Iterative Methods for Overflow Queueing Models II

R. H. Chan*
Department of Mathematics
University of Hong Kong

Summary: Preconditioned conjugate gradient methods are employed to find the steady-state probability distribution of Markovian queueing networks that have overflow capacity. Different singular preconditioners that can be handled by separation of variables are discussed. The resulting preconditioned systems are nonsingular. Numerical results show that the number of iterations required for convergence grows very slowly with the queue sizes.

Subject Classifications: AMS : 65N20, 65F10, 60K25

1 Introduction

In a Markovian queueing network, most of the quantities of interest, for example the blocking probability and the waiting time for customers in various queues, can be expressed in terms of the steady-state probability distribution. The distribution is the solution of the Kolmogorov balance equations. The resulting matrix system has dimension N, where N is the total number of states in the network. The matrix, called the generating matrix, is non-symmetric and is known to have a one dimensional null-space. The steady-state probability distribution is the normalized right null-vector of this matrix.

The method we employ to find the null-vector is based on the preconditioned conjugate gradient method. The preconditioner is a singular matrix of order N which can be handled by separation of variables. Although the original matrix is singular, we can reduce the problem to solving a non-singular system by computing the components of the eigenvector which is orthogonal to the null-space of this chosen separable problem. The nonzero components of the eigenvector correspond to the states where overflow between the queues take

*This research was supported in part by the National Science Foundation grant DCR-8405506 and DCR-8602563

place. The effective dimension of the problem can usually be reduced by an order n, where n is the individual queue size.

In Chan [6], we applied our method to overflow networks with two queues. The preconditioner was chosen to be the generating matrix of the corresponding 2-queue free model. We observed numerically that the number of iterations required to attain a given accuracy is almost constant independent of the queue size n. We then proved analytically that when there is only one server in each queue, the number of iterations required can grow no faster than $O(\log^2 n)$. As a corollary, we also established the fast convergence of the preconditioned conjugate gradient method when applied to the oblique boundary value problems with the correponding Neumann problem as preconditioner.

In this paper, we generalize our method in three different directions. In § 2, we apply our method to overflow queueing networks with rectangular state-spaces but arbitrary number of queues. As preconditioner, we use the generating matrix of the corresponding free model. In § 3, we consider other possible preconditioners for these networks by first perturbing the singular system into a nonsingular one. We introduce a family of preconditioners which includes the generating matrix of the free model as a limiting case. These preconditioners are separable such that the corresponding systems can be solved economically. In § 4, we consider networks where the state-spaces are not rectangular. We will employ the technique of substructuring to find our optimal preconditioners. The numerical results are reported in § 5. A comparison is made between this method and the point SOR method. We see that our method has a much better performance. In fact, the number of iterations required to attain a given accuracy grows very slowly with the queue sizes.

2 q-Queue Overflow Models

Let us begin by introducing the notations that we will be using. Given a network with q queues, we assume that customers are coming from q independent Poisson sources. In the i-th queue, $1 \leq i \leq q$, there are s_i parallel servers, and $n_i - s_i - 1$ waiting spaces. Customers enter the queue with mean arrival rate $\lambda_i > 0$. The service time distributions of the servers are independent and each of them is exponential distributed with mean $\mu_i^{-1} > 0$. Let $p_{i_1,i_2,\ldots,i_q}$ denote the steady-state probability distribution which gives the probability of state $(i_1, \ldots, i_q)$, i.e., the probability that i_j customers are in the j-th queue, $1 \leq j \leq q$. Since $0 \leq i_j < n_j$, $1 \leq j \leq q$, the total number of states in the system is $N = \prod_{j=1}^{q} n_j$. For simplicity, let I_k be the identity matrix of order k and δ_{ij} be the Kronecker delta.

The idea of preconditioning the singular generating matrix of an overflow queueing model by the singular generating matrix of the corresponding free-model, as discussed in Chan [6], can easily be extended to more general overflow queueing networks, provided that both matrices have the same dimension. Con-

sider the following q-queue overflow networks. Customers from the i-th Poisson source, $1 \le i \le q$, will enter, wait and be served at the i-th queue if it is not yet full. In particular, customers cannot jump between the waiting lines and the overflow of customers from any queue can occur only when the queue is full. These conditions will ensure that all the N states in the network are accessible. Hence the generating matrix A will be of the same order as the preconditioner A_0, the generating matrix of the q-queue free model.

When a particular queue is full, customers entering the queue will overflow and be served at other queues according to some given queueing disciplines. For simplicity, let us denote by

$$i_1 \mapsto i_2 \mapsto \cdots \mapsto i_j \tag{2.1}$$

the queueing discipline that customers from the i_1-th Poisson source can overflow and be served at the i_j-th queue if the i_1-th, i_2-th, $\cdots$, i_{j-1}-th queues are all full and the i_j-th queue is not yet full.

To avoid ambiguity, we assume that for any given queue there is at most one direction of overflow of customers, i.e.

$$\cdots \mapsto i \mapsto j \mapsto \cdots \text{ and } \cdots \mapsto i \mapsto k \mapsto \cdots \implies j = k. \tag{2.2}$$

Moreover, to prevent customers from wandering within the network, we assume that any given queueing discipline does not form a loop. More precisely, in (2.1),

$$k \neq l \implies i_k \neq i_l, \ \ 1 \le k, l \le j. \tag{2.3}$$

Let p be the steady-state probability distribution vector of this network. Then p is the solution of the following problem:

$$\begin{cases} Ap = 0, \\ \mathbf{1}^* p = 1, \\ p_{i_1, i_2, \cdots, i_q} \ge 0. \end{cases} \tag{2.4}$$

Here A is the generating matrix of the network and $\mathbf{1}$ is the N-vector of all ones. Similar to the 2-queue model case discussed in Chan [6], we partition the generating matrix A as

$$A = A_0 + R_0. \tag{2.5}$$

Here A_0 is the generating matrix of the q-queue free model and is given by

$$A_0 = \sum_{j=1}^{q} \bigotimes_{i=1}^{q} G_i^{\delta_{ij}}, \tag{2.6}$$

where

$$G_i = \begin{bmatrix} \lambda_i & -\mu_i & & & & & & \\ -\lambda_i & \lambda_i+\mu_i & -2\mu_i & & & & 0 & \\ & -\lambda_i & \lambda_i+2\mu_i & -3\mu_i & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & -\lambda_i & \lambda_i+s_i\mu_i & -s_i\mu_i & & \\ & & & & \cdot & \cdot & \cdot & \\ & 0 & & & & -\lambda_i & \lambda_i+s_i\mu_i & -s_i\mu_i \\ & & & & & & -\lambda_i & s_i\mu_i \end{bmatrix} \tag{2.7}$$

$1 \le i \le q$, is a matrix of order n_i and is the generating matrix of a 1-queue free model. We remark that the steady-state probability distribution vector p_0 corresponding to A_0 is given by

$$p_0 = \bigotimes_{i=1}^{q} S_i^2 \mathbf{1}_i, \tag{2.8}$$

where $\mathbf{1}_i$ is the n_i-vector of all ones and S_i is the diagonal similarity transformation that symmetrizes G_i with normalization condition $\mathbf{1}_i^* S_i^2 \mathbf{1}_i = 1$. Indeed, for all $i = 1, \ldots q$, $S_i^2 \mathbf{1}_i$ is the steady-state probability distribution vector corresponding to G_i, i.e. $G_i S_i^2 \mathbf{1}_i = 0$.

The residual matrix R_0 consists of terms that correspond to the overflow disciplines (2.1). To obtain R_0, let us first define

$${}^jR_i \equiv \lambda_i \cdot \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & 0 & \\ & \cdot & \cdot & & & \\ & & \cdot & \cdot & & \\ & 0 & & -1 & 1 & \\ & & & & -1 & 0 \end{bmatrix}, \tag{2.9}$$

a square matrix of order n_j, ${}^je_i$ to be the i-th unit vector in R^{n_j} and $E_i \equiv {}^ie_{n_i}\,{}^ie^*_{n_i}$. Then any queueing discipline of the form $i \mapsto j$ implies that R_0 contains the following term

$$R_{ij} = \bigotimes_{k=1}^{q} E_k^{\delta_{ik}} \left({}^kR_i\right)^{\delta_{jk}}. \tag{2.10}$$

This is because overflow from the i-th queue to the j-th queue amounts to adding

$$\lambda_i \cdot \delta_{k_i n_i - 1}(1 - \delta_{k_j n_j - 1}) p_{k_1,\ldots,k_q} \tag{2.11}$$

to one side of the Kolmogorov balance equations and

$$\lambda_i \cdot \delta_{k_i n_i - 1}(1 - \delta_{0 k_j}) p_{k_1,\ldots,k_j-1,\ldots,k_q} \tag{2.12}$$

to the other side, for $1 \leq k_j < n_j$, $1 \leq j \leq q$. The term in (2.11) indicates that customers are leaving the state $(k_1, \ldots, k_q)$ at an additional rate λ_i when the i-th queue is full and the j-th queue is not yet full. The term in (2.12) indicates that customers are entering the state $(k_1, \ldots, k_q)$ at an additional rate λ_i from the state $(k_1, \ldots, k_j - 1, \ldots, k_q)$ when the i-th queue is full.

Queueing disciplines involving more queues mean that the terms added to R_0 have more E_i factors replaced by the I_{n_i} factors. Thus they are more sparse. For example, the queueing discipline $i \mapsto j \mapsto k$ means that we have to add to R_0 the term

$$R_{ijk} = \bigotimes_{l=1}^{q} E_l^{\delta_{il}} E_l^{\delta_{jl}} ({}^l R_i)^{\delta_{kl}}. \tag{2.13}$$

This follows from the fact that we have to add

$$\lambda_i \cdot \delta_{l_i n_i - 1} \delta_{l_j n_j - 1} (1 - \delta_{l_k n_k - 1}) p_{l_1, \ldots, l_q}$$

and

$$\lambda_i \cdot \delta_{l_i n_i - 1} \delta_{l_j n_j - 1} (1 - \delta_{0 l_k}) p_{l_1, \ldots, l_k - 1, \ldots, l_q}$$

for $1 \leq l_j < n_j$, $1 \leq j \leq q$, to the two sides of the Kolmogorov balance equations. In general, the queueing discipline in (2.1) implies that R_0 has the term

$$R_{i_1, \cdots, i_j} = \bigotimes_{k=1}^{q} E_k^{\delta_{i_1 k}} \cdots E_k^{\delta_{i_{j-1} k}} ({}^k R_i)^{\delta_{i_j k}}. \tag{2.14}$$

We note that by assumption (2.3), every term added to R_0 has one and only one ${}^j R_i$ factor in it. Here the i and j indicate the original source and the final destination of the overflowed customers respectively. Thus all the terms added to R_0 have zero column sums, non-negative diagonal and non-positive off-diagonal entries. This implies that A is an irreducible matrix with zero column sums, strictly positive diagonal and non-positive off-diagonal entries, as A_0 is already a matrix having these properties. Hence by Frobenius theory on positive matrix, (see, for instance, Varga [16]), the steady-state probability distribution vector p of the network exists and is unique.

To obtain p, we first recall that (see Chan [6])

$$R^N = \text{span} < p_0 > \oplus Im(A_0), \tag{2.15}$$

and that the generalized inverse A_0^+ of A_0 is invertible on

$$Im(A_0) = \{x \in R^N | \mathbf{1}^* x = 0\}. \tag{2.16}$$

In view of this, we can write p as

$$p = p_0 + A_0^+ \xi_0, \tag{2.17}$$

where $\xi_0 \in Im(A_0)$. Substituting this Ansatz into the equation $Ap = 0$, we get,

$$(I + R_0 A_0^+)\xi_0 = -R_0 p_0. \tag{2.18}$$

We claim that the matrix $AA_0^+ = I + R_0 A_0^+$ is non-singular, i.e. the singularity of A is cancelled exactly by the singularity of A_0. In fact, we have

Lemma 1 Consider a system of the form

$$\begin{cases} (A_0 + R_0)p &= 0, \\ \mathbf{1}^* p &= 1, \\ p_j &\geq 0. \end{cases}$$

If the solution p exists and is unique and $\mathbf{1}^* R_0 = \mathbf{0}$ then the matrix $(I + R_0 A_0^+)$ is non-singular.

Proof: We first note that $\mathbf{1}^* R_0 = \mathbf{0}$ implies that $Im(R_0) \subseteq Im(A_0)$. Thus $(I + R_0 A_0^+)$ maps $Im(A_0)$ into itself. Moreover, the existence and uniqueness of p implies the existence and uniqueness of a $\xi_0 \in Im(A_0)$ that satisfies $(I + R_0 A_0^+)\xi_0 = -R_0 p_0$. Thus the matrix is invertible in $Im(A_0)$. Suppose y is in the kernel of this matrix. By (2.15), there exists a unique β and $x \in Im(A_0)$ such that $y = \beta p_0 + x$. Hence $(I + R_0 A_0^+)y = 0$ implies that $-\beta p_0 = (I + R_0 A_0^+)x + \beta R_0 A_0^+ p_0$. Since $Im(R_0) \subseteq Im(A_0)$, the right hand side is in $Im(A_0)$. Thus by (2.15) again, $\beta = 0$ and $(I + R_0 A_0^+)x = 0$. Since $x \in Im(A_0)$, the last equation implies $x = 0$. Hence $y = 0$. Thus the matrix is non-singular. □

Next we claim that those terms added to R_0 are sparse. In fact, since overflow occurs only when at least one of the queues, say the i-th queue, is full, the corresponding term in R_0 has at most N/n_i non-zero rows. Moreover, since every such term represents the overflow from one queue to another, every non-zero row has at most two non-zero entries. If we permit all possible directions of overflow within the network, the total number of non-zero rows in R_0 will be bounded above by

$$m_q \equiv N \cdot \sum_{i=1}^{q} \frac{1}{n_i}. \tag{2.19}$$

Moreover, the total number of non-zero entries in every such row will not exceed $q + 1$, since there are only q queues in the network. Thus R_0 is also sparse.

By (2.18), we see that $\xi_0 \in Im(R_0)$. Thus, using the sparsity of R_0, we can reduce the system in (2.18) to a system which has order at most $m_q \approx qn^{q-1}$. However, for general networks, this is often too large to be handled by direct methods. Since they would require $O(m_q^2) = O(q^2 n^{2q-2})$ storage spaces. If we use the conjugate gradient method together with the source-and-target technique of Banegas [2], then in each iteration, only $O(qn^{q-1})$ storage is required. (For $q = 2$, the storage requirement is $O(n^2)$, for we have to store the n_i by n_i matrices that diagonalize G_i, see Chan [6].) The cost per iteration

will depend on the cost of computing $(I + R_0 A_0^+)\xi$ for $\xi \in Im(R_0)$. Notice that for any given vector x, the vector-matrix multiplication $R_0 x$ requires at most $(q+1)m_q$ operations. This work is negligible when compared to the work of computing $A_0^+ \xi$, which is on the order of $O(qn^{q+1})$ provided that we first obtain the spectral decomposition of A_0, i.e, diagonalize G_i, $1 \leq i \leq q$, see Chan [6]. Once ξ_0 is found, we can obtain p by using (2.17) and (2.8). We note that if only a linear functional $l^* p$ of p is required, where l is any N-vector, then the overall storage requirement will remain at $O(qn^{q-1})$ as we do not have to store the entries of p explicitly. This is of considerable advantage over the SOR method which would require $O(n^q)$ storage.

In § 5, we apply this method to the 3-queue model discussed in Kaufman [12]. It has queueing disciplines $1 \mapsto 2$, $2 \mapsto 3$ and $1 \mapsto 2 \mapsto 3$. Our method requires $O(n^2)$ storage, and using the sparity of the problem, the operation count is reduced to $O(n^3)$ per iteration. This algorithm is given in the Appendix of Chan [5]. We remark that our method is quite suitable for designing an algorithm that can accept the number of queues and the queueing disciplines as input parameters.

For single-server cases, $s_i = 1$, using the continuous analogy mentioned in Chan [6], the term $R_{i_1,\cdots,i_q}$ that make up R_0 corresponds to a forward difference operator on a particular face in the q-dimensional cube (see (2.9)). These faces correspond to states where overflow occurs. Thus R_0 is similar to an operator which is zero in the q-dimensional cube, but with tangential derivatives on some of the faces. A_0 is similar to a second order elliptic operator with constant coefficients and transport terms actin on this q-dimensional cube with Neumann boundary conditions everywhere, (see (2.7) with $s_i = 1$ and $\lambda_i \approx \mu_i$.) This operator is of the form

$$\sum_{i=1}^{q} (\lambda_i + \mu_i)\partial_{ii} + 2\sum_{i=1}^{q} (n_i - 1)(\lambda_i - \mu_i)\partial_i.$$

The matrix $A = A_0 + R_0$ therefore resembles this same operator but with oblique derivatives on some particular faces. Hence the boundary conditions of A and A_0 are of the same type. In our previous papers, see Chan [5] and [6], we have shown that a Neumann problem can be used sucessfully to precondition an oblique derivative problem in the constant coefficient case, we therefore expect our method to have fast convergence for these kind of models provided that the elliptic terms are not dominated by the transport terms, i.e. $(\lambda_i - \mu_i) = O(n^{-\alpha})$ with $\alpha > 1$.

In § 5, we see that under this assumption, our method converges much faster than the point-SOR method. The number of iterations required to attain a given accuracy increases like $O(\log n)$ when the queue size n increases. We also note that when $0 \leq \alpha < 1$, it is easy to show that the solution p tends to p_0 exponentially fast as $n \to \infty$, see Chan [6] § 3.2.

3 Other Separable Preconditioners

The preconditioner A_0 discussed in § 2 is not the only viable preconditioner for these queueing models. In this section, we will develop other possible preconditioners for these models. We will only consider separable preconditioners here because such systems can be solved economically. For simplicity, we confine ourselves to the 2-queue model discussed in Chan [6] and Kaufman [12], namely, a model with overflow discipline $1 \mapsto 2$. The idea can easily be extended to more general networks.

Let us consider the model in which overflow is permitted only from the first queue into the second, i.e. $1 \mapsto 2$. The corresponding equation of this model is given by

$$Ap = (A_0 + R_0)p = 0, \tag{3.1}$$

where

$$R_0 = ({}^1e_{n_1}{}^1e^*_{n_1}) \otimes {}^2R_1. \tag{3.2}$$

Here A_0 is given by (2.6) with $q = 2$, and ${}^2R_1$ is defined in (2.9). It is easy to check that the generating matrix A has a one dimensional null-space with positive null-vector p, hence we can fix one component of p, and solve the resulting non-singular system. More precisely, since p is positive, and is unique up to a multiple constant, we can always set p_N, the last entry of p, to 1 and partition the system (3.1) as

$$Ap = \begin{bmatrix} B & d \\ c^* & \eta \end{bmatrix} \begin{bmatrix} \tilde{p} \\ p_N \end{bmatrix} = 0.$$

Using the facts that A is irreducible and has zero column sum, we see that B is irreducibly diagonally dominant and hence nonsingular. Thus we can proceed to solve the reduced system $B\tilde{p} = -d$ by direct or iterative methods, see Kaufman [12] and Funderlic and Mankin [10]. However, it is impossible to design a separable preconditioner for B because its dimension is $n_1 n_2 - 1$. To get around this, we can, instead of considering submatrices of A, consider a perturbed version of (3.1). More precisely, we fix p_N such that $\lambda_1 p_N = 1$, or equivalently, let $\lambda_1({}^1e_{n_1}{}^1e^*_{n_1} \otimes {}^2e_{n_2}{}^2e^*_{n_2})p = e_N$. We then obtain p from

$$\tilde{A}p = \{A + \lambda_1({}^1e_{n_1}{}^1e^*_{n_1} \otimes {}^2e_{n_2}{}^2e^*_{n_2})\}p = e_N. \tag{3.3}$$

Notice that $\tilde{A}$ is irreducibly diagonally dominant and therefore non-singular. Since $\tilde{A}$ is of order $n_1 n_2$, it is now possible to design separable preconditioners for $\tilde{A}$.

(I) A family of separable preconditioners for $\tilde{A}$

Let us partition $\tilde{A}$ as $\tilde{A} = \tilde{A}_1 + \tilde{R}_1$, where

$$\tilde{A}_1 = V_1 \otimes I_{n_2} + I_{n_1} \otimes G_2, \tag{3.4}$$

$$\tilde{R}_1 = \lambda_1 \cdot \{{}^1e_{n_1}{}^1e^*_{n_1} \otimes \text{tridiag}(-1, 0, 0)\}, \tag{3.5}$$

$$V_1 = G_1 + \lambda_1 \cdot {}^1e_{n_1}{}^1e^*_{n_1}. \tag{3.6}$$

Clearly $\tilde{A}_1$ is separable and V_1 is irreducibly diagonally dominant. Hence $\tilde{A}_1$ is non-singular. We can write $p = \tilde{A}_1^{-1}\xi_1$ and solve the preconditioned system

$$\tilde{A}\tilde{A}_1^{-1}\xi_1 = (I + \tilde{R}_1\tilde{A}_1^{-1})\xi_1 = e_N.$$

Because of the sparsity of $\tilde{R}_1$, we can reduce this to an n_2 by n_2 system. Unfortunately, numerical results show that the convergence rate for this preconditioned system is very slow.

Notice that $\tilde{A}_1$ resembles the finite difference approximation of a second order elliptic operator on the square with a Dirichlet boundary condition on one of the sides and Neumann boundary conditions on the remaining sides. In fact, if $\lambda_i = \mu_i = s_i = 1$, then $V_1 = \text{tridiag}(-1, 2, -1) - e_1^* e_1$. This is exactly the finite difference approximation of a simple second order ordinary differential operator with a Neumann type data at one end and a Dirichlet type data at the other. Thus the slow convergence is due to the fact that $\tilde{A}_1$ is not a good approximation to $\tilde{A}$, it changes the oblique derivative in $\tilde{A}$ into a Dirichlet boundary condition; see Chan [5].

Notice that $\tilde{A}_1$ is not the only non-singular separable preconditioner for $\tilde{A}$, in fact, there exists a family of non-singular separable preconditioners. Let us define, for any β,

$$V_\beta = G_1 + \beta\lambda_1 \cdot ({}^1e_{n_1}\,{}^1e_{n_1}^*), \tag{3.7}$$

$$\tilde{A}_\beta = V_\beta \otimes I_{n_2} + I_{n_1} \otimes G_2, \tag{3.8}$$

$$\tilde{R}_\beta = \lambda_1 \cdot \{{}^1e_{n_1}\,{}^1e_{n_1}^* \otimes \text{tridiag}(-1, 1-\beta, 0)\}. \tag{3.9}$$

We note that $\tilde{A} = \tilde{A}_\beta + \tilde{R}_\beta$. Clearly $\tilde{A}_\beta$ is separable. When $\beta > 0$, V_β is irreducibly diagonally dominant and hence $\tilde{A}_\beta$ is nonsingular. We can then define $p = \tilde{A}_\beta^{-1}\xi_\beta$, and solve for ξ_β. These preconditioners correspond to operators with a mixed type of boundary condition on the side in question. Our numerical results show that the performance improves when β gets closer to zero. This can also be explained by using the continuous analogy.

Let us consider the case when $\beta = 0$. We obtain $V_0 = G_1$ and $\tilde{A}_0 = A_0$. This is the preconditioner considered previously. It is singular and corresponds to the operator with Neumann boundary conditions on every sides. Hence we cannot set $p = \tilde{A}_0^{-1}\xi_0$ and solve for ξ_0. However, we can still design a singular separable preconditioner for the non-singular matrix $\tilde{A}$, see (II) below.

For $\beta < 0$, the preconditioners again correspond to operators with mixed type boundary conditions. Numerical results show that the convergence rate is slower when β becomes more negative. We remark that V_β in (3.7) can be symmetrized by a diagonal matrix, but it is no longer definite. More precisely, by the Cauchy interlace theorem (see Parlett [14]), V_β, and hence $\tilde{A}_\beta$, has one negative eigenvalue.

(II) Separable Preconditioner for $\tilde{A}$ when $\beta = 0$

By (2.15), there exists unique scalar α and $\xi_0 \in Im(A_0)$ such that $p = \alpha p_0 + A_0^+ \xi_0$. Since we have set $\lambda_1 p_N = 1$, α is no longer arbitrary. Thus we need an extra equation for α. Recalling $\tilde{A} = A_0 + \tilde{R}_0$, (3.3) becomes

$$\tilde{A}(\alpha p_0 + A_0^+ \xi_0) = (I + \tilde{R}_0 A_0^+)\xi_0 + \alpha \tilde{R}_0 p_0 = e_N.$$

Moreover, since $\xi_0 \in Im(A_0)$, we have $\mathbf{1}^* \xi_0 = 0$. Combining these equations, we have the following $(N+1)$ by $(N+1)$ system

$$Ff \equiv \begin{bmatrix} (I + \tilde{R}_0 A_0^+) & \tilde{R}_0 p_0 \\ \mathbf{1}^* & 0 \end{bmatrix} \begin{bmatrix} \xi_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} e_N \\ 0 \end{bmatrix}. \tag{3.10}$$

We claim that F is non-singular. To prove this, suppose that $(\xi, \beta)^*$ is in the kernel of F. This implies that $\tilde{A}(\beta p_0 + A_0^+ \xi) = 0$ and $\mathbf{1}^* \xi = 0$. Since $\tilde{A}$ is non-singular, the first condition implies that $\beta p_0 + A_0^+ \xi = 0$. The second condition implies that $\xi \in Im(A_0)$, which by the definition of A_0^+, implies that $A_0^+ \xi \in Im(A_0)$. Thus by (2.15), we have $\beta = 0$ and $A_0^+ \xi = 0$. By the invertability of A_0^+ on $Im(A_0)$, we have $\xi = 0$. Hence F is non-singular and it is legitimate to solve for f in (3.10). By the sparsity of $\tilde{R}_0$, we can also reduce (3.10) to an $(n_2 + 1)$ by $(n_2 + 1)$ system.

Notice that by (3.3), $\tilde{A}$ differs from A by a rank one matrix. Thus $\tilde{R}_0$ differs from the R_0 in (3.2) by a rank one matrix. Hence the N by N leading sub-matrix of FF^* differs from the preconditioned matrix $(I + R_0 A_0^+)(I + R_0 A_0^+)^*$ by at most a rank three matrix. Using the Cauchy interlace theorem, the singular values of $(I + R_0 A_0^+)$ will interlace the singular values of F, except possibly a few outlying ones. In particular, if the singular values of $(I + R_0 A_0^+)$ are clustered, so will be the singular values of F. (We remark that, in Chan [6], we have shown that the singular values of $(I + R_0 A_0^+)$ are indeed clustered around $(1 + \lambda_1/\lambda_2)^{\frac{1}{2}}$ in the single-server case.) The numerical results in § 5 show that the convergence rate for these two systems are very similar.

4 An Overflow Model with Restricted State-Space

In this section, we consider a model in which overflow occurs even before a queue is full. The resulting problem is still a homogeneous system of the form $Ap = 0$, but with some of the entries of p being set to zero. Thus the dimension N_A of A is less than the dimension N of the preconditioner A_0. We introduce two methods here to solve this queueing problem. In the first, we partition the state space into subspaces in which we can find separable preconditioners. In the second, we embed the state space into one where we can use A_0 as a preconditioner.

Let us consider the following 2-queue network. Customers entering the first queue will wait and be served at the second queue if all the spaces in the first queue is filled. Moreover, we assume that a customer waiting for service in

the first queue is moved to and served at the second queue if a server in the second queue becomes available. Thus, some of the states are not admissible here. More precisely, we have

$$p_{i,j} = 0 \quad s_1 < i < n_1, 0 \leq j < s_2. \tag{4.1}$$

We may associate the values of $p_{i,j}$ with the following L-shaped region:

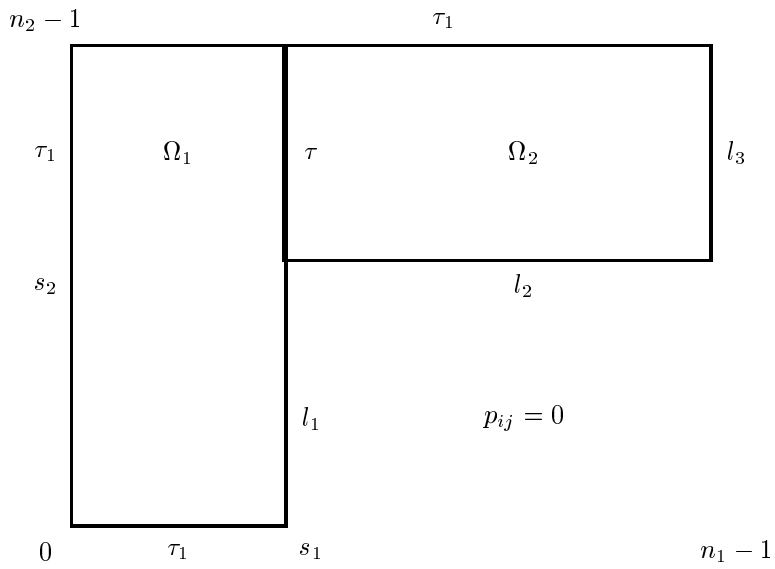

Figure 1

In the figure, l_1, l_2, l_3 are line segments defined by

$$l_1 \equiv \{s_1\} \times [0, s_2 - 1], \tag{4.2}$$

$$l_2 \equiv [s_1 + 1, n_1 - 1] \times \{s_2\}, \tag{4.3}$$

$$l_3 \equiv \{n_1 - 1\} \times [s_2, n_2 - 2]. \tag{4.4}$$

For simplicity, we let

$$\tau_2 \equiv l_1 \cup l_2 \cup l_3 \cup \{(s_1, s_2), (n_1 - 1, n_2 - 1)\}, \tag{4.5}$$

and τ_1 to be the set of all boundary states that are not in τ_2.

We remark that this model is similar to the one discussed in Kaufman, Serry and Morrison [11], except that we have added one more feature. Namely, we permit overflow from the first queue to the second queue when the first queue is full. The Kolmogorov balance equations of our model are given by

$$\begin{aligned}
&[\lambda_1(1 - \delta_{in_1-1}\delta_{jn_2-1}\chi_{j-s_2}) + \lambda_2(1 - \delta_{jn_2-1}) + \min(i, s_1)\mu_1 + \min(j, s_2)\mu_2]p_{i,j} \\
&= (1 - \chi_{i-1-s_1}\chi_{s_2-1-j})[\lambda_1(1 - \delta_{i0})p_{i-1,j} + (1 - \delta_{jn_2-1})\min(j + 1, s_2)\mu_2 p_{i,j+1}] \\
&+ (1 - \delta_{j0})[\lambda_1(\delta_{is_1}\chi_{s_2-j} + \delta_{in_1-1}\chi_{j-s_2}) + \lambda_2(1 - \chi_{i-1-s_1}\chi_{s_2-j})]p_{i,j-1} \\
&+ (1 - \delta_{in_1-1})[(1 - \chi_{i-s_1}\chi_{s_2-1-j})\min(i + 1, s_1)\mu_1 + s_2\mu_2\chi_{i-s_1}\delta_{js_2}]p_{i+1,j},
\end{aligned} \tag{4.6}$$

for $0 \le i < n_1$ and $0 \le j < n_2$. Here

$$\chi_l = \begin{cases} 1, & l \ge 0, \\ 0, & l < 0. \end{cases} \tag{4.7}$$

We note that these equations imply (4.1) and that the steady-state probability distribution p satisfies the homogeneous equation

$$Ap = 0. \tag{4.8}$$

Here A is the generating matrix of dimension N_A. Since p is a probability distribution, we supplement (4.8) with

$$\mathbf{1}^* p = 1, \tag{4.9}$$

$$p_{i,j} \ge 0. \tag{4.10}$$

It is straightforward to check that A is an irreducible matrix with zero column sums, strictly positive diagonal and non-positive off-diagonal entries. Hence the solution p to (4.8) - (4.10) exists and is unique. Moreover,

$$p_{i,j} > 0. \tag{4.11}$$

Considering the continuous analogy, we find that the matrix A resembles an second order elliptic operator acting on the L-shaped region with Neumann

boundary conditions on τ_1 and oblique boundary conditions on τ_2. The idea of the previous sections would suggest the partition

$$A = \bar{A} + \bar{R}, \tag{4.12}$$

with $\bar{A}$ resembling the same operator but with Neumann boundary conditions everywhere. $\bar{R}$ will then be an operator that is zero in the L-shape region, but has tangential derivatives along τ_2. We note that $\bar{A}$ has the form

$$\bar{A} = \begin{bmatrix} T_1 & D_1 & \mathbf{0} \\ E_1 & C_1 & D_2 \\ \mathbf{0} & E_2 & T_2 \end{bmatrix}. \tag{4.13}$$

Here T_i represents couplings between the pairs of states in Ω_i, C_1 couplings between the pairs of states on the interface τ, and D_i and E_i couplings between the pairs belonging to Ω_i and τ. The dimension of C_1 is equal to the number of states on τ, which is equal to $n_2 - s_2$. This is small when compared to the dimension of T_i, which is equal to the total number of states in Ω_i. The dimension of T_1 is $s_1 n_2$ and that of T_2 is $(n_2 - s_2)(n_1 - s_1 - 1)$. We note that in (4.13)

$$E_2 = [-\lambda_1 \cdot I_{n_2 - s_2}, \mathbf{0}]^*, \tag{4.14}$$

$$D_2 = [-s_1\mu_1 \cdot I_{n_2 - s_2}, \mathbf{0}], \tag{4.15}$$

where $\mathbf{0}$ is the zero matrix of order $(n_2 - s_2)$ by $(n_1 - s_1 - 2)$. Thus they are sparse.

We claim that the matrix $\bar{R}$ is sparse. We first note that, depending on the ordering of the states, any index j, $1 \le j \le N_A$, corresponds to a unique state (j_1, j_2) in the L-shaped domain. Using this notation, it is straightforward to check that

$$\bar{R} = \bar{R}_1 + \bar{R}_2. \tag{4.16}$$

Here $\bar{R}_1$ is a diagonal matrix given by

$$(\bar{R}_1)_{jj} = \begin{cases} \lambda_1 + s_2\mu_2 & (j_1, j_2) = (n_1 - 1, s_2), \\ \lambda_1 & (j_1, j_2) \in l_1 \cup l_3 \backslash \{(n_1 - 1, s_2)\}, \\ s_2\mu_2 & (j_1, j_2) \in l_2 \backslash \{(n_1 - 1, s_2)\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

and $\bar{R}_2$ is given by

$$(\bar{R}_2)_{kj} = \begin{cases} -\lambda_1 & (k_1, k_2 - 1) \text{ and } (j_1, j_2) \in l_1 \cup l_3, \\ -s_2\mu_2 & (k_1 + 1, k_2) \text{ and } (j_1, j_2) \in l_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

Thus the number of non-zero rows is equal to the number of states on τ_2, which is equal to

$$N_{\bar{R}} \equiv n_1 + n_2 - s_1 - 1. \tag{4.19}$$

Moreover, every such rows has at most two non-zero entries. It can be shown that $\bar{A}$ is still singular with a one dimensional null-space. However, $\bar{A}$ is not separable. Hence $\bar{A}^+x$ cannot be computed economically. In the following, we will design singular, separable preconditioners that are close to A in the sense that they represent the same operator in the L-shaped region and have the same type of boundary conditions. The first idea comes from the theory of substructuring.

4.1 Partitioning of the State-Space

The method of substructuring has been used for solving elliptic problems defined in irregular regions, see Bjϕrstad and Widlund [3], Dryja [8], Buzbee et al. [4] and Concus et al. [7]. The idea is to partition the problem into subproblems which correspond to subregions into which the original region has been partitioned. We can then solve each of these subproblems separately by direct methods while the interactions between the subregions are solved by a direct or iterative method. Since the number of nodes on the interface usually is small compared to the number of nodes in each subregions, the size of this interface problem is usually small compared to the original problem. If we are using iterative methods to solve the interface problem, then in each iteration, we will have to solve the subproblems once in each subregion. However, if the boundary conditions for the original region are such that separation of variables is possible, then solving the subproblems by direct methods will require very little work.

To be more specific, let us consider the problem of solving Laplace's equation in the L-shaped region depicted in Figure 1, with Neumann boundary conditions everywhere. Follows the idea from Bjϕrstad and Widlund [3], we construct the following preconditioner. We first solve the problem defined on Ω_1 with Neumann boundary conditions on the boundary of Ω_1 including τ. This is a separable problem. Having solved this problem, we use the value of the solution on the interface τ as Dirichlet data and solve a Dirichlet-Neumann problem on Ω_2 with Dirichlet boundary condition on the interface τ and Neumann boundary conditions on the remaining three sides. This problem is also separable.

Using the analogy between the queueing model and this continuous problem, we construct our preconditioner accordingly. The numerical results in § 5 show that this preconditioner is very good. In matrix terms, we partition our matrix A as

$$A = \tilde{A} + \tilde{R}, \tag{4.20}$$

where

$$\tilde{A} \equiv \begin{bmatrix} \tilde{A}_0 & \mathbf{0} \\ \tilde{E}_2 & T_2 \end{bmatrix} = \begin{bmatrix} T_1 & D_1 & \mathbf{0} \\ E_1 & C_2 & \mathbf{0} \\ \mathbf{0} & E_2 & T_2 \end{bmatrix}. \tag{4.21}$$

Here

$$\tilde{E}_2 \equiv [\mathbf{0}, E_2], \tag{4.22}$$

and

$$\tilde{A}_0 \equiv \begin{bmatrix} T_1 & D_1 \\ E_1 & C_2 \end{bmatrix} \tag{4.23}$$

with

$$C_2 = C_1 - \lambda_1 \cdot I_{n_2 - s_2}. \tag{4.24}$$

Comparing (4.21) with (4.13), we have

$$\tilde{R} = \bar{R} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_1 - C_2 & D_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.25}$$

Since $\bar{R}$, D_2 and $C_1 - C_2$ are sparse, $\tilde{R}$ is also sparse. In fact it has at most

$$m \equiv n_1 + 2n_2 - s_1 - s_2 - 2 \tag{4.26}$$

non-zero rows, and every such rows has at most two non-zero entries. Notice that the sub-matrix $\tilde{A}_0$ corresponds to a Neumann problem on the subregion Ω_1. In fact,

$$\tilde{A}_0 = \tilde{G}_1 \otimes I_{n_2} + I_{s_1+1} \otimes G_2, \tag{4.27}$$

where $\tilde{G}_1$, of dimension $s_1 + 1$, is the same as G_i in (2.7) but with s_1 replacing $n_i - 1$ there. Thus $\tilde{A}_0$ is the generating matrix of a 2-queue free model with s_1 spaces in the first queue and $n_2 - 1$ spaces in the second queue. Hence $\tilde{A}_0$ is separable, has a one dimensional null-space and a positive null-vector. Let us denote its null-vector by $\tilde{p}_0$.

On the other hand, T_2 corresponds to a mixed type problem defined on Ω_2. In fact,

$$T_2 = V_1 \otimes I_{n_2 - s_2} + I_{n_1 - s_1 - 1} \otimes V_2. \tag{4.28}$$

Here

$$V_1 \equiv \text{tridiag}(-\lambda_1, \lambda_1 + s_1\mu_1, -s_1\mu_1) - \lambda_1 \cdot e_{n_1 - s_1 - 1} e^*_{n_1 - s_1 - 1} \tag{4.29}$$

is a matrix of order $n_1 - s_1 - 1$ and

$$V_2 \equiv \text{tridiag}(-\lambda_2, \lambda_2 + s_2\mu_2, -s_2\mu_2) - s_2\mu_2 \cdot e_1 e_1^* - \lambda_2 \cdot e_{n_2 - s_2} e^*_{n_2 - s_2} \tag{4.30}$$

is a matrix of order $n_2 - s_2$. It is clear that T_2 is separable and since V_1 is irreducibly diagonally dominant, T_2 is non-singular. Thus by (4.21), $\tilde{A}$ is singular, and has a one dimensional null-space. The null-vector of $\tilde{A}$ is given by

$$\tilde{p} = \begin{bmatrix} \tilde{p}_0 \\ -T_2^{-1} \tilde{E}_2 \tilde{p}_0 \end{bmatrix}. \tag{4.31}$$

We note that the $\tilde{p}_{ij}$ are not necessarily positive. Let us define the generalized inverse $\tilde{A}^+$ of $\tilde{A}$ as

$$\tilde{A}^+ = \begin{bmatrix} \tilde{A}_0^+ & \mathbf{0} \\ -T_2^{-1}\tilde{E}_2\tilde{A}_0^+ & T_2^{-1} \end{bmatrix}. \tag{4.32}$$

We have

Lemma 2

(i) $Im(\tilde{A}) = \{x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in R^{N_A} | x_1 \in Im(\tilde{A}_0)\}$.

(ii) $\tilde{A}^+$ is invertible on $Im(\tilde{A})$. More precisely, for all $y \in Im(\tilde{A})$, there exists a unique $x \in Im(\tilde{A})$, such that

$$\tilde{A}^+ x = y, \tag{4.33}$$

where $x = \tilde{A}y$. Thus for all $y \in Im(\tilde{A})$,

$$\tilde{A}\tilde{A}^+ y = \tilde{A}^+\tilde{A}y = y. \tag{4.34}$$

(iii) For all $y \in R^{N_A}$, there exist a unique α and $\xi \in Im(\tilde{A})$ such that,

$$y = \alpha\tilde{p} + \tilde{A}^+\xi. \tag{4.35}$$

(iv) Let p be the solution to (4.8) - (4.10), then there exist a unique $\alpha \neq 0$ and $\xi \in Im(\tilde{A})$ such that,

$$p = \alpha\tilde{p} + \tilde{A}^+\xi. \tag{4.36}$$

Proof: We first note that $\tilde{A}_0$ is the generating matrix of a 2-queue free model, hence (i) - (iii) follow easily from the fact that T_2 is non-singular and the remarks we made in (2.15) - (2.16). Thus let us prove (iv). By (iii), it suffices to show that $\alpha \neq 0$. Suppose $\alpha = 0$, then $p = \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \tilde{A}^+\xi \in Im(\tilde{A})$. By (i), $p_0 \in Im(\tilde{A}_0)$, hence $\mathbf{1}^* p_0 = 0$, contradicting (4.11). □

We remark that even though $\tilde{A}$ cannot be symmetrized, we still have a decomposition of the state space as in (4.35). Notice that in (4.21), $\tilde{A}$ is in block lower triangular form. Since $\tilde{A}_0$ and T_2 are separable, and in view of (4.14) and (4.22), $\tilde{E}_2$ is sparse, thus $\tilde{A}^+x$ can be computed easily for any $x \in Im(\tilde{A})$. We remark that $\tilde{E}_2$ picks up the Dirichlet data on τ.

Using (iv) in Lemma 2 and since p is unique up to a multiple constant, we may let

$$p = \tilde{p} + \tilde{A}^+\xi_0, \tag{4.37}$$

and normalize it by (4.9) after we find it. Putting this Ansatz into (4.8), we get

$$(I + \tilde{R}\tilde{A}^{+})\xi_0 = -\tilde{R}\tilde{p}. \tag{4.38}$$

It can easily be checked that $\tilde{R}\tilde{p} \neq 0$. We remark that since $\tilde{R}$ has m non-zero rows, where m is given by (4.26), (4.38) is practically an m by m system.

Let us calculate the cost of computing $\tilde{R}\tilde{A}^{+}x$ for any x. Since $\tilde{R}$ has at most $2m$ non-zero entries, $\tilde{R}x$ can be computed in $2m$ operations. Next let us consider solving

$$\tilde{A}\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix},$$

where $x_0 \in Im(\tilde{A}_0)$. Notice that by (4.21), $\tilde{A}_0 y_0 = x_0$. Since $\tilde{A}_0$ is separable, this system can be solved in $O(n_2^2)$ operations. Let us remark that when $s_1 \ll n_2$, we can diagonalize $\tilde{G}_1$ and then solve the resulting tridaigonal systems with respect to G_2. This requires only $4n_2 s_1$ operations, see Chan [5]. Having found y_0, we solve

$$T_2 y_1 = x_1 - \tilde{E}_2 y_0. \tag{4.39}$$

Since $\tilde{E}_2$ is sparse, the right hand side of this equation can be computed in $n_2 - s_2$ operations. Since T_2 is separable too, y_1 can be solved by first diagonalizing V_2 and then solving the resulting tridiagonal systems. We note that we can use the Fast Fourier transform to perform the diagonalization. This follows from the fact that V_2 is the generating matrix of a 1-queue single-server model with the service rate $s_2\mu_2$. Thus the work for solving (4.39) is roughly $5(n_2 - s_2)(n_1 - s_1) + 2(n_2 - s_2)\log(n_2 - s_2)$. The first term here is the work required to solve the resulting tridiagonal systems. Combining these results, we see that the work required to compute $\tilde{R}\tilde{A}^{+}x$ is

$$4n_2 s_1 + 5(n_2 - s_2)(n_1 - s_1) + 2(n_2 - s_2)\log(n_2 - s_2), \tag{4.40}$$

and the memory requirement is

$$s_1^2 + O(n_i - s_i). \tag{4.41}$$

We note that there are many other viable separable preconditioners. For example, instead of solving the Dirichlet-Neumann problem corresponding to T_2, we may solve a Dirichlet problem on Ω_2. This is also a separable problem. However, using the continuous analogy, see Bjϕrstad and Widlund [3], we expect that this preconditioner will not lead to an optimal method.

4.2 Embedding of the State-Space

Capacitance matrix methods have been developed for solving elliptic problems in irregular regions such as the L-shaped region in Figure 1, see O'Leary and Widlund [13], Proskurowski and Widlund [15] and Astrakhantsev [1]. The idea

is to embed the state-space into a larger space where there is a separable preconditioner. Here we design an algorithm that adopts this approach.

Recall that A_0 resembles a Neumann problem on the whole rectangular region $[0, n_1 - 1] \times [0, n_2 - 1]$. If we order the states in the L-shaped region first, then we can write

$$A_0 = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{4.42}$$

where A_{11} and A_{22} are square matrices of dimension N_A and $N - N_A$ respectively. We claim that A_{22} is nonsingular. In fact, by the definition of A_0,

$$A_{22} = V_1 \otimes I_{s_2} + I_{n_1 - s_1 - 1} \otimes \tilde{G}_2, \tag{4.43}$$

where V_1 is given by (4.29) and $\tilde{G}_2$, of dimension s_2, is the principal submatrix of G_2 obtained by deleting the last $n_2 - s_2$ columns and rows of G_2. Since V_1 and $\tilde{G}_2$ are irreducibly diagonally dominant, A_{22} is nonsingular.

Consider the N by N matrix

$$A_N \equiv \begin{bmatrix} A & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix}, \tag{4.44}$$

where A is the generating matrix in (4.8). Clearly if p is the solution to (4.8) - (4.10), then $p_N = \begin{bmatrix} p \\ \mathbf{0} \end{bmatrix}$ is the unique solution to

$$A_N p_N = 0, \tag{4.45}$$

$$\mathbf{1}_N^* p_N = 1, \tag{4.46}$$

and

$$(p_N)_{k,j} \geq 0, \quad 1 \leq k, j \leq N. \tag{4.47}$$

Here $\mathbf{1}_N = (1, 1, ..., 1) \in R^N$. Since $\mathbf{1}_N^* A_0 = 0$ and $\mathbf{1}^* A = 0$, it follows that

$$\mathbf{1}_N^* A_N = 0. \tag{4.48}$$

Moreover, by (2.15) and (4.46), there exists a unique $\xi_0 \in Im(A_0)$, such that

$$p_N = p_0 + A_0^+ \xi_0, \tag{4.49}$$

where p_0 is given by (2.8). Define

$$R_N \equiv A_N - A_0 = \begin{bmatrix} A - A_{11} & \mathbf{0} \\ -A_{21} & \mathbf{0} \end{bmatrix}. \tag{4.50}$$

Then (4.45) and (4.49) imply that

$$(I + R_N A_0^+) \xi_0 = -R_N p_0. \tag{4.51}$$

By (4.50) and Lemma 1, the matrix $(I + R_N A_0^+)$ is nonsingular. Thus we can solve (4.51) either by direct or iterative methods.

We claim that the matrix R_N is sparse. In fact, by the definition of A_0

$$(A_{21})_{k,j} = \begin{cases} -\lambda_1 & (k_1 - 1, k_2) \text{ and } (j_1, j_2) \in l_1, \\ -s_2\mu_2 & (k_1, k_2 + 1) \text{ and } (j_1, j_2) \in l_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.52}$$

On the other hand, it is straightforward to check that

$$A_{11} = \bar{A} + D, \tag{4.53}$$

where $\bar{A}$ is given by (4.13) and D is a diagonal matrix such that

$$D_{jj} = \begin{cases} \lambda_1 & (j_1, j_2) \in l_1, \\ s_2\mu_2 & (j_1, j_2) \in l_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.54}$$

By (4.12) and (4.53), $A - A_{11} = \bar{R} - D$. From (4.16) - (4.18), we see that $\bar{R} - D$ still has $N_{\bar{R}}$ nonzero rows and every such rows has at most two nonzero entries. Here $N_{\bar{R}}$ is given by (4.19). Moreover, by (4.50) and (4.52), the number of nonzero rows in R_N is equal to

$$N_R \equiv N_{\bar{R}} + s_2 = n_1 + n_2 + s_2 - s_1 - 1, \tag{4.55}$$

and every such rows has at most two nonzero elements. Hence R_N is sparse.

Using the sparsity of R_N, we can reduce the dimension of the problem (4.51) to N_R. If conjugate gradient type methods are used, then in each step, we have to compute a vector of the form $R_N A_0^+ \xi$ where $\xi \in Im(R_N)$. We note that though ξ is sparse, the computation of $R_N A_0^+ \xi$ still requires $O(n_i^3)$ work and $O(n_i^2)$ storage spaces. These counts are considerably higher than the counts given in (4.40) and (4.41).

This algorithm has not yet been tested. However, considering the fast convergence of the capacitance method for elliptic problems and the continuous analogue of the queueing models, we conjecture that this algorithm also has a fast convergence rate.

5 Numerical Results

In this section, we report on the numerical results for the models discussed in previous sections. All the computations were carried out on the Cyber-760 at the Mathematics and Computing Laboratory of the Courant Institute. Single precision, between fourteen and fifteen decimal digits, was used throughout. Craig's method, used in the computations, is a version of the ordinary conjugate gradient method applied to the normal equations; see Elman [9]. The Orthodir

method is a generalized conjugate gradient method in which the transpose of the matrix is not needed; see Young and Jea [17]. Convergence is said to occur at the k-th step if

$$\frac{||r_k||_2}{||r_0||_2} \leq \text{ tolerance.}$$

Here r_k is the residual vector at the k-th step and

$$||x||_2^2 \equiv \frac{1}{m}\sum_{i=1}^{m} x_i^2 \quad \forall x \in R^m.$$

The initial iterant x_0 is chosen to be identically zero.

Tables 1 gives the results of our method when applied to the 3-queue model discussed in § 2. The number of iterations increases like order $O(\log n_i)$. Table 2 shows the time in seconds required in each phase of the algorithm. Tables 3 and 4 compare our method with the point SOR method. ω^* is the optimal relaxation factor obtained experimentally. We see that our method performs much better than the point SOR method especially when the s_i are small.

Tables 5 and 6 report on the performance of the family of preconditioners discussed in § 3. The parameter β in the tables indicates which preconditioner we are using. If $\beta \neq 0$, the preconditioner is $\tilde{A}_\beta$ which is nonsingular and is given by (3.8). If $\beta = 0$, the preconditioner is defined by (3.10). Recall that when $\beta = 1$, the preconditioner resembles a Dirichlet problem while when $\beta = 0$, it resembles a Neumann problem. We see that the number of iterations decreases as $|\beta| \to 0$. However, we remark that for sufficiently small β, arithmetic overflow will occur. This is because by (3.7) and (3.8), the smallest eigenvalue of $\tilde{A}_\beta$ tends to zero as $|\beta| \to 0$. For comparison, we also report in the tables the number of iterations required if A_0 is used as preconditioner.

For the L-shpaed region described in § 4, tables 7 and 8 give the number of iterations required for convergence when the method introduced in § 4.1 is used. In all cases, we see that the number of iterations increases at most like $O(\log n_i)$.

Table 1: Number of iterations by the Orthodir Method (tolerance = 10^{-6})

Parameters		$s_i\mu_i = \lambda_i + (n_i-1)^{-\alpha}, \lambda_i = 1, i = 1,2,3$											
(n_1,n_2,n_3)	N	s_i	α			s_i	α			s_i	α		
			1	2	3		1	2	3		1	2	3
(4,4,4)	64	1	10	10	10	3	9	9	9	3	9	9	9
(8,8,8)	512	1	14	14	14	3	14	14	14	6	13	13	13
(16,16,16)	4096	1	18	18	18	3	18	18	18	9	17	17	17

Table 2: Time in seconds by the Orthodir Method (tolerance = 10^{-6})

Parameters	$s_i\mu_i = \lambda_i + (n_i - 1)^{-2}, \lambda_i = 1, s_i = 3, i = 1, 2, 3$		
$n_1 = n_2 = n_3$	4	8	16
Initialization	0.015	0.065	0.308
Iteration	0.362	3.492	28.407
No. of iterations	9	14	18
Time per iteration	0.0402	0.249	1.578
Generating p	0.039	0.407	4.526
Total time	0.416	3.964	33.241

Table 3: Comparison with the Point SOR method

Parameters			$s_i\mu_i = \lambda_i + (n_i - 1)^{-2}, \lambda_i = 1, i = 1, 2, 3$								
Method			point SOR: Initial guess $p = 0$								Orthodir
n_i	s_i	N	ω^*	Relaxation factor ω							Iterations
				1.0	1.4	1.5	1.6	1.7	1.8	1.9	ω^*	
4	1	64	1.700	474	206	159	115	69	190	**	69	10
4	3	64	1.593	183	76	54	34	53	104	**	30	9
8	1	512	1.831	**	**	907	660	489	300	**	242	14
8	7	512	1.715	458	199	154	111	64	77	305	49	12

** more than 1000 iterations

Table 4: Time Comparison between the Orthodir and the Point SOR methods

Parameters	$s_i\mu_i = \lambda_i + (n_i - 1)^{-\alpha}, \lambda_i = 1, i = 1, 2, 3, \alpha = 2$					
Problem	$n_i = 4, s_i = 3$		$n_i = 8, s_i = 7$		$n_i = 8, s_i = 1$	
Dimension N	64		512		512	
Method	Orthodir	pt SOR	Orthodir	pt SOR	Orthodir	pt SOR
ω^*	...	1.593	...	1.715	...	1.831
No. of iterations	9	30	12	49	14	242
Time for iteration	0.364	0.498	2.822	5.936	3.365	31.225
Time per iteration	0.0405	0.0166	0.2352	0.1211	0.2404	0.1290
Total time	0.420	0.529	3.274	5.997	3.815	31.282

Table 5: Number of Iterations by the Craig's Method, (tolerance = 10^{-10})

Parameters	$\frac{\lambda_i}{\mu_i} = 1 - \frac{1}{2}(n_i - 1)^{-2}, \mu_i = s_i = 1, i = 1, 2$											
(n_1, n_2)	β											A_0
	1.00	.75	.50	.25	.10	.01	.001	0	-.01	-.25	-.75	
(8,8)	8	8	8	8	8	8	9	10	8	9	9	7
(16,16)	16	16	15	13	12	9	9	12	9	16	22	8
(32,32)	28	26	23	20	15	10	10	14	10	23	39	10
(64,64)	48	43	37	28	20	12	11	17	12	38	**	12

** more than 30 iterations

Table 6: Number of Iterations by the Orthodir Method, (tolerance = 10^{-10})

Parameters	$\frac{\lambda_i}{\mu_i} = 1 - \frac{1}{2}(n_i - 1)^{-2}, \mu_i = s_i = 1, i = 1, 2$											
(n_1, n_2)	β											A_0
	1.00	.75	.50	.25	.10	.01	.001	0	-.01	-.25	-.75	
(8,8)	7	8	8	8	8	8	**	9	8	8	8	7
(16,16)	15	15	15	16	16	16	**	17	16	16	16	15
(32,32)	24	23	22	21	21	21	**	26	21	23	28	21
(64,64)	36	33	31	27	25	24	**	30	25	32	44	25

** more than 30 iterations

Table 7: Number of Iterations by the Orthodir Method, (tolerance = 10^{-6})

Parameters	$s_i\mu_i = \lambda_i + (n_i - 1)^{-\alpha}, \lambda_i = 1, i = 1, 2$					
(n_1, n_2)	$\alpha = 1$		$\alpha = 2$		$\alpha = 3$	
	(s_1, s_2)	Iterations	(s_1, s_2)	Iterations	(s_1, s_2)	Iterations
(10,10)	(2,2)	11	(5,5)	10	(4,4)	11
(20,20)	(4,4)	15	(5,5)	15	(8,8)	13
(40,40)	(8,8)	17	(5,5)	18	(16,16)	16
(80,80)	(16,16)	20	(5,5)	22	(32,32)	18

Table 8: Number of Iterations by the Orthodir Method, (tolerance = 10^{-6})

Parameters	$s_i\mu_i = \lambda_i + (n_i - 1)^{-\alpha}, \lambda_i = 1, i = 1, 2$					
(s_1, s_2)	$\alpha = 3$		$\alpha = 2$		$\alpha = 2$	
	(n_1, n_2)	Iterations	(n_1, n_2)	Iterations	(n_1, n_2)	Iterations
(10,10)	(17,18)	12	(27,28)	15	(15,15)	11
(10,10)	(25,26)	14	(35,36)	16	(30,30)	16
(10,10)	(41,42)	17	(51,52)	18	(60,60)	19
(10,10)	(73,74)	20	(83,84)	21	(120,120)	23

Acknowledgements

I would like to thank Professor Olof Widlund for his help in the preparation of this paper.

References

1. Astrakhantsev, G., *Method of Fictitious Domains for a Second-order Elliptic Equation with Natural Boundary Conditions.* U.S.S.R. Comput. Maths Math. Phys., V18, 1978, pp. 114-121.

2. Banegas A., *Fast Poisson Solvers for Problems with Sparsity.* Math. Comp., V32, 1978, pp. 441-446.

3. Bjϕrstad, P., & Widlund, O., *Iterative Methods for the Solutions of Elliptic Problems on Regions Partitioned into Substructures.* SIAM J. Numer. Anal., V23, 1986, pp. 1097-1120.

4. Buzbee, B., Dorr, F., George, J., & Golub, G., *The Direct Solution of the Discrete Poisson Equations on Irregular Regions.* SIAM J. Numer. Anal., V8, 1971, pp. 722-736.

5. Chan, R., *Iterative Methods for Overflow Queueing Models.* NYU Comp. Sci. Dept. Tech. Rep., # 171, 1985.

6. Chan, R., *Iterative Methods for Overflow Queueing Models I.* Numer. Math., V51, 1987, pp. 143-180.

7. Concus, P., Golub, G., & O'Leary, D., *A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations.* Proc. Symposium on Sparse Matrix Computations. Bunch, J.R., & Rose, D.J., eds. Academic Press, 1976.

8. Dryja, M., *A Capacitance Matrix Method for Dirichlet Problem on Polygon Region.* Numer. Math., V39, 1982, pp. 51-64.

9. Elman, H., *Iterative Methods for Large, Sparse Nonsymmetric Systems of Linear Equations.* Ph.D thesis. Department of Computer Science, Yale University, 1982.

10. Funderlic, R., & Mankin, J., *Solution of Homogeneous Systems of Linear Equations arising from Compartmental Models.* SIAM J. Sci. Statist. Comput., V2, 1981, pp. 375-383.

11. Kaufman, L., Seery, J., & Morrison, J., *Overflow Models for Dimension PBX Feature Packages.* The Bell System Technical Journal, V60, 1981, pp. 661-676.

12. Kaufman, L., *Matrix Methods for Queueing Problems.* SIAM J. Sci. Statist. Comput., V4, 1982, pp. 525-552.

13. O'Leary, D., & Widlund, O., *Capacitance Matrix Methods for the Helmholtz Equation on General Three-dimensional Regions.* Math. Comp., V33, 1979, pp. 849-879.

14. Parlett, B., *The Symmetric Eigenvalue Problem.* Prentice-Hall, New Jersey, 1980.

15. Proskurowski, W., & Widlund, O., *A Finite Element-Capacitance Matrix Method for the Neumann Problem for Laplace's Equation.* SIAM J. Sci. Statist. Comput., V1, 1980, pp.410-425.

16. Varga, R., *Matrix Iterative Analysis.* Prentice-Hall, New Jersey, 1962.

17. Young & Jea. *Generalized Conjugate Gradient Acceleration of Non-symmetric Iterative method.* Linear Algebra Appl., V34, 1980, pp. 159-194.