An Iterative Solution of Two-Dimensional Birth and Death Processes

ALEXANDRE BRANDWAJN

Ecole Nationale Supérieure des Télécommunications, Paris, France

(Received October 1976; accepted October 1978)

This paper presents an iterative, seminumerical method for solving the balance equations of finite two-dimensional birth and death processes. The method is seminumerical in that it uses the formal knowledge of the stationary probability distribution of one variable, and the iteration is applied to the conditional probabilities of the second variable given the first one. Sufficient convergence conditions for this approach are discussed. An always convergent entirely numerical alternative solution is also presented. Empirical results indicate that both methods perform, in many cases, several times better (in terms of time required) than the commonly used Gauss-Seidel method. Possible generalizations to processes of more than two dimensions are also indicated.

IN THIS PAPER, we present a seminumerical iterative approach to the solution of the balance equations of a finite two-dimensional birth and death process (in the case of an infinite process, an approximate solution can be obtained by truncating at some upper values of the state variables). The method (believed to be novel) is based on using the notion of equivalence which yields a formal explicit solution for the stationary probability distribution of one of the state variables. An iterative technique is then used to obtain the probability distribution of the second state variable conditioned on the first one. In Section 2 we discuss sufficient conditions under which our method converges. These conditions do not guarantee that the method always converges. Therefore, in Section 3 an always convergent, entirely numerical approach is also presented. Both methods require roughly the same storage—that for storing the vector of probabilities of the state variables. With regard to the speed of convergence, the seminumerical method is very fast in many cases, while the always convergent method requires a significantly greater number of iterations with, however, lower computational complexity.

The last section is devoted to the results of an empirical study of the performances of our methods. In many cases these results indicate execution times several times better than those of the commonly used Gauss-Seidel iteration. An additional advantage of the methods proposed is that at no time can an element of the approximate solution computed

Operations Research
Vol. 27, No. 3, May-June 1979

0030-364X/79/2703-0595 $01.25

at an iteration become negative, as may be the case, for example, in the over-relaxation method.

Let us now define the birth and death process considered. We assume a process with two variables: n_1 and n_2, taking on the values $0, 1, \cdots, N_1$ and $0, 1, \cdots, N_2$, respectively. The birth rates are $\lambda_1(n_1, n_2)$ for n_1, where $\lambda_1(N_1, n_2) = 0, \forall n_2$, and $\lambda_2(n_1, n_2)$ for n_2, where $\lambda_2(n_1, N_2) = 0, \forall n_1$, and the death rates are $\mu_1(n_1, n_2)$, where $\mu_1(0, n_2) = 0, \forall n_2$, and $\mu_2(n_1, n_2)$, where $\mu_2(n_1, 0) = 0, \forall n_1$, respectively.

The balance equations for our process can be written as

$$\begin{aligned} &-[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2) + \mu_1(n_1, n_2) + \mu_2(n_1, n_2)]p(n_1, n_2) \\ &+ \lambda_2(n_1, n_2 - 1)p(n_1, n_2 - 1) + \mu_2(n_1, n_2 + 1)p(n_1, n_2 + 1) \\ &+ \lambda_1(n_1 - 1, n_2)p(n_1 - 1, n_2) + \mu_1(n_1, n_2)p(n_1 + 1, n_2) = 0, \\ &\quad n_1 = 0, \cdots, N_1; \qquad n_2 = 0, \cdots, N_2, \end{aligned} \tag{1}$$

where $p(n_1, n_2)$ denotes the stationary probability of n_1 and n_2, and it is assumed that $p(n_1, n_2) = 0$ if any of n_1 or n_2 is negative or greater than its corresponding maximum value. We assume that such a stationary probability distribution exists for our process (see [1] for instance), so that (1) has a *unique solution* which satisfies the normalization condition

$$\sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} p(n_1, n_2) = 1. \tag{2}$$

In the next section we present a method for obtaining $p(n_1, n_2)$, which, unlike some other numerical methods, does not attempt to solve (1) directly, but rather a system derived from (1).

1. THE METHOD

We start by noticing that it is easy to obtain, *formally*, the stationary probability distribution for one of the state variables, say n_1. This distribution will be noted $p(n_1)$. Indeed, let $p_{n_1}(n)$ be the stationary conditional probability of n_2 given $n_1, p_{n_1}(n_2) = \text{Prob}\{n_2 | n_1\}$. It is not difficult to show that, at the stationary state, our process has the same probability distribution $p(n_1)$ (i.e., that the process is equivalent with respect to $n_1 \cdots$) as a one-dimensional birth and death process with birth rate

$$l(n_1) = \sum_{n_2=0}^{N_2} p_{n_1}(n_2)\lambda_1(n_1, n_2), \qquad n_1 = 0, \cdots, N_1 - 1; \tag{3}$$

and death rate

$$m(n_1) = \sum_{n_2=0}^{N_2} p_{n_1}(n_2)\mu_1(n_1, n_2), \qquad n_1 = 1, \cdots, N_2. \tag{4}$$

We shall assume henceforth that these rates are non null (the influence of this assumption will be discussed later on). $p(n_1)$ is then given by

$$p(n_1) = 1/G \prod_{i=1}^{n_1} l(i-1)/m(i), \qquad n_1 = 0, \cdots, N_1; \tag{5}$$

where an empty product is assigned the value + 1, and G is a normalization constant.

Now, $p(n_1, n_2)$ can be expressed as

$$p(n_1, n_2) = p(n_1)p_{n_1}(n_2). \tag{6}$$

Using (6) together with the distribution $p(n_1)$ (5) in the balance equations of our process (1) we obtain the following equations for the conditional probabilities $p_{n_1}(n_2)$:

$$\begin{aligned} &-[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2) + \mu_1(n_1, n_2) + \mu_2(n_1, n_2)]p_{n_1}(n_2) \\ &+ \lambda_2(n_1, n_2 - 1)p_{n_1}(n_2 - 1) + \mu_2(n_1, n_2 + 1)p_{n_1}(n_2 + 1) \\ &+ m(n_1)\lambda_1(n_1 - 1, n_2)p_{n_1-1}(n_2)/l(n_1 - 1) \\ &+ l(n_1)\mu_1(n_1 + 1, n_2)p_{n_1+1}(n_2)/m(n_1 + 1) = 0, \\ &\quad n_1 = 0, \cdots, N_1; \qquad n_2 = 0, \cdots, N_2, \end{aligned} \tag{7}$$

where it is assumed that $p_{n_1}(n_2) = 0$ if any of n_1, n_2 is negative or greater than its corresponding maximum value.

Equation (7) involves only the birth and death rates of our process and the conditional probabilities $p_{n_1}(n_2)$. It results from the assumption on the uniqueness of the solution of (1) that (7) has a unique solution satisfying the probability condition $\sum_{n_2=0}^{N_2} p_{n_1}(n_2) = 1$, for all $n_1 = 0, \cdots, N_1$.

In order to obtain $p(n_1, n_2)$, the stationary state distribution for our process, it suffices, given (6) and (5) with (3) and (4), to compute $p_{n_1}(n_2)$. We thus propose the following iterative scheme for solving equation (7). (We use a superscript to indicate the iteration number.) Choose initial distributions $p^0_{n_1}(n_2)$ so that neither $m^o(n_1)$ nor $l^o(n_1)$ are zero ($m^i(n_1)$ and $l^i(n_1)$, $i = 0, 1, \cdots$, are given by (3) and (4) with $p^i_{n_1}(n_2)$ instead of $p_{n_1}(n_2)$).

Starting from $n_1 = 0$, solve at iteration i, $i = 1, 2, \cdots$,

$$\begin{aligned} &-[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2) + \mu_1(n_1, n_2) + \mu_2(n_1, n_2)]p^i_{n_1}(n_2) \\ &+ \lambda_2(n_1, n_2 - 1)p^i_{n_1}(n_2 - 1) + \mu_2(n_1, n_2 + 1)p^i_{n_1}(n_2 + 1) \\ &+ m^i(n_1)\lambda_1(n_1 - 1, n_2)p^i_{n_1-1}(n_2)/l^i(n_1 - 1) \\ &+ l^i(n_1)\mu_1(n_1 + 1, n_2)p^{i-1}_{n_1+1}(n_2)/m^{i-1}(n_1 + 1) = 0, \end{aligned} \tag{8}$$

$$n_2 = 0, \cdots, N_2 - 1; \qquad \sum_{n_2=0}^{N_2} p^i_{n_1}(n_2) = 1; \tag{9}$$

for consecutive values of n_1.

Note that, *for each* n_1, (8) constitutes a relatively simple recurrence relation for the corresponding $p^i_{n_1}(n_2)$. In practice, the latter can be

computed by letting $p^i_{n_1}(n_2) = a(n_2)p^i_{n_1}(0) + b(n_2)m^i(n_1) + c(n_2)l^i(n_1)$, with $a(0) = 1$, and $b(0) = c(0) = 0$. The remaining $a(n_2)$, $b(n_2)$ and $c(n_2)$ are determined recurrently from (8), and the three unknowns $p^i(0)$, $m^i(n_1)$, and $l^i(n_1)$ are computed using the normalization condition (9), and the definitions for $m^i(n_1)$ and $l^i(n_1)$ (i.e., (3) and (4)). Note also that (8), (9) guarantee that all the $p^i_{n_1}(n_2)$ will be non-negative if this is the case for the $p^{i-1}_{n_1}(n_2)$.

It is clear that if our scheme converges, i.e., if $\lim_{i\to\infty} p^i_{n_1}(n_2)$ exists for all n_1, n_2, we have $\lim_{i\to\infty} p^i_{n_1}(n_2) = p_{n_1}(n_2)$.

Before giving, in the next section, sufficient conditions for our scheme to converge, let us return briefly to the assumption that $m(n_1) \neq 0$ for $n_1 = 1, \cdots, N$ and $l(n_1) \neq 0$ for $n_1 = 0, \cdots, N_1 - 1$. This assumption merely ensures that $p(n_1)$ has the form given by (5). Relaxing this assumption would simply modify $p(n_1)$ so that equations (8) would have to be modified correspondingly. This would not invalidate, however, the basic idea of our method—that of computing the $p^i_{n_1}(n_2)$ from a simple recurrence relation independently for each n_1.

Finally, note that the storage requirement for our method is quite moderate. We essentially need arrays to store the birth and death rates, a single array for storing the $p^i_{n_1}(n_2)$ computed at each iteration, and also some storage for the recurrent computation of $p^i_{n_1}(n_2)$, reused for each n_1 (arrays $a(n_2)$, $b(n_2)$ and $c(n_2)$). The latter storage could be reduced by one third if we use $m^{i-1}(n_1)$ and $l^{i-1}(n_1)$ instead of $m^i(n_1)$ and $l^i(n_1)$ in (8). The $p^i_{n_1}(n_2)$, for each n_1, can then be expressed as $p^i_{n_1}(n_2) = a(n_2)p^i_{n_1}(0) + d(n_2)$, with $a(0) = 1$ and $d(0) = 0$. Once $a(n_2)$ and $d(n_2)$ computed for all values of n_2, $p^i_{n_1}(0)$ is determined using (9). In practice, for some cases, one may also find convenient to store the $m(n_1)$ and $l(n_1)$ rather than to recompute them using the $p_{n_1}(n_2)$.

2. SUFFICIENT CONVERGENCE CONDITIONS

It is well known from the theory of iterative numerical methods (see [2] for example), that a necessary and sufficient convergence condition for an iterative scheme of the form

$$X^i = f(X^{i-1}), \tag{10}$$

to solve the set of equations: $X = f(X)$ (X is a vector of unknowns), is that $f(X)$ satisfies a Lipschitz condition $\|f(X') - f(X'')\| \leq L\|X' - X''\|$ with $L < 1$ for every X', X'' in the domain where (10) is to converge.

An upper bound for L may be obtained by considering the norm of the Jacobi matrix for f, i.e., the matrix

$$\begin{matrix} \partial f_1/\partial X_1 & \cdots & \partial f_1/\partial X_k \\ \vdots & & \\ \partial f_k/\partial X_1 & \cdots & \partial f_k/\partial X_k \end{matrix}$$

where $X_1, \cdots, X_k$ are the elements of X, and $f_1, \cdots, f_k$ denote the corresponding equations in f. In our case the functions f are defined implicitly by (8), and, as a consequence, the Jacobi matrix, J, whose elements are the partial derivatives $\partial p^i_{n_1}(n_2)/\partial p^{i-1}_{m_1}(m_2)$, is the solution of the matrix equation

$$AJ = B. \tag{11}$$

Denote by $\delta_{x,y}$ the Kronecker delta. $a_{s,t}$, an element of the matrix A, where s corresponds to (n_1, n_2) (we shall write symbolically $s = (n_1, n_2)$), is given by

$$a_{s,t} = \begin{cases} [\lambda_1(n_1 - 1, N_2) - \lambda_1(n_1 - 1, k_2)]m^i(n_1)\lambda_1(n_1 - 1, n_2) \\ \quad \cdot p^i_{p_1-1}(n_2)/[l^i(n_1 - 1)]^2 \\ \quad + \delta_{k_2,n_2} m^i(n_1)\lambda_1(n_1 - 1, n_2)/l^i(n_1 - 1), \\ \text{for} \quad t = (n_1 - 1, k_2), \qquad k_2 = 0, 1, \cdots, N_2 - 1; \\ [\lambda_1(n_1, k_2) - \lambda_1(n_1, N_2)]\mu_1(n_1 + 1, n_2)p^{i-1}_{n_1+1}(n_2)/m^{i-1}(n_1 + 1) \\ \quad + [\mu_1(n_1, k_2) - \mu_1(n_1, N_2)]\lambda_1(n_1 - 1, n_2) \\ \quad \cdot p^i_{n_1-1}(n_2)/l^i(n_1 - 1) \\ \quad - \delta_{n_2,N_2-1}\mu_2(n_1, N_2) + h(k_2, n_2), \\ \text{for} \quad t = (n_1, k_2), \qquad k_2 = 0, 1, \cdots, N_2 - 1; \\ 0, \quad \text{otherwise;} \end{cases} \tag{12}$$

where $n_1 = 0, \cdots, N_1$; $n_2 = 0, \cdots, N_2 - 1$;

$$h(k_2, n_2) = \begin{cases} -[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2) + \mu_1(n_1, n_2) \\ \quad + \mu_2(n_1, n_2)], \qquad \text{if} \quad k_2 = n_2; \\ \lambda_2(n_1, n_2 - 1), \qquad \text{if} \quad k_2 = n_2 - 1; \\ \mu_2(n_1, n_2 + 1), \qquad \text{if} \quad k_2 = n_2 + 1 \quad \text{and} \quad n_2 \neq N_2 - 1; \\ 0, \quad \text{otherwise.} \end{cases}$$

$b_{s,r}$, an element of the matrix B, where $s = (n_1, n_2)$ and $r = (m_1, m_2)$, n_1, $m_1 = 0, \cdots N_1$; $n_2, m_2 = 0, \cdots, N_2 - 1$, is given by

$$b_{s,r} = \delta_{n_1+1,m_1} \frac{l^i(n_1)\mu_1(n_1 + 1, n_2)}{m^{i-1}(n_1 + 1)} \cdot \left[\frac{p^{i-1}_{n_1+1}(n_2)[\mu_1(m_1, m_2) - \mu_1(m_1, N_2)]}{m^{i-1}(n_1 + 1)} - \delta_{n_2,m_2} \right]. \tag{13}$$

A sufficient convergence condition is $\|J\| < 1$, i.e., $\|A^{-1}B\|^{<1}$, and, hence, a stronger condition

$$\|A^{-1}\| < 1/\|B\|. \tag{14}$$

Equation (14) is of more theoretical than practical interest, since it does not seem easy to determine which matrix norm (induced by a vector norm) will yield the closest bound for the Lipschitz constant, nor does it seem easy to obtain nontrivial bounds on $\|A^{-1}\|$ without inverting A. We note, however, that, in order for (14) to be satisfied, we must have

$$\|B\| < \|A\|, \tag{15}$$

and this is readily checked if we use, for example, the matrix norm $\| \|_1$, i.e., the largest sum of the absolute values of the elements of a column. A closer look at the elements of A and B indicates also that if $\lambda_1(n_1, n_2)$ and $\mu_1(n_1, n_2)$ are much smaller than $\lambda_2(n_1, n_2)$ and $\mu_2(n_1, n_2)$, the elements of the Jacobi matrix will be close to zero. This means that our scheme will converge rapidly if the transition rates which change the condition variable are small compared to other transition rates. Our method actually exploits this kind of ill-conditioning which is often a difficulty for other numerical methods. In practice, the method exhibits a reasonably fast convergence also when $\lambda_1(n_1, n_2)$ and $\mu_1(n_1, n_2)$ are of the same order of magnitude as $\lambda_2(n_1, n_2)$ and $\mu_2(n_1, n_2)$ (see Section 4).

The choice of the initial set of conditional probabilities $p^0_{n_1}(n_2)$ clearly affects the number of iterations needed to reach a given accuracy. There is a close relation between our iterative scheme and the equivalence and decomposition approximation for queueing networks [3, 4]. This is of interest for us here for it implies that the approximate solution obtained by decomposition is often a good starting point for the iterative procedure. In the case of the birth and death process considered in this paper, the decomposition yields

$$p^0_{n_1}(n_2) = g(n_1) \prod_{j=1}^{n_2} \lambda_2(n_1, j-1)/\mu_2(n_1, j),$$

$$n_2 = 0, \cdots, N_2; \qquad n_1 = 0, \cdots, N_1; \tag{16}$$

where $g(n_1)$ is a normalization constant. It is intuitively clear that the smaller the λ_1, μ_1 as compared to λ_2, μ_2, the closer (16) to the conditional probability distribution $p_{n_1}(n_2)$, although it can be shown that this is not the only factor (see [4] for example).

Finally, before closing this section, let us note that the applicability of our method is not restricted to the kind of two-dimensional birth and death processes considered in the introduction. The method applies equally well to processes involving transitions of the form $(n_1, n_2) \rightarrow (n_1 \pm 1, n_2 \pm 1)$. Denote by $\alpha_2(n_1, n_2)$ and $\alpha_1(n_1, n_2)$ the corresponding transition rates. Equations (3) and (4) have to be modified as follows:

$l(n_1) = \sum_{n_2=0}^{N_2} p_{n_1}(n_2)\lambda_1(n_1, n_2) + \alpha_2(n_1, n_2)$, and $m(n_1) = \sum_{n_2=0}^{N_2} p_{n_1}(n_2)\mu_1(n_1, n_2) + \alpha_1(n_1, n_2)$, and the term $-[\alpha_1(n_1, n_2) + \alpha_2(n_1, n_2)]p^i_{n_1}(n_2) + m^i(n_1)\alpha_2(n_1 - 1, n_2 + 1)p^i_{n_1-1}(n_2 + 1)/l^i(n_1 - 1) + l^i(n_1)\alpha_1(n_1 + 1, n_2 - 1)p^{i-1}_{n_1+1}(n_2 - 1)/m^{i-1}(n_1 + 1)$, added to (8). More generally, the method can be applied to processes of any finite number of dimensions. This point will be developed in a subsequent paper.

In the next section we present an alternative iterative scheme which is applied directly to (1), and whose convergence is always guaranteed.

3. AN ALWAYS CONVERGING ALTERNATIVE SCHEME

Let us consider the initial system of balance equations for our process (1), and let us note that one of the problems which arise when one wants to apply classical iterative schemes, such as the Gauss-Seidel iteration, or over-relaxation, is the normalization of the solution. Two main approaches seem to be used. In the first one, all the equations of (1) are used in the iteration process to compute a set of values, say $\tilde{p}^i(n_1, n_2)$, and the new approximation $p^i(n_1, n_2)$ is obtained as $\tilde{p}^i(n_1, n_2)/\sum \tilde{p}^i(n_1, n_2)$. This has a negative effect on the computation time and, especially when the size of the state space is important, on the accuracy, since some of the $\tilde{p}^i(n_1, n_2)$ may be very small with respect to the sum. In the second approach, one of the equations in (1) is replaced by the normalization condition (2). The latter may either be used directly as another linear equation of the system to be solved, or used to reduce the number of equations by one by setting one of the $p(n_1, n_2)$ to unity minus the sum of all the other probabilities. This does not have the negative effect on computation time of the first approach, but it does have a negative effect on accuracy, since errors tend to accumulate on the eliminated element.

Recently, Gaver and Humfeld proposed [5] to use a modified Gauss-Seidel iteration, and to normalize only when convergence is attained. This, however, has the drawback that the usual convergence test—the largest absolute value of the difference between consecutive iterates smaller than a given value—cannot be used since the normalization affects the tested value. (Incidentally, note that the use of conditional probabilities, by dividing the state space into many independently normalized "subspaces," considerably reduces accuracy problems due to normalization in the method of Section 1.)

We now propose an iterative scheme which uses only the equation (1), and which does not require any of the normalization approaches discussed above. The scheme is as follows. Choose an order of considering the states such that it corresponds to a steady increase or decrease of the variables. As an example, we shall choose the order of increasing n_1, n_2, i.e., first all the $p(0, n_2)$, $n_2 = 0, \cdots, N_2$, then all $p(1, n_2)$, etc. Set any

initial probability distribution $p^0(n_1, n_2)$. At iteration i, $i = 1, 2, \cdots$, compute

$$\begin{aligned} p^i(n_1, n_2) &= 1/\{1 + \rho[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2)]\} \\ &\quad \cdot \{p^{i-1}(n_1, n_2)[1 - \rho[\mu_1(n_1, n_2) + \mu_2(n_1, n_2)]] \\ &\quad + \rho[\lambda_1(n_1 - 1, n_2)p^i(n_1 - 1, n_2) \\ &\quad + \lambda_2(n_1, n_2 - 1)p^i(n_1, n_2 - 1) \\ &\quad + \mu_1(n_1 + 1, n_2)p^{i-1}(n_1 + 1, n_2) \\ &\quad + \mu_2(n_1, n_2 + 1)p^{i-1}(n_1, n_2 + 1)]\}, \\ &\quad n_1 = 0, \cdots, N_2; \qquad n_2 = 0, \cdots, N_2, \end{aligned} \tag{17}$$

where ρ is a positive real number.

We start by noting the low computational complexity of the iteration, and the low storage requirements, lower than for the method of Section 1, since no extra storage is needed for the computation.

Let us now restrict our attention to values of ρ in the interval $(0, 1/\max_{n_1,n_2}[\mu_1(n_1,n_2) + \mu_2(n_1, n_2)]$. With these limits for ρ, it is clear that $p^i(n_1, n_2)$ may never become negative. Moreover, the $p^i(n_1, n_2)$ are automatically normalized with respect to unity. Indeed, (17) may be rewritten as

$$\begin{aligned} p^i(n_1, n_2) &= p^{i-1}(n_1, n_2) + \rho\{-[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2)]p^i(n_1, n_2) \\ &\quad [\mu_1(n_1, n_2) + \mu_2(n_1, n_2)]p^{i-1}(n_1, n_2) + \lambda_1(n_1 - 1, n_2)p^i(n_1 - 1, n_2) \\ &\quad + \lambda_2(n_1, n_2 - 1)p^i(n_1, n_2 - 1) + \mu_1(n_1 + 1, n_2)p^{i-1}(n_1 + 1, n_2) \\ &\quad + \mu_2(n_1, n_2 + 1)p^{i-1}(n_1, n_2 + 1)\}, \\ &\quad n_1 = 0, \cdots, N_1; \qquad n_2 = 0, \cdots, N_2. \end{aligned} \tag{18}$$

Summing (18) over all the states we easily obtain $\sum_{n_1,n_2} p^i(n_1, n_2) = \sum_{n_1,n_2} p^{i-1}(n_1, n_2)$, because of the very nature of the balance equations of a Markovian system. From (18) it is also clear that, if this scheme converges, we have $\lim_{i\to\infty} p^i(n_1, n_2) = p(n_1, n_2)$.

It is not difficult to show that our scheme always converges, for any initial probability distribution. Let $d^i(n_1, n_2) = p^{i+1}(n_1, n_2) - p^i(n_1, n_2)$. Using (18) we obtain

$$\begin{aligned} &\{1 + \rho[\lambda_1(n_1, n_2) + \lambda_2(n_1, n_2)]\}\,|d^i(n_1, n_2)| \\ &\leq \{1 - \rho[\mu_1(n_1, n_2) + \mu_2(n_1, n_2)]\}\,|d^{i-1}(n_1, n_2)| \\ &\quad + \rho\{\lambda_1(n_1 - 1, n_2)\,|d^i(n_1 - 1, n_2)| \\ &\quad + \lambda_2(n_1, n_2 - 1)\,|d^i(n_1, n_2 - 1)| \\ &\quad + \mu_1(n_1 + 1, n_2)\,|d^{i-1}(n_1 + 1, n_2)| \end{aligned} \tag{19}$$

$$+ \mu_2(n_1, n_2 + 1)|d^{i-1}(n_1, n_2 + 1)|\},$$

$$n_1 = 0, \cdots, N_1; \qquad n_2 = 0, \cdots, N_2; \qquad i = 1, 2, \cdots.$$

Due to the fact that both $p^i(n_1, n_2)$ and $p^{i+1}(n_1, n_2)$ are normalized, we have $\sum_{n_1,n_2} d^i(n_1, n_2) = 0$, $i = 0, 1, \cdots$. This means that for at least one couple (n_1, n_2) the inequality in (19) must be strict. Hence, summing over all the states we obtain $\sum_{n_1,n_2} |d^i(n_1, n_2)| < \sum_{n_1,n_2} |d^{i-1}(n_1, n_2)|$, and the unconditional convergence of our scheme follows readily.

It is clear from (18) that the choice of ρ is of importance for the speed of convergence. The latter will be low if ρ is close to zero, and will thus increase as ρ increases, at least up to certain limit. We did not succeed in determining theoretically an optimum value (in terms of asymptotic convergence speed) for ρ. In practice, however, $\rho = 1/\max_{n_1,n_2}(\mu_1(n_1, n_2) + \mu_2(n_1, n_2))$ is apparently a good choice.

Finally, note that there is no difficulty in extending the applicability of this iterative scheme to the solution of the balance equations of any Markovian system. Orderings of system states other than the one used throughout this section may in some cases be preferable with respect to the speed of convergence. The iterative procedure has then to be arranged according to the ordering chosen.

In the next section we present the results of an empirical study of the performances of the methods proposed in this paper.

4. EMPIRICAL RESULTS ON PERFORMANCE

We have tested the methods of Section 1 and 3 (which will be called henceforth Method 1 and 2, respectively) for a number of values of the birth and death rates of the process considered. At the same time, an over-relaxation iteration with the parameter set to (1) (i.e., an iteration very close to the Gauss-Seidel method) was run, so as to compare the performances of our method with that of a commonly used procedure. The normalization condition was explicitly used as an equation of the system to be solved. All the methods were programmed in Algol and run on a CII IRIS 80 computer using double precision (64 bits) floating arithmetic. The same convergence test, corresponding to a difference between two consecutive iterates of $p_{n_1}(n_2)$ less than 10^{-4}, was used throughout the experiments. In all cases, the initial distribution corresponded to $p^i_{n_1}(n_2) = 1/(N_2 + 1)$, $n_1 = 0, \cdots, N_1$.

Below are some results which seem to be significant of the behavior of the methods considered. (M_1: Method 1; M_2: Method 2; M_0: over-relaxation).

I. $N_1 = 10$, $N_2 = 10$, birth and death rates constant,
$\lambda_1 = 1$, $\mu_1 = 2$

		M_1	M_2	M_0
1. $\lambda_2 = 100$ $\mu_2 = 200$	No. of iterations	4	41	∞*
	Time in *mn*	0.01	0.03	

* No convergence after 1000 iterations.

		M_1	M_2	M_0
2. $\lambda_2 = 10$ $\mu_2 = 20$	No. of iterations	8	45	168
	Time in *mn*	0.02	0.03	0.11
		M_1	M_2	M_0
3. $\lambda_2 = 1$ $\mu_2 = 2$	No. of iterations	31	82	355
	Time	0.05	0.06	0.22

II. $N_1 = 10$, $N_2 = 15$, birth and death rates constant,

		M_1	M_2	M_0
1. $\lambda_1 = 1$, $\mu_1 = 2$, $\lambda_2 = 10$, $\mu_2 = 20$	No. of iterations	9	62	169
	Time	0.03	0.06	0.16

III. $N_1 = 25$, $N_2 = 10$, birth and death rates constant, $\lambda_1 = 1$, $\mu_1 = 2$

		M_1	M_2	M_0
1. $\lambda_2 = 10$ $\mu_2 = 20$.	No. of iterations	8	45	400
	Time	0.04	0.08	0.60
		M_1	M_2	M_0
2. $\lambda_2 = 1$ $\mu_2 = 2$	No. of iterations	31	82	461
	Time	0.11	0.13	0.72

IV. $N_1 = 10$, $N_2 = 10$, birth rates constant, $\mu_1(n_1, n_2) = 2$, if $n_2 = 0$, 1, if $n_2 \neq 0$.

		M_1	M_2	M_0
1. $\mu_2(n_1, n_2) = 20$, if $n_1 = 0$, 10, if $n_1 \neq 0$.	No. of iterations	8	62	205
	Time	0.02	0.04	0.13
		M_1	M_2	M_0
2. $\mu_2(n_1, n_2) = 2$, if $n_1 = 0$, 1, if $n_1 \neq 0$.	No. of iterations	37	127	66
	Time	0.05	0.09	0.05

We observe that, in most cases, the methods proposed are considerably superior to the commonly used iterative scheme. Method 1, despite its higher computational complexity, tends to outperform Method 2, as long as the transition rates which change the condition variable (n_1, in our case) are not greater than other transition rates. The influence of the size of the problem is illustrated in points II and III. We note that both methods behave reasonably as the dimensions of the problem increase; Method 1 seems less sensitive to the increase of N_1 than Method 2 with the ordering of states chosen. As a whole the results favor the use of Method 1. It should be noted, however, that the difference in time between Methods 1 and 2 is not, in many cases, very important, and Method 2 does have some advantages over Method 1: it uses almost directly the familiar balance equations of the process, the computation at each iteration is much simpler and requires less storage, and, finally, but importantly, its convergence is guaranteed.

REFERENCES

1. T. L. SAATY, *Elements of Queueing Theory*, McGraw-Hill, New York, 1961.
2. P. HEINRICI, *Elements of Numerical Analysis*, John Wiley & Sons, New York, 1964.
3. A. BRANDWAJN, "A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods," *Acta Informatica* **4,** 11–47 (1974).
4. A. BRANDWAJN, "A Note on the Accuracy of the Equivalence and Decomposition Approach, "Technical Report ENST-D-76007, Ecole Nationale Supérieure des Télécommunications, Paris, May 1976.
5. D. P. GAVER, G. HUMFELD, "Multitype Multiprogramming Models," *Acta Informatica* **7,** 111–121 (1976).