently, the Logarithmic Reduction algorithm will require many iterations to complete. We construct examples which illustrate this in Section 7.

It is possible to interpret the n-th iteration of the Latouche-Ramaswami method in terms of a transformed fluid flow process. First, we note that the Riccati equation (2) can be written in the form

$$(T_{11} + \Psi T_{21})\Psi + \Psi(-\vartheta I_{22}) = -T_{12} - \Psi(\vartheta I_{22} + T_{22}).$$

As in Section 2, we can solve this using the iteration,

$$(T_{11}+\Psi^*(k)T_{21})\Psi^*(k+1)+\Psi^*(k+1)(-\vartheta I_{22}) = -T_{12}-\Psi^*(k)(\vartheta I_{22}+T_{22}), \quad (19)$$

with $\Psi^*(0) = \mathbf{0}$. This, by an argument analogous to Lemma 1 in Section 3, is equivalent to

$$\Psi^*(k+1) = \int_0^\infty e^{(T_{11}+\Psi^*(k)T_{21})y}\Big[T_{12} + \Psi^*(k)(\vartheta I_{22} + T_{22})\Big]e^{-\vartheta I_{22}y}dy. \quad (20)$$

The integral (20) is in a convenient form for interpretation in terms of a fluid flow model. In Lemma 3 below, we show the connection between the Latouche-Ramaswami method and the iteration (20). Furthermore, by applying this lemma, we shall give the physical interpretation of the n-th iteration of the Latouche-Ramaswami method in terms of a transformed fluid flow model.

Lemma 3 *The n-th step of the Logarithmic Reduction algorithm [19] applied to the QBD constructed in [21], for $n \geq 0$, yields the matrix*

$$G_n = \begin{bmatrix} \mathbf{0} & \Psi^*(2^{n+1}-1) \\ \mathbf{0} & P_{22} + P_{21}\Psi^*(2^{n+1}-1) \end{bmatrix},$$

where $\Psi^*(n)$ satisfies the recursion in (20).

Proof
By [19], $G_n = G(2^{n+1}-1)$, where the matrix $G(k+1)$, for $k \geq 0$, is calculated by the linear progresssion algorithm in [19], in which

$$\begin{aligned} G(0) &= \mathbf{0}, \\ G(k+1) &= \left[I - A_1 - A_0 G(k)\right]^{-1} A_2. \end{aligned}$$

From this, it can easily be shown by mathematical induction that

$$G(k+1) = \begin{bmatrix} \mathbf{0} & \Psi^*(k+1) \\ \mathbf{0} & P_{22} + P_{21}\Psi^*(k+1) \end{bmatrix},$$

where $\Psi^*(k+1)$ satisfies equation (19), or equivalently (20). ■

Now, we give the physical interpretation of the n-th iteration of the Latouche-Ramaswami method, denoted by Ψ_n^{LR}. First, by Lemma 3,

$$G_0 = \begin{bmatrix} 0 & \Psi^*(1) \\ 0 & P_{22} + P_{21}\Psi^*(1) \end{bmatrix},$$

with

$$\Psi^*(1) = \int_0^\infty e^{T_{11}y} T_{12} e^{-\vartheta I_{22} y} dy.$$

The physical interpretation of $\Psi^*(1)$ is equivalent to the physical interpretation of the matrix $P(1)$, given in Section 3.1, for the process in which T_{22} is

replaced with $-\vartheta I_{22}$. Since it is a nonzero matrix, we denote this as the first iteration of the Latouche-Ramaswami method. We have $\Psi_1^{LR} = \Psi^*(1)$.

The $(n+1)$-st iteration of the Latouche-Ramaswami method is the matrix $\Psi^*(2^{n+1}-1)$, which is given by

$$\int_0^\infty e^{(T_{11}+\Psi^*(2^{n+1}-2)T_{21})y}\Big[T_{12}+\Psi^*(2^{n+1}-2)(\vartheta I_{22}+T_{22})\Big]e^{-\vartheta I_{22}y}dy.$$

The physical interpretation of $\Psi^*(2^{n+1}-1)$ is equivalent to the physical interpretation of the matrix $\grave{\Psi}_{2^{n+1}-1}$, given in Section 3.3, for the process in which T_{22} is replaced with $-\vartheta I_{22}$ and T_{12} with the nonnegative matrix $\Big[T_{12}+\Psi^*(2^{n+1}-2)(\vartheta I_{22}+T_{22})\Big]$. We have $\Psi_{n+1}^{LR} = \Psi^*(2^{n+1}-1)$.

Although our interpretation of the n-th iteration of the Latouche-Ramaswami method departs from the original process, as the generator $\mathcal{T}$ needs to be modified in each iteration, it is given within a fluid flow environment. This feature of our interpretation is useful in the analysis of the algorithm in terms of the properties of the fluid flow processes, and hence in making predictions about its performance, such as the one given earlier in this section.

Remark 2 An interesting observation is that, when the process is such that the equation $T_{22} = -\vartheta I_{22}$ is satisfied, then the matrices $\grave{\Psi}_m$ for the Last-Entrance algorithm and $\Psi^*(m)$ are identical for all $m \geq 1$. This can be easily verified by substituting $-\vartheta I_{22}$ for T_{22} in (20) and (10). An example of such a process is given in Section 7. We also have a simple proof (by induction) of the fact that $\Psi^*(m) \leq \grave{\Psi}_m$ for all $m \geq 1$, and any $\mathcal{T}$. From this it follows that $\Psi_n^{LR} \leq \grave{\Psi}_{2^n-1}$ for all $n \geq 1$.

Furthermore, our numerical experience suggests that

- if $T_{22} = -\vartheta I_{22}$, then $\Psi_n = \Psi_n^{LR}$ for all $n \geq 1$. The number of iterations required for Newton's and Latouche-Ramaswami methods is then equal in such cases.

- Alternatively, if $T_{22} \neq -\vartheta I_{22}$, then $\Psi_n > \Psi_n^{LR}$ for all $n \geq 1$. The number of iterations required for Newton's method is then smaller than the number of iterations required for the Latouche-Ramaswami method.

We have not been able to find an example in which the number of iterations required for Newton's method is larger than the number of iterations required for the Latouche-Ramaswami method. ■

Finally, we show the convergence of the algorithms (8)-(11) in the theorem below. The convergence of fixed-point iteration FP3 [13] was established earlier by Guo [13] via algebraic methods. We present here an alternative proof, in which we apply the physical interpretations of the algorithms. The convergence of Newton's method was also established by Guo [13], but under different assumptions,which require that $\Psi_1 > \mathbf{0}$ instead of only the irreducibility of $\mathcal{T}$, assumed here.

Theorem 1 *The matrices* $\hat{\Psi}_n$, $\acute{\Psi}_n$, $\grave{\Psi}_n$ *and* Ψ_n *converge to* Ψ *as* $n \to \infty$.

Proof
From the physical interpretation of Newton's method in Section 3.4, we know that, for all n, $\Psi_n \leq \Psi$, as every sample path in Ω_n contributes to Ψ. Furthermore, by Section 3.4, every sample path contributing to Ψ lies in Ω_n for some n. Hence, the convergence of Newton's method follows. The proof of the remaining results is analogous. ■

4 Comparisons

We now compare the algorithms (8) to (11) with respect to the number of iterations required. First, in Theorem 2 we compare the n-th and $(n+1)$-st iterations of the algorithms. Then, in Theorem 3 we establish the relationships between the matrices $\hat{\Psi}_n$, $\acute{\Psi}_n$, $\grave{\Psi}_n$, Ψ_n and Ψ_n^A,for all $n \geq 1$. The immediate consequence of these results is that, in comparison to fixed-point iteration FP3 [13], the First-Exit algorithm, the Last-Entrance algorithm, and Asmussen's iteration [3, Theorem 3.1], Newton's method [13] is the algorithm that requires the least number of iterations.

The results $\mathbf{0} \leq \Psi_1$ and $\Psi_n < \Psi_{n+1}$, $n \geq 1$, for Newton's method, are more general here than in Guo [13], as we assume only the irreducibility of $\mathcal{T}$. The results for Newton's method in Guo [13] require that $\Psi_1 > \mathbf{0}$, which is an unnecessary condition in our model, while the important assumption of the irreducibility of $\mathcal{T}$ is missing there. Similarly, the results for fixed-point iteration FP3 [13] here differ from the results in Guo [13], as we assume the irreducibility of $\mathcal{T}$ and thus can prove a strict inequality in at least one place in $\hat{\Psi}_n \leq \hat{\Psi}_{n+1}$, for all $n \geq 1$. The proofs in Guo [13] employ algebraic methods, while we translate the mathematical concepts into the physical interpretations of what happens in the process.

Theorem 2 *For $n \geq 1$ we have*

$$\acute{\Psi}_n < \acute{\Psi}_{n+1}, \quad \grave{\Psi}_n < \grave{\Psi}_{n+1}, \quad \Psi_n < \Psi_{n+1} \quad \textit{and} \quad \Psi_n^A < \Psi_{n+1}^A, \tag{21}$$

for $n \geq 0$ we have $\hat{\Psi}_n \leq \hat{\Psi}_{n+1}$ with a strict inequality in at least one place.
Furthermore, $\hat{\Psi}_m > \mathbf{0}$ for some $m \geq 1$ and

$$\mathbf{0} \leq \Psi_1^A \leq \hat{\Psi}_1 = \acute{\Psi}_1 = \grave{\Psi}_1 = \Psi_1 \quad \textit{with} \quad \mathbf{0} \neq \Psi_1^A.$$

Proof
These results may be directly derived from the physical interpretations of the algorithms given in the previous section. We shall present here the proof of some of these inequalities. The remaining results can be established in an analogous way. The inequalities are easy to prove, but care is needed when proving the strict inequalities.

- $\hat{\Psi}_m > \mathbf{0}$ for some $m \geq 1$.

  Let $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$. By the irreducibility of $\mathcal{T}$, i and j must communicate. Hence, there exists a finite sequence of one-step transitions $\mathcal{S} \to \mathcal{S}$, beginning with transition $i \to \mathcal{S}$ and ending with transition $\mathcal{S} \to j$, such that the corresponding entries in $\mathcal{T}$ are all positive. This sequence has a finite number $k \geq 1$ of transitions $\mathcal{S}_1 \to \mathcal{S}_2$. Therefore, from the physical interpretation of fixed-point iteration FP3 and (15),

  $$0 < [P(k)]_{ij} \leq \sum_{l=1}^{k} [P(l)]_{ij} \leq [\hat{\Psi}_k]_{ij}.$$

  This shows that for given $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, $[\hat{\Psi}_k]_{ij} > 0$ for some $k \geq 1$. Since we can choose m to be the maximum of all such k, taken over all i and j, the result follows.

- $\acute{\Psi}_n < \acute{\Psi}_{n+1}$ for all $n \geq 1$.

  From the physical interpretation of the First-Exit algorithm, $\acute{\Psi}_n \leq \acute{\Psi}_{n+1}$. We proceed by induction and first show that $\acute{\Psi}_1 < \acute{\Psi}_2$.

  Let $i \in \mathcal{S}_1$, $j \in \mathcal{S}_2$. By the irreducibility of $\mathcal{T}$, there exists a finite sequence of one-step transitions $\mathcal{S} \to \mathcal{S}$, beginning with transition $i \to \mathcal{S}$ and ending with transition $\mathcal{S} \to j$, such that the corresponding entries in $\mathcal{T}$ are all positive and there are at least two transitions $\mathcal{S}_1 \to \mathcal{S}_2$ in this sequence. Consequently, the set of sample paths such that

– starting from level zero in phase i, the process remains in the set $\mathcal{S}_1$ until it reaches some fluid level $y > 0$, which is directly followed by a transition $\mathcal{S}_1 \to \mathcal{S}_2$,

– next, the downward process with generator $T_{22} + T_{21}\acute{\Psi}_1$ begins, which takes place until the fluid level first reaches 0 and does so in phase j, and

– during this downward process the following event must occur at least once: a transition $\mathcal{S}_2 \to \mathcal{S}_1$ immediately followed by a path contributing to $\acute{\Psi}_1$

must have positive probability.

From the physical interpretation of the First-Exit algorithm, such paths contribute to $[\acute{\Psi}_2]_{ij}$. Also, since there are at least two peaks in them, they do not contribute to $[\acute{\Psi}_1]_{ij}$. Therefore, we have $[\acute{\Psi}_1]_{ij} < [\acute{\Psi}_2]_{ij}$ for all $i \in \mathcal{S}_1$, $j \in \mathcal{S}_2$, and so $\acute{\Psi}_1 < \acute{\Psi}_2$.

Now, suppose that $\acute{\Psi}_n < \acute{\Psi}_{n+1}$ for some $n \geq 1$. Then, the following set of paths must have positive probability. Paths such that:

– starting from level zero in phase i, the process remains in the set $\mathcal{S}_1$ until it reaches some fluid level $y > 0$, which is directly followed by a transition $\mathcal{S}_1 \to \mathcal{S}_2$,

– next, the downward process with generator $T_{22} + T_{21}\acute{\Psi}_{n+1}$ begins, which takes place until the fluid level reaches 0 and does so in phase j, and

– during this downward process the following event must occur at least once: a transition $\mathcal{S}_2 \to \mathcal{S}_1$ immediately followed by a path contributing to $\acute{\Psi}_{n+1}$ but not to $\acute{\Psi}_n$.

From the physical interpretation of the First-Exit algorithm, such paths contribute to $[\acute{\Psi}_{n+2}]_{ij}$ but not to $[\acute{\Psi}_{n+1}]_{ij}$. Therefore, $[\acute{\Psi}_{n+1}]_{ij} < [\acute{\Psi}_{n+2}]_{ij}$ for all $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, and so $\acute{\Psi}_{n+1} < \acute{\Psi}_{n+2}$. Consequently, the result follows by mathematical induction. ■

The result below was established earlier by Guo [14]. We present an alternative proof.

Corollary 1 Ψ *is a positive matrix.*

Proof
By Theorem 2, $\mathbf{0} < \Psi_n$ for all $n \geq 2$ and so, by Theorem 1, $\Psi > \mathbf{0}$. ■

Theorem 3 *For $n \geq 2$ we have*

$$\hat{\Psi}_n < \acute{\Psi}_n < \Psi_n, \quad \hat{\Psi}_n < \grave{\Psi}_n < \Psi_n, \quad \textit{and} \quad \Psi_n^A \leq \acute{\Psi}_n. \tag{22}$$

Proof
We show that $\hat{\Psi}_n < \acute{\Psi}_n$ for $n \geq 2$. The remaining results follow by analogous argument.

From the physical interpretation of the fixed-point iteration FP3 and the First-Exit algorithm, $\hat{\Psi}_1 = \acute{\Psi}_1$ and $\hat{\Psi}_2 \leq \acute{\Psi}_2$. By the irreducibility of $\mathcal{T}$, for all $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, paths which contribute to $[\acute{\Psi}_2]_{ij}$ and have at least three peaks in them, have positive probability mass. To show this fact, use the argument analogous to the earlier proof of $\acute{\Psi}_n < \acute{\Psi}_{n+1}$ in Theorem 2. From the physical interpretation of fixed-point iteration FP3, such paths do not contribute to $[\hat{\Psi}_2]_{ij}$. Therefore, $\hat{\Psi}_2 < \acute{\Psi}_2$.

Now, suppose that $\hat{\Psi}_n \leq \acute{\Psi}_n$ for some $n \geq 2$. By (21), we have $\mathbf{0} < \acute{\Psi}_n < \acute{\Psi}_{n+1}$. By the irreducibility of $\mathcal{T}$, the matrix T_{21} must have at least one positive entry, and so the matrix $T_{21}\acute{\Psi}_n$ must have at least one row with all positive entries. Therefore $\acute{\Psi}_n T_{21}\acute{\Psi}_n < \acute{\Psi}_{n+1}T_{21}\acute{\Psi}_n$, and hence

$$T_{12} + \hat{\Psi}_n T_{21}\hat{\Psi}_n \leq T_{12} + \acute{\Psi}_n T_{21}\acute{\Psi}_n < T_{12} + \acute{\Psi}_{n+1}T_{21}\acute{\Psi}_n.$$

Consequently,

$$\int_0^\infty e^{T_{11}y}(T_{12} + \hat{\Psi}_n T_{21}\hat{\Psi}_n)e^{T_{22}y}dy < \int_0^\infty e^{T_{11}y}(T_{12} + \acute{\Psi}_{n+1}T_{21}\acute{\Psi}_n)e^{T_{22}y}dy.$$

By (8), the left-hand side of this inequality equals $\hat{\Psi}_{n+1}$. Denote the right-hand side of this inequality by Z. By Theorem 9.2 in [7] and the fact that eigenvalues with the maximum real part of each of the matrices T_{11} and T_{22} are negative [6, Lemma 2], Z satisfies the equation

$$T_{11}Z + ZT_{22} + T_{12} + \acute{\Psi}_{n+1}T_{21}\acute{\Psi}_n = \mathbf{0}.$$

By (4), $Z = \acute{\Psi}_{n+1}$ is the solution of the above equation. Hence, $\hat{\Psi}_{n+1} < \acute{\Psi}_{n+1}$ and so, by mathematical induction, we have that $\hat{\Psi}_n < \acute{\Psi}_n$ for all $n \geq 2$. ■

5 Convergence Rates

In this section we analyze convergence of the algorithms defined in Section 3. We refer the reader to [20] for a summary of basic results on convergence. Although the results of this section are achieved by algebraic methods, they have important physical implications.

Let (ν_1, ν_2) be the stationary probability vector of the phase process, that is, the Markov process governed by the generator $\mathcal{T}$, partitioned according to $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, and define

$$\mu_F = \nu_1 \mathbf{e} - \nu_2 \mathbf{e}. \tag{23}$$

Then μ_F is known as the *recurrence measure* of the process $(X(t), \varphi(t))$. We shall see in this analysis that the rates of convergence of each algorithm are closely related to the recurrence measure μ_F. Furthermore, the worst performance of each algorithm can be expected when the process $(X(t), \varphi(t))$ is null recurrent, that is when $\mu_F = 0$.

It is important to remember in this analysis, that an algorithm having the best convergence rate is not necessarily the best performer. A quadratically convergent algorithm may require a large number of iterations. For instance, in Example 3 of Section 7, the Latouche-Ramaswami method requires the highest number of iterations, even though this method is quadratically convergent. Further, the overall performance of an algorithm in terms of computation time depends on the complexity of each iteration and the number of iterations required. We shall analyze the overall performance of the algorithms in Section 7.

Remark 3 Whether $\mu_F = 0$ or not, depends on the singularity or nonsingularity of the matrix

$$\Pi = (T_{22} + T_{21}\Psi)^t \oplus (T_{11} + \Psi T_{21}). \tag{24}$$

By [6], if for a given $\mathcal{T}$, the matrix Π is nonsingular, then the process $(X(t), \varphi(t))$ characterized by the generator $\mathcal{T}$ is positive recurrent ($\mu_F < 0$) or transient ($\mu_F > 0$), while if the matrix Π is singular, then the process $(X(t), \varphi(t))$ is null recurrent. In physical terms, the singularity or nonsingularity of the matrix Π can be explained in the following way. Consider two processes $(X_1, \varphi(t))$ and $(X_2, \varphi(t))$ with generators $U_1 = (T_{11} + \Psi T_{21})$ and $U_2 = (T_{22} + T_{21}\Psi)$ respectively. If both these processes are recurrent, then

by [22] the matrices U_1 and U_2 both have 0 as their dominant eigenvalue. Hence, by [12] the matrix Π is singular, which in turn implies the null recurrence of the fluid flow process $(X(t), \varphi(t))$. Alternatively, by [22] and [12], if one of the processes is recurrent and the other transient, then the matrix Π is nonsingular. In summary

- if both $(X_1, \varphi(t))$ and $(X_2, \varphi(t))$ are recurrent, then the fluid flow process $(X(t), \varphi(t))$ is null recurrent,
- if the process $(X_1, \varphi(t))$ is transient and $(X_2, \varphi(t))$ is recurrent, then the fluid flow process $(X(t), \varphi(t))$ is positive recurrent, while
- if the process $(X_1, \varphi(t))$ is recurrent and $(X_2, \varphi(t))$ is transient, then the fluid flow process $(X(t), \varphi(t))$ is transient. ■

Let us recall a few basic concepts from [20], which we will need in this analysis. First, we define a measure of the rate of convergence of iterative processes. Let $p \geq 1$. Then, an R-factor for p of the sequence $\{A_k\}$ converging to A, as $k \to \infty$, is

$$R_p\{A_k\} = \begin{cases} \limsup_{k\to\infty} ||A_k - A||^{1/k} & \text{if } p = 1, \\ \limsup_{k\to\infty} ||A_k - A||^{1/p^k} & \text{if } p > 1. \end{cases} \tag{25}$$

Now, if $0 < R_1\{A_k\} < 1$, then we say that the convergence is R-linear, while if $R_1\{A_k\} = 1$ or $R_1\{A_k\} = 0$, then the convergence is R-sublinear or R-superlinear, respectively. If $0 < R_2\{A_k\} < 1$, then the convergence is R-quadratic. When $R_1\{A_k\} > 0$, R-quadratic convergence is not possible.

Let $\hat{R}_1$ be the R-factor for 1 [20] of fixed-point iteration FP3 [13], that is $\hat{R}_1 = \limsup_{n\to\infty}(||\hat{\Psi}_n - \Psi||)^{\frac{1}{n}}$. Similarly, let $\acute{R}_1$, $\grave{R}_1$, R_1^A, R_1 and R_1^{LR} be the R-factors for 1 of the First-Exit algorithm, the Last-Entrance algorithm, Asmussen's iteration [3, Theorem 3.1], Newton's method [13] and the Latouche-Ramaswami method, respectively.

By Guo [13], the convergence of Newton's method is R-quadratic and so $R_1 = 0$, when the matrix Π is nonsingular. By Remark 3, this occurs when $(X(t), \varphi(t))$ is positive recurrent or transient. By Theorem 7.2.3 in [19], the recurrence measure μ_{LR} of the discrete QBD [21, 9] in the Latouche-Ramaswami method is

$$\mu_{LR} = \pi(A_0 - A_2)\mathbf{e},$$

where π is the stationary probability vector of the Markov process with generator $Q = A_0 + A_1 + A_2$, with A_0, A_1, A_2 given by (18). It is easy to show that

$$\mu_{LR} = \mu_F/(1 + \nu_1 \mathbf{e}). \tag{26}$$

Consequently, by [15], the convergence of the Logarithmic Reduction algorithm applied in the Latouche-Ramaswami method is also R-quadratic and $R_1^{LR} = 0$, when the process $(X(t), \varphi(t))$ is positive recurrent or transient. Note however that, when the process $(X(t), \varphi(t))$ is null recurrent (and so the QBD is also null recurrent), the convergence may be linear. Guo [13, 15] conjectured that when the QBD is null recurrent, the convergence of both Newton's method and the Latouche-Ramaswami method is R-linear with R-factor equal to 1/2.

Theorem 4 below determines the convergence of the First-Exit algorithm, the Last-Entrance algorithm and Asmussen's iteration [3, Theorem 3.1]. For completeness, we have included Guo's result for fixed-point iteration FP3 [13]. We have added our proof of the fact that $\hat{R}_1$ is a positive number, which was not shown in [13]. This fact confirms that neither R-superlinear nor R-quadratic convergence is possible. The consequence of this theorem is that, when the process $(X(t), \varphi(t))$ is positive recurrent or transient, then the convergence of fixed-point iteration FP3 [13], the First-Exit algorithm, the Last-Entrance algorithm and Asmussen's iteration [3, Theorem 3.1] is R-linear, while when the process $(X(t), \varphi(t))$ is null recurrent, then this convergence is R-sublinear.

We introduce the notation $\rho(A)$ for the spectral radius of matrix A. From (22) we immediately have

$$\hat{R}_1 \geq \acute{R}_1, \quad \hat{R}_1 \geq \grave{R}_1, \quad \text{and} \quad R_1^A \geq \acute{R}_1, \tag{27}$$

and we are interested in the exact values of these R-factors. This is established in the next theorem.

Theorem 4 *If the process $(X(t), \varphi(t))$ is positive recurrent or transient then*

$$\begin{aligned}
0 < \hat{R}_1 &= \rho(-[T_{22}^t \oplus T_{11}]^{-1}[(T_{21}\Psi)^t \oplus \Psi T_{21}]) < 1, \\
0 < \acute{R}_1 &= \rho(-[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]^{-1}[\mathbf{0}_{22} \oplus \Psi T_{21}]) < 1, \\
0 < \grave{R}_1 &= \rho(-[T_{22}^t \oplus (T_{11} + \Psi T_{21})]^{-1}[(T_{21}\Psi)^t \oplus \mathbf{0}_{11}]) < 1,
\end{aligned}$$

and

$$0 < \min_{i \in \mathcal{S}_1} \rho(-[(T_{22} + T_{21}\Psi + \mathcal{T}_{ii} I_{22})^t \oplus \mathbf{0}_{11}]^{-1}[\mathbf{0}_{22} \oplus (T_{11} + \Psi T_{21} - \mathcal{T}_{ii} I_{11})])$$

$$
\begin{aligned}
&\leq R_1^A \\
&\leq \max_{i \in \mathcal{S}_1} \rho(-[(T_{22} + T_{21}\Psi + \mathcal{T}_{ii} I_{22})^t \oplus \mathbf{0}_{11}]^{-1}[\mathbf{0}_{22} \oplus (T_{11} + \Psi T_{21} - \mathcal{T}_{ii} I_{11})]) \\
&< 1.
\end{aligned}
$$

Otherwise, $\hat{R}_1 = \acute{R}_1 = \grave{R}_1 = R_1^A = 1$.

Proof

First, we shall consider the First-Exit algorithm. The proof of the remaining results is similar. For any $s_1 \times s_2$ matrix Z let $\mathcal{G}(Z)$ be the unique solution of

$$T_{11}\mathcal{G}(Z) + \mathcal{G}(Z)(T_{22} + T_{21}Z) = -T_{12}. \tag{28}$$

By (4) and Theorem 1, $\acute{\Psi}_{n+1} = \mathcal{G}(\acute{\Psi}_n)$, which defines a one-step stationary iteration [20], and $\Psi = \mathcal{G}(\Psi)$. The first F-derivative of the function $\mathcal{G}(Z)$ at Ψ, denoted by $\mathcal{G}'_\Psi$ is a *linear operator* such that

$$\lim_{||H|| \to 0} \frac{||\mathcal{G}(\Psi + H) - \mathcal{G}(\Psi) - \mathcal{G}'_\Psi(H)||}{||H||} = 0, \tag{29}$$

see, for example, Definition 3.1.5 of [20]. If the F-derivative exists, then it is unique. We are interested in $\rho(\mathcal{G}'_\Psi)$, as by [20], this is useful in calculating $\acute{R}_1$. Our approach is similar to the proof of Theorem 3.2 in Guo and Laub [16]. First, we determine $\mathcal{G}'_\Psi$.

Consider the linear operator $F(H)$ which, for any $s_1 \times s_2$ matrix H, is defined to be the unique solution of

$$T_{11}F(H) + F(H)(T_{22} + T_{21}\Psi) = -\Psi T_{21} H. \tag{30}$$

By evaluating equation (28) at $Z = \Psi + H$ and then subtracting equations (28) evaluated at $Z = \Psi$ and (30), we have

$$
\begin{aligned}
T_{11}\frac{\Big(\mathcal{G}(\Psi + H) - \mathcal{G}(\Psi) - F(H)\Big)}{||H||} \quad &+ \quad \frac{\Big(\mathcal{G}(\Psi + H) - \mathcal{G}(\Psi) - F(H)\Big)}{||H||}(T_{22} + T_{21}\Psi) \\
&= \frac{\Big(\Psi - \mathcal{G}(\Psi + H)\Big)T_{21}H}{||H||}.
\end{aligned}
$$

By Theorem 9.2 in [7], we can express this in the integral form

$$\frac{\Big(\mathcal{G}(\Psi + H) - \mathcal{G}(\Psi) - F(H)\Big)}{||H||} = \int_0^\infty e^{T_{11}u}\frac{\Big(\mathcal{G}(\Psi + H) - \Psi\Big)T_{21}H}{||H||}e^{(T_{22}+T_{21}\Psi)u}du$$

and hence verify that

$$\lim_{||H||\to 0} \frac{||\mathcal{G}(\Psi + H) - \mathcal{G}(\Psi) - F(H)||}{||H||} = 0.$$

Thus, by (29), $\mathcal{G}'_{\Psi}(H) = F(H)$.

The real number λ is an eigenvalue of the operator $\mathcal{G}'_{\Psi}$ if and only if $\mathcal{G}'_{\Psi}(H) = \lambda H$ for some $H \neq \mathbf{0}$. Assume that such λ and H exist, and so, by (30),

$$-\Psi T_{21} H = \lambda[T_{11}H + H(T_{22} + T_{21}\Psi)]. \tag{31}$$

This is equivalent to

$$-[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]^{-1}[\mathbf{0}_{22} \oplus \Psi T_{21}]vecH = \lambda vecH, \tag{32}$$

where the operator $vecH$ was defined earlier as the ordered stack of the columns of H. Consequently

$$\rho(\mathcal{G}'_{\Psi}) = \rho(-[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]^{-1}[\mathbf{0}_{22} \oplus \Psi T_{21}]). \tag{33}$$

Now, suppose that the process $(X(t), \varphi(t))$ is positive recurrent or transient. Then, by Remark 3, one of the processes $(X_1(t), \varphi(t))$ or $(X_2(t), \varphi(t))$ must be recurrent, while the othermust be transient. Suppose that $(X_1(t), \varphi(t))$ is recurrent and $(X_2(t), \varphi(t))$ is transient. The proof for the alternative case is analogous. By [22], we know that then $\rho(T_{11} + \Psi T_{21}) = 0$ and $\rho(T_{22} + T_{21}\Psi)^t = \rho(T_{22} + T_{21}\Psi) < 0$. By [12], if $\{\alpha_i\}$ and $\{\beta_j\}$ are the eigenvalues of $(T_{22} + T_{21}\Psi)^t$ and $T_{11} + \Psi T_{21}$, respectively, then $\{\alpha_i + \beta_j\}$ are the eigenvalues of $(T_{22} + T_{21}\Psi)^t \oplus (T_{11} + \Psi T_{21})$. Hence, $\rho(\Pi) < 0$, where Π is defined in Remark 3. Note that Π is an ML-matrix [23]. Furthermore, as $\mathcal{T}$ is irreducible, so are $(T_{22} + T_{21}\Psi)^t$ and $T_{11} + \Psi T_{21}$. Therefore, Π is irreducible. Consequently, from [23, Theorem 2.6] it follows that $-\Pi^{-1} > 0$. We can verify that

$$-\Pi = -[(T_{22} + T_{21}\Psi)^t \oplus T_{11}] - [\mathbf{0}_{22} \oplus \Psi T_{21}] \tag{34}$$

is a regular splitting [24] of the matrix $-\Pi$. From [24, Theorem 3.15] and (33) it follows that $0 < \rho(\mathcal{G}'_{\Psi}) < 1$. The equality $\acute{R}_1 = \rho(\mathcal{G}'_{\Psi})$ can be shown by using an argument similar to the one used in the proof of Theorem 3.2 in Guo and Laub [16]. We have thus shown that, if the process $(X(t), \varphi(t))$ is positive recurrent or transient, then

$$0 < \acute{R}_1 = \rho(-[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]^{-1}[\mathbf{0}_{22} \oplus \Psi T_{21}]) < 1.$$

Alternatively, suppose that the process $(X(t), \varphi(t))$ is null recurrent. Then, by Remark 3 the matrix Ψ is singular, and so $\Pi vecH = \mathbf{0}$ for some $H \neq \mathbf{0}$. By (34),

$$\mathbf{0} = \Pi vecH = -[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]vecH - [\mathbf{0}_{22} \oplus \Psi T_{21}]vecH.$$

Consequently,

$$-[(T_{22} + T_{21}\Psi)^t \oplus T_{11}]^{-1}[\mathbf{0}_{22} \oplus \Psi T_{21}]vecH = vecH,$$

and hence, by (32) , we have $\rho(\mathcal{G}_\Psi^{'}) = 1$. Finally, to see that $\acute{R}_1 = \rho(\mathcal{G}_\Psi^{'})$, in the null-recurrent case, modify the generator $\mathcal{T}$ by putting $T_{12}^{(k)} = [(k+1)/k]T_{12}$, $k \geq 1$, so that the new process $(X^{(k)}(t), \varphi^{(k)}(t))$ is positive recurrent. Next, repeat the argument for the positive recurrent process $(X^{(k)}(t), \varphi^{(k)}(t))$ and then take the limit as $k \to \infty$. This completes the proof of the result for the First-Exit algorithm. Since we have shown that $\acute{R}_1 > 0$, by (27) it follows that $\hat{R}_1 > 0$ and $R_1^A > 0$. ■

6 The Complexity of the n-th Iteration

The performance of an algorithm depends on two important factors: the number of iterations required for the algorithm to converge close enough to the solution and numerical complexity of each iteration. In Section 4 we considered the first factor, and determined that the number of iterations required for Newton's method is smaller than the number for fixed-point iteration FP3 [13], the First-Exit algorithm, the Last-Entrance algorithm, and Asmussen's iteration [3, Theorem 3.1]. Furthermore, we indicated that our numerical experience suggests that the number of iterations required for Newton's method is also less than or equal to the number of iterations required for the Latouche-Ramaswami method, with equality if $T_{22} = -\vartheta I_{22}$. We now consider the second factor.

The estimates below are given for the iterations $k \geq 2$. By [19], the complexity of each iteration of the Latouche-Ramaswami method is given by $\mathcal{W}_{LR} \approx \frac{50}{3}s^3$ (flops), where $s = |\mathcal{S}|$. Our estimate of the complexity of each iteration of the Bini-Meini-Ramaswami method is $\mathcal{W}_{BMR} \approx \frac{44}{3}s^3$. In [13], Guo gave the complexity of the Newton's method and fixed-point iteration FP3 in the special case when $|\mathcal{S}_1| = |\mathcal{S}_2|$. We now calculate the complexity of

each iteration of the five algorithms: $\mathcal{W}_N$ (Newton's method), $\mathcal{W}_{FP3}$ (fixed-point iteration FP3 [13]), $\mathcal{W}_{FX}$ (the First-Exit algorithm), $\mathcal{W}_{LE}$ (the Last-Entrance algorithm), and $\mathcal{W}_A$ (Asmussen's iteration [3, Theorem 3.1]). In this calculation, we shall use the fact that the sum of two $m \times n$ matrices requires mn operations, and that the product of an $m \times n$ matrix A and $n \times q$ matrix B requires $2mnq - mq$ operations [5].

Let $m = |\mathcal{S}_1|$ and $n = |\mathcal{S}_2|$. Then, in Newton's method, computing the matrix $T_{11} + \Psi_n T_{21}$ requires $2nm^2$ operations, computing the matrix $T_{22} + T_{21}\Psi_n$ requires $2mn^2$ operations, and computing the matrix $-T_{12} + \Psi_n T_{21}\Psi_n$ requires $2mn^2 + 2nm^2 - m^2$ operations. Suppose that the matrix Ψ_{n+1} is computed from (6) by using the algorithm in [4]. The complexity of this calculation is about $20m^3 + 20n^3 + 5(m^2n + n^2m)$ for large m and n [11]. Consequently,

$$\begin{aligned}
\mathcal{W}_N &\approx 20m^3 + 20n^3 + 9(m^2n + n^2m), && (35)\\
\mathcal{W}_{FP3} &\approx 7(m^2n + n^2m) - m^2, && (36)\\
\mathcal{W}_{FX} &\approx 20n^3 + 5m^2n + 7n^2m, && (37)\\
\mathcal{W}_{LE} &\approx 20m^3 + 7m^2n + 5n^2m. && (38)
\end{aligned}$$

The estimates for the first iteration, which requires more operations, are larger.

In Asmussen's iteration [3, Theorem 3.1], each iteration can be computed by solving equations $x_i A_i = b_i$ for $i = 1, \ldots, m$, where the vector x_i is the i-th row of the matrix Ψ^A_{n+1}, the matrix $A_i = -\mathcal{T}_{ii} I_{22} - T_{22} - T_{21}\Psi^A_n$, and the vector b_i is the i-th row of the matrix $T_{12} + T_{11}\Psi^A_n - \mathcal{T}_{ii}\Psi^A_n$. For each i, computing the matrix A_i requires $2n^2m + n$ operations. Computing the vector b_i requires $2mn + n$ operations. The complexity of computing the vector x_i from the equation $x_i A_i = b$ is $\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{1}{6}n$ [17]. Consequently, as there are m equations, we have

$$\mathcal{W}_A \approx \Big(\frac{2}{3}n^3 + \frac{3}{2}n^2 + 2n^2m + 2mn + \frac{11}{6}n\Big)m.$$

Assuming constant $s = m + n$, Asmussen's iteration performs best for small n or m. In the case $m = n = \frac{s}{2}$ we have $\mathcal{W}_A \approx \frac{1}{6}s^4$, $\mathcal{W}_{FP3} \approx \frac{7}{4}s^3$, $\mathcal{W}_{FX} \approx 4s^3$, $\mathcal{W}_{LE} \approx 4s^3$, $\mathcal{W}_N \approx \frac{29}{4}s^3$ and

$$\mathcal{W}_{FP3} < \mathcal{W}_{FX} = \mathcal{W}_{LE} < \mathcal{W}_N < \mathcal{W}_{BMR} < \mathcal{W}_{LR}.$$

7 Examples

Finally, we ask what is the overall performance of the algorithms discussed in this paper. The answer to this question is that it depends on the nature of the fluid flow process considered in the analysis. In the following examples, we shall see that Newton's method is the most reliable general performer in all circumstances. We also have an example in which Asmussen's iteration [3, Theorem 3.1] is the best performer. These examples have been chosen to illustrate the predictions that we made based on the physical interpretations of the algorithms in Section 3. All examples are indicative of the expected performance of the algorithms in similar situations.

We shall use the uniform notation $\Psi(n)$ for the matrix obtained in the n-th step. For positive recurrent or null recurrent processes, the inequality

$$||\mathbf{e} - \Psi(n)\mathbf{e}||_\infty \leq \epsilon \times \mathbf{e}$$

is used as the stopping criterion with $\epsilon = 1e-05$ in all such examples. For transient processes, we use the inequality

$$||\Psi(n+1)\mathbf{e} - \Psi(n)\mathbf{e}||_\infty \leq \epsilon \times \mathbf{e}$$

as the stopping criterion, with $\epsilon = 1e-09$.

We have implemented all the algorithms in MATLAB. Average CPU times were obtained by running each algorithm 100 times, unless indicated otherwise. We caution against paying too much attention to these CPU times, because the algorithms are likely to have been differentially affected by the inherent optimization features of MATLAB. To give a fair comparison, each of the algorithms would have to be implemented in comparable code.

In Example 1 below, we consider a null recurrent process with equal off-diagonal entries in $\mathcal{T}$. The fixed-point iteration FP3, the First-Exit algorithm (FX), the Last-Entrance algorithm (LE), and Asmussen's iteration (A) all require so many iterations that it is impractical to use them. Newton's method (N) requires less iterations than both the Latouche-Ramaswami method (LR) and the Bini-Meini-Ramaswami method (BMR).

Example 2 is a modified version of Example 1, created so as to obtain large off-diagonal entries in T_{11}. In such processes, sample paths in Ψ with very many single transitions $\mathcal{S}_1 \to \mathcal{S}_1$ will have a large total probability mass. As predicted in Section 3.5, this makes the Asmussen's iteration extremely slow.

Note that ϑ is large. As predicted in Section 3.6, this results in an increased number of iterations required for both QBD methods. The performance of the remaining algorithms is comparable to Example 1.

Example 3 illustrates a positive recurrent process with no off-diagonal entries in T_{11} and hence no single transitions $\mathcal{S}_1 \to \mathcal{S}_1$, and $s_1 << s_2$. For such processes, Asmussen's iteration is indeed the best performer, due to its small complexity per iteration (see Section 6). Note that, due to the large value of ϑ, the number of iterations required in both QBD methods are high, even though the process is *strongly* positive recurrent and hence convergence of these algorithms is quadratic!

In Example 4 we consider a transient process with $T_{22} = -\vartheta I_{22}$. Observe that the n-th iterations of Newton's and Latouche-Ramaswami methods are equal and hence the number of iterations required for both methods is the same. We note that in all our numerical experience, whenever $T_{22} = -\vartheta I_{22}$, then the number of iterations in Newton's method was *equal to* the number of iterations in the Latouche-Ramaswami method, and *smaller than* the latter, otherwise.

In the tables below an asterisk next to a method indicates that the algorithm did not converge in the given number of iterations.

Example 1 Consider the null recurrent process with generator

$$\mathcal{T} = \left[\begin{array}{rr|rr} -0.003 & 0.001 & 0.001 & 0.001 \\ 0.001 & -0.003 & 0.001 & 0.001 \\ \hline 0.001 & 0.001 & -0.003 & 0.001 \\ 0.001 & 0.001 & 0.001 & -0.003 \end{array}\right].$$

The outcome of using the various algorithms is as follows:

Algorithm	Number of iterations	Average CPU times	Error
FP3*	50000	21.5500	3.9990e-05
FX*	50000	23.2100	2.0000e-05
LE*	50000	23.6500	2.0000e-05
N	17	0.0089	0.7629e-05
A*	50000	14.8700	2.9999e-05
LR	18	0.0064	0.5722e-05
BMR	19	0.0074	0.5722e-05

Note that, by Section 6, the total complexity (including all the iterations) of Newton's method for this example is about eight times smaller than the total complexity of the Bini-Meini-Ramaswami method and about ten times smaller than the total complexity of the Latouche-Ramaswami method. Theoretically, Newton's method should perform much better than both QBD methods. However, this theoretical advantage is not reflected in the average CPU time observed in this example. In fact, the average CPU time is higher for Newton's method than for the two other methods. As mentioned above, we conjecture that this is because the algorithm has been differentially affected by the inherent optimisation features of MATLAB.

Example 2 Consider the null recurrent process with generator

$$\mathcal{T} = \left[\begin{array}{rr|rr} -100.002 & 100 & 0.001 & 0.001 \\ 100 & -100.002 & 0.001 & 0.001 \\ \hline 0.001 & 0.001 & -0.003 & 0.001 \\ 0.001 & 0.001 & 0.001 & -0.003 \end{array}\right].$$

The outcome of using the various algorithms is as follows:

Algorithm	Number of iterations	Average CPU times	Error
FP3*	50000	21.8300	4.0000e-05
FX*	50000	23.2600	2.0016e-05
LE*	50000	24.0000	2.0006e-05
N	17	0.0090	0.7649e-05
A*	500000	150.3100	990.2951e-05
LR	29	0.0101	0.9343e-05
BMR	30	0.0113	0.9321e-05

Example 3 Consider the strongly positive recurrent ($\mu_F = -0.8000$) process with $s_1 = 2$, $s_2 = 18$, and with all off-diagonal entries in T_{11} equal to 0, all entries in T_{12} equal to 0.001, all entries in T_{21} equal to 0.001, and all off-diagonal entries in T_{22} equal to 10. (All diagonal entries are such that the row sums are equal to 0). We observe that

Algorithm	Number of iterations	Average CPU times	Error
FP3	7	0.0229	0.6008e-05
FX	6	0.0214	0.1673e-05
LE	6	0.0201	0.1673e-05
N	3	0.0111	0.0186e-05
A	6	0.0052	0.1673e-05
LR	17	0.0151	0.3905e-05
BMR	18	0.0170	0.3905e-05

The average CPU time is the best for the Asmussen's iteration, which is the best performing algorithm for this example.

Example 4 Consider the weakly transient ($\mu_F = 0.0169$) process with generator

$$\mathcal{T} = \left[\begin{array}{cc|cc} -0.0030 & 0.0001 & 0.0019 & 0.0010 \\ 0.0001 & -0.0030 & 0.0019 & 0.0010 \\ \hline 0.0015 & 0.0015 & -0.0030 & 0 \\ 0.0029 & 0.0001 & 0 & -0.0030 \end{array}\right].$$

The outcome of using the various algorithms is as follows:

Algorithm	Number of iterations	Average CPU times	Error
FP3	770	0.3456	0.9991e-09
FX	411	0.2001	0.9874e-09
LE	411	0.2045	0.9874e-09
N	10	0.0053	0.9651e-09
A	424	0.1344	0.9842e-09
LR	10	0.0048	0.9651e-09
BMR	11	0.0048	0.9329e-09

In this example the matrices Ψ_n and Ψ_n^{LR} are identical for all n.

8 Conclusion

We have considered and analyzed several algorithms that can be used to calculate return probabilities for a class of fluid flow models. We have given the physical interpretation of each algorithm within the fluid flow environment, and compared them with respect to the number of iterations, numerical complexity, and convergence. We conclude that theoretically Newton's

method [13] is the most reliable of these methods, because of the small number of iterations required and $\mathcal{O}(s^3)$ complexity. However, the implementation of this method is slightly more difficult. The number of iterations required for other algorithms may be large, depending on the physical properties of the process under consideration. In summary:

- When the process is null recurrent, weakly transient or weakly positive recurrent, then the fixed-point iteration FP3 [13], the First-Exit algorithm, the Last-Entrance algorithm and Asmussen's iteration [3, Theorem 3.1] require a large number of iterations.
- When off-diagonal entries in T_{11} are large, then Asmussen's iteration [3, Theorem 3.1] requires a large number of iterations. This behaviour is independent of the recurrence measure of the process.
- When ϑ is large, then both QBD methods require many iterations. This is also independent of the recurrence measure of the process.
- In Section 4 we proved that Newton's method [13] requires less iterations than fixed-point iteration FP3 [13], the First-Exit algorithm, the Last-Entrance algorithm and Asmussen's iteration [3].
- We do not have an example in which Newton's method [13] requires more iterations than QBD methods. Our numerical experience suggests that when $T_{22} = -\vartheta I_{22}$, then the Latouche-Ramaswami and Newton's methods are equivalent and that Newton's method irequires fewer iterations otherwise.

Acknowledgement

The authors would like to thank the Australian Research Council for funding this research through Discovery Grant number DP0209921. Also, they would like to thank the anonymous referees for making a number of constructive suggestions.

References

[1] A. Ahn and V. Ramaswami. Transient analysis of fluid flow models via stochastic coupling to a queue. *Stochastic Models*, **20**(1):71–104, 2004.

[2] D. Anick, D. Mitra and M. M. Sondhi. Stochastic theory of a data handling system with multiple sources. *Bell System Technical Journal*, **61**:1871–1894, 1982.

[3] S. Asmussen. Stationary distributions for fluid flow models with or without Brownian noise. *Stochastic Models*, **11**:21–49,1995.

[4] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation AX+XB=C. *Communications of the ACM*, **15**(9):820–826, 1972.

[5] S. Baase. *Computer algorithms: introduction to design and analysis*, Addison-Wesley Publishing Company, 1978.

[6] N.G. Bean, M.M. O'Reilly and P.G. Taylor. Hitting probabilities and hitting times for stochastic fluid flows. Submitted for publication.

[7] R. Bhatia and P. Rosenthal. How and why to solve the operator equation $AX - XB = Y$. *Bulletin of the London Mathematical Society*, **29**:1–21, 1997.

[8] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM Journal on Matrix Analysis and Applications*, **17**(4):906–926, 1996.

[9] A. da Silva Soares and G. Latouche. Further results on the similarity between fluid queues and QBDs. In G. Latouche and P. Taylor, editors, *Matrix-Analytic Methods Theory and Applications* , World Scientific Press 2002, pages 89–106.

[10] J. D. Gardiner, A. J. Laub, J. J. Amato and C. B. Moler. Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Transactions on Mathematical Software*, **18**(2):223–231, 1992.

[11] G. H. Golub, S. Nash and C. Van Loan. A Hessenberg-Schur method for the problem $AX + XB = C$. *IEEE Transactions on Automatic Control*, **AC-24**(6):909–913, 1979.

[12] A. Graham. *Kronecker products and matrix calculus with applications.* Ellis Horwood Limited, 1981.

[13] C-H. Guo. Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices. *SIAM Journal on Matrix Analysis and Applications*, **23**(1):225–242, 2001.

[14] C-H Guo. A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation. *Linear Algebra and its Applications*, **357**:299–302, 2002.

[15] C-H Guo. Convergence analysis of the Latouche-Ramaswami algorithm for null recurrent Quasi-Birth-Death processes. *SIAM Journal on Matrix Analysis and Applications*, **23**(3):744–760, 2002.

[16] C-H Guo and A.J. Laub. On the iterative solution of a class of nonsymmetric algebraic Riccati equations. *SIAM Journal on Matrix Analysis and Applications*, **22**(2):376–391, 2000.

[17] L. I. Kronsjö. *Algorithms: their complexity and efficiency.* John Wiley & Sons, 1979.

[18] G. Latouche. Newton's iteration for non-linear equations in Markov chains. *IMA Journal of Numerical Analysis.* **14**:583–598, 1994.

[19] G. Latouche and V. Ramaswami. *Introduction to matrix analytic methods in stochastic modeling.* American Statistical Association and SIAM, Philadelphia 1999.

[20] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Academic Press, New York, 1970.

[21] V. Ramaswami. Matrix analytic methods for stochastic fluid flows. *Proceedings of the 16th International Teletraffic Congress*, Edinburgh, 7-11 June 1999, pages 1019–1030.

[22] L.C. Rogers. Fluid models in queueing theory and Wiener-Hopf factorization of Markov chains. *The Annals of Applied Probability*, **4**(2):390–413, 1994.

[23] E. Seneta. *Non-negative matrices and Markov chains.* Springer-Verlag, 1981.

[24] R. S. Varga. *Matrix iterative analysis.* Prentice-Hall, 1962.

[25] D. Williams. A "potential theoretic" note on the quadratic Wiener-Hopf equation for Q-matrices. *Seminaire de Probabilities XVI, Lecture Notes in Math.*, **920**:91–94, 1982.