Boundary Value Methods for Transient Solutions of Queueing Networks with Variant Vacation Policy

Raymond H. Chan[a,1], Spike T. Lee[a,1], Hai-Wei Sun[b,2,*]

[a] *Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China*
[b] *Department of Mathematics, University of Macau, Macao, China*

Abstract

Boundary value methods are applied to find transient solutions of $M/M/2$ queueing systems with two heterogeneous servers under a variant vacation policy. An iterative method is employed to solve the resulting large linear system and a Crank-Nicolson preconditioner is used to accelerate the convergence. Numerical results are presented to demonstrate the efficiency of the proposed method.

Keywords: Queueing systems, heterogeneous servers, variant vacation policy, boundary value method, Crank-Nicolson preconditioner

1. Introduction

Queueing networks with vacation were proposed in the 1970s to overcome the deficiency of classical queueing networks in modeling complex hi-tech systems [19]. Server vacations may literally mean a lack of work, or figuratively stand for server failure, server maintenance, or server taking another assigned job, and hence the introduction of server vacations makes waiting-line systems more lifelike. The applications of vacation queueing systems lie in various areas such as flexible manufacturing systems, lane control at

*Corresponding author
Email addresses: rchan@math.cuhk.edu.hk (Raymond H. Chan), thlee@math.cuhk.edu.hk (Spike T. Lee), HSun@umac.mo (Hai-Wei Sun)

[1] Research supported by the HKRGC Grant CUHK 400510.
[2] Research partially supported by the research grant 033/2009/A from FDCT of Macao and MYRG140(Y1-L2)-FST11-SHW from University of Macau.

border-crossing stations, and data transfer in telecommunication systems. The thorough development of queueing networks with vacation can be found in survey papers by Teghem [18], Doshi [10, 11], and the monographs by Takagi [17], and Tian and Zhang [19].

Homogeneity of service rates is a general assumption in the study of multiserver queueing system and it ensures that all servers in the system provide services at an identical rate. However, the hypothesis of homogeneous systems is feasible only when the service process is mechanically or electronically controlled. For human servers, they are more likely to perform the same assignment at different service rates. Therefore *heterogeneous* servers are introduced and their service time distributions may be different for different servers. The combination of server vacations and heterogeneous servers is more practical in real-life situations [14, 21, 22].

In [23], Yue *et al.* proposed an $M/M/2$ queueing system with one queue and two heterogeneous servers under a variant vacation policy, in which the two servers will *simultaneously* take *at most* J vacations when the system becomes empty. They carried out a steady-state analysis and obtained the stationary distributions of system size and mean system size. Moreover, they studied the distribution of the amount of vacations taken, and the conditional stochastic decomposition properties of queue length and waiting time. The analytic results in [23], however, are based on the assumption of infinite queueing spaces which may not be practical in general.

For the sake of practicality, in this paper, we aim to consider the problem with finite queueing spaces and find the transient solution of the queueing system in [23]. It is well known that the transient solution of a queueing system can be numerically approximated by discretizing the Kolmogorov backward equation and solving the resulting ordinary differential equation (ODE) system [9, 16]. Classical initial value methods (IVMs) like Runge–Kutta methods are natural candidates but they are computationally more expensive than multistep methods of comparable accuracy. In this work, we follow the idea in [7] and apply the boundary value methods (BVMs) to solve the ODE systems. BVMs are the generalization of implicit linear multistep formulas (LMFs), and by using those with unconditional stability [5] one can disregard the restrictions on the step sizes for stability reasons. However, temporal discretization with BVMs requires solutions of larger linear systems than with Runge–Kutta methods or LMFs used as IVMs. Fortunately, owing to the block tridiagonal structure of the transition rate matrix, the resulting linear system is sparse and therefore we could resort to iterative methods.

Preconditioning techniques were long used to speed up the convergence process of iterative methods when solving large sparse linear systems produced by BVMs [4]. Over the years, different preconditioners were proposed, including the well-known T. Chan circulant preconditioner [2, 8], the P-circulant preconditioner [2], the Strang-type circulant preconditioner [2, 6], the skew-circulant preconditioner [3], and recently the Crank-Nicolson (CN) preconditioner [13, 20]. In [20], the CN preconditioner is paired with BVMs for pricing options in the jump-diffusion model and the numerical results therein show that the CN preconditioner contributes to smaller computational cost and fewer iterations than the Strang-type preconditioner. In this paper, we mainly discuss the usage of the CN preconditioner because we will see from the numerical results that it requires cheaper computational cost than other methods.

The rest of the paper is organized as follows. In section 2, we outline the two-server queueing system and its variant vacation policy. In section 3, we briefly introduce the BVMs and apply them to discretize the Kolmogorov backward equation to obtain an ODE system. In section 4, we form the CN preconditioner and study some of its properties when used in the iterative method. In section 5, we present the numerical results. In section 6, we give some concluding remarks and ideas of possible future work.

2. Transient solution for queueing network with variant vacation policy

In this paper, we consider the $M/M/2$ queueing system with two heterogeneous servers under a variant vacation policy proposed in [23]. Customers arrive and join a single queue according to the first-come first-serviced (FCFS) principle. The arrival of customers is modeled by the Poisson process with rate λ. When the system becomes empty, the two servers will simultaneously take a vacation of length V, where V is an exponentially distributed variable with parameter θ. When the servers are back from a vacation, they either resume working immediately if they find at least one customer waiting in the queue, or leave for another vacation of the same length V. The two servers will only take at most J vacations and after that, they will stay active in the system to provide services even when the system becomes empty.

The service rates of the two heterogeneous servers are modeled by exponential distributions with rates μ_1 and μ_2 for Server 1 and Server 2 respectively. Note that $\mu_1 \neq \mu_2$ since the two servers are heterogeneous. When both servers are free at the moment, Server 1 will step up to serve the new arriving customer. Finally, all the stochastic processes involved in the system are assumed to be independent.

It is noted that $M/M/2$ vacation queueing systems are modeled by quasi-birth-and-death (QBD) processes, the generalization of the birth-and-death process from a one-dimensional state space to a multidimensional state space [19]. Let $X(t)$ be the number of customers in the system at time t and $L(t) = j,\ j = 0, 1, \ldots, J + 1$ be the status of the servers at time t. The state (i, j) means that $i \geq 0$ customers are in the system and both servers are taking the $(j+1)$-th vacation for $j = 0, 1, \ldots, J-1$. Moreover, the state $(0, J)$ means that the system is empty while both servers are free. The state $(1, J)$ means that one customer is in the system while Server 1 is busy and Server 2 is free. The state $(1, J+1)$ means that one customer is in the system while Server 2 is busy and Server 1 is free. The state (i, J) means that $i \geq 2$ customers are in the system while both servers are busy.

For the two-dimensional Markov process $\{(X(t), L(t)),\ t \geq 0\}$ with the state space

$$\Omega_0 = \{(1, J+1)\} \cup \{(i, j),\ i \geq 0,\ j = 0, 1, \ldots, J\},$$

the infinitesimal generator of the process is given by [23]:

$$Q_0 = \begin{bmatrix} B_{00} & B_{01} & & & & \\ B_{10} & B_{11} & B_{12} & & & \\ & B_{21} & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & A_2 & A_1 & \ddots \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix}.$$

The size of the submatrices A_0, A_1, A_2 is $(J+1)$-by-$(J+1)$. Their explicit forms are $A_0 = \lambda I_{J+1}$, where I_{J+1} is the identity matrix of size $(J+1)$-by-$(J+1)$,

$$A_1 = \begin{bmatrix} -(\lambda+\theta) & & & & \theta \\ & -(\lambda+\theta) & & & \theta \\ & & \ddots & & \vdots \\ & & & -(\lambda+\theta) & \theta \\ & & & & -(\lambda+\mu_1+\mu_2) \end{bmatrix},$$

and

$$A_2 = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \mu_1 + \mu_2 \end{bmatrix}.$$

The other submatrices which represent boundary states are

$$B_{00} = \begin{bmatrix} -(\lambda+\theta) & \theta & & & \\ & -(\lambda+\theta) & \theta & & \\ & & \ddots & \ddots & \\ & & & -(\lambda+\theta) & \theta \\ & & & & -\lambda \end{bmatrix},$$

$$B_{01} = \begin{bmatrix} \lambda & & & & 0 \\ & \lambda & & & 0 \\ & & \ddots & & \vdots \\ & & & \lambda & 0 \end{bmatrix}, \quad B_{10} = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ \mu_1 & & & 0 \\ \mu_2 & & & \end{bmatrix},$$

$$B_{11} = \begin{bmatrix} -(\lambda+\theta) & & & & \theta & \\ & -(\lambda+\theta) & & & \theta & \\ & & \ddots & & \vdots & \\ & & & -(\lambda+\theta) & \theta & \\ & & & & -(\lambda+\mu_1) & \\ & & & & & -(\lambda+\mu_2) \end{bmatrix},$$

$$B_{12} = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \\ & & & \lambda \end{bmatrix}, \quad B_{21} = \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & \mu_2 & \mu_1 \end{bmatrix},$$

where B_{00} is a $(J+1)$-by-$(J+1)$ matrix, B_{01} and B_{21} are $(J+1)$-by-$(J+2)$ matrices, B_{10} and B_{12} are $(J+2)$-by-$(J+1)$ matrices, and B_{11} is a $(J+2)$-by-$(J+2)$ matrix.

In [23], Yue *et al.* started from the matrix Q_0 and studied the underlying properties of the vacation queueing system. The steady-state analysis in [23] is built on the assumption of infinite queue capacity. In this paper, we are interested in finite queueing spaces and transient probabilities.

Let n be the number of queueing spaces in the system. Then the state space is trimmed down to

$$\Omega = \{(1, J+1)\} \cup \{(i, j),\ 0 \le i \le n,\ j = 0, 1, \ldots, J\}.$$

Suppose $p_{i,j}(t)$ is the probability that the network is in state (i, j) at time t. Let

$$p(t) = [p_{0,0}(t), \ldots, p_{0,J}(t), p_{1,0}(t), \ldots, p_{1,J+1}(t), p_{n,0}(t), \ldots, p_{n,J}(t)]^{\intercal},$$

then the Kolmogorov backward equation is

$$\frac{dp(t)}{dt} = Q^{\intercal} p(t), \tag{2.1}$$

where Q is the block tridiagonal transition rate matrix

$$Q = \begin{bmatrix} B_{00} & B_{01} & & & & & & \\ B_{10} & B_{11} & B_{12} & & & & & \\ & B_{21} & A_1 & A_0 & & & & \\ & & A_2 & A_1 & A_0 & & & \\ & & & A_2 & A_1 & \ddots & & \\ & & & & \ddots & \ddots & A_0 & \\ & & & & & A_2 & A_1 & A_0 \\ & & & & & & A_2 & A_1 + A_0 \end{bmatrix}. \tag{2.2}$$

From now on we let

$$N = (n+1)J + n + 2$$

denote the size of matrix Q. We remark that the matrix Q satisfies

$$Q\mathbf{1} = \mathbf{0}, \tag{2.3}$$

where $\mathbf{1}$ and $\mathbf{0}$ are $\mathbb{R}^N$ vectors of all ones and zeros respectively. Hence each row of Q sums to zero.

3. Boundary value method

In this section, we consider how to solve (2.1) by BVMs, which are numerical methods based on LMFs and renowned for high order accuracy and unconditional stability [2, 5]. For simplicity, we first consider solving a general IVP:

$$\begin{cases} u'(t) = g(t, u), \quad t \in [0, T], \\ u(0) = u_0, \end{cases} \tag{3.1}$$

by BVMs. Suppose that the time domain $[0, T]$ is divided into m timesteps with $h = T/m$ and $t_j = jh$, $j = 0, 1, \ldots, m$. Let u_j and g_j be the approximations to $u(t_j)$ and $g(t_j, u(t_j))$ respectively. By using a μ-step, and s-th order accurate LMF, we have the following relations:

$$\sum_{l=-\nu}^{\mu-\nu} \alpha_{l+\nu} u_{j+l} = h \sum_{l=-\nu}^{\mu-\nu} \beta_{l+\nu} g_{j+l}, \quad j = \nu, \ldots, m - \mu + \nu. \tag{3.2}$$

The BVM in (3.2) should be used with ν initial conditions and $(\mu - \nu)$ final conditions. However, the IVP (3.1) only gives the initial value u_0. To obtain the other initial and final values, we apply adequate difference methods and get the additional $(\mu - 1)$ equations which also preserve the s-th order accuracy:

$$\sum_{l=0}^{\mu} \alpha_l^{(k)} u_l = h \sum_{l=0}^{\mu} \beta_l^{(k)} g_l, \quad k = 1, \ldots, \nu - 1, \tag{3.3}$$

and

$$\sum_{l=0}^{\mu} \alpha_{\mu-l}^{(k)} u_{m-l} = h \sum_{l=0}^{\mu} \beta_{\mu-l}^{(k)} g_{m-l}, \quad k = m - \mu + \nu + 1, \ldots, m, \tag{3.4}$$

where α_l and β_l are determined by different BVMs [5]. Applying the s-th order accurate BVM formulas (3.2)–(3.4) to (3.1), we obtain a linear system in matrix form as follows:

$$L_s \mathbf{u} = h R_s \mathbf{g} + \mathbf{e}_1 u_0 + O(h^s), \tag{3.5}$$

where $\mathbf{u} = [u_0, u_1, \ldots, u_m]^\intercal$, $\mathbf{g} = [g_0, g_1, \ldots, g_m]^\intercal$, $\mathbf{e}_1 = [1, 0, \ldots, 0]^\intercal \in \mathbb{R}^{(m+1)}$, and L_s and R_s are $(m+1)$-by-$(m+1)$ banded matrices:

$$L_s = \begin{bmatrix} 1 & \cdots & 0 & & & & \\ \alpha_0^{(1)} & \cdots & \alpha_\mu^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & & \\ \alpha_0^{(\nu-1)} & \cdots & \alpha_\mu^{(\nu-1)} & & & & \\ \alpha_0 & \cdots & \alpha_\mu & & & & \\ & \alpha_0 & \cdots & \alpha_\mu & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \alpha_0 & \cdots & \alpha_\mu \\ & & & & \alpha_0^{(m-\mu+\nu+1)} & \cdots & \alpha_\mu^{(m-\mu+\nu+1)} \\ & & & & \vdots & \vdots & \vdots \\ & & & & \alpha_0^{(m)} & \cdots & \alpha_\mu^{(m)} \end{bmatrix}$$

and

$$R_s = \begin{bmatrix} 0 & \cdots & 0 & & & & \\ \beta_0^{(1)} & \cdots & \beta_\mu^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & & \\ \beta_0^{(\nu-1)} & \cdots & \beta_\mu^{(\nu-1)} & & & & \\ \beta_0 & \cdots & \beta_\mu & & & & \\ & \beta_0 & \cdots & \beta_\mu & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \beta_0 & \cdots & \beta_\mu \\ & & & & \beta_0^{(m-\mu+\nu+1)} & \cdots & \beta_\mu^{(m-\mu+\nu+1)} \\ & & & & \vdots & \vdots & \vdots \\ & & & & \beta_0^{(m)} & \cdots & \beta_\mu^{(m)} \end{bmatrix}.$$

See [2, 3, 5, 6, 7] for details.

Recall that our goal is to find the transient solution $p(T)$ in (2.1) for some finite time T, given the initial state $p(0)$. We adhere to the idea in [7] and adopt the BVMs to discretize (2.1). In this paper, we use the three-step ($\mu = 3$), fourth order ($s = 4$) extended trapezoidal rules of second kind

(ETR$_2$) [5] in (3.2)–(3.4). For the stability properties for ETR$_2$, readers are referred to the theoretical studies in [1].

Let p_k denote the discrete approximation of $p(t_k)$ for $k = 0, 1, \ldots, m$. In particular, p_0 is indeed the given initial state of the queueing system. The ETR$_2$ mainly involves

$$\frac{1}{12}(-p_{k-2} - 9p_{k-1} + 9p_k + p_{k+1}) = \frac{hQ^\intercal}{2}(p_{k-1} + p_k), \quad k = 2, 3, \ldots, m-1,$$

with the additional equations

$$\begin{aligned} \frac{1}{24}(-17p_0 + 9p_1 + 9p_2 - p_3) &= \frac{hQ^\intercal}{4}(p_0 + 3p_1), \\ \frac{1}{24}(p_{m-3} - 9p_{m-2} - 9p_{m-1} + 17p_m) &= \frac{hQ^\intercal}{4}(3p_{m-1} + p_m). \end{aligned}$$

We pile up the vectors p_k, $k = 0, 1, \ldots, m$ and let $\mathbf{p} = [p_0^\intercal, p_1^\intercal, \ldots, p_m^\intercal]^\intercal$. Combining all the ETR$_2$-related equations, we get the resulting ODE system

$$A\mathbf{p} \equiv (L_4 \otimes I_N - hR_4 \otimes Q^\intercal)\mathbf{p} = \mathbf{e}_1 \otimes p_0 \equiv \mathbf{b}, \tag{3.6}$$

where L_4 and R_4 are $(m+1) \times (m+1)$ matrices

$$L_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & & & \\ -\frac{17}{24} & \frac{9}{24} & \frac{9}{24} & -\frac{1}{24} & & & \\ -\frac{1}{12} & -\frac{9}{12} & \frac{9}{12} & \frac{1}{12} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{12} & -\frac{9}{12} & \frac{9}{12} & \frac{1}{12} \\ & & \frac{1}{24} & -\frac{9}{24} & -\frac{9}{24} & \frac{17}{24} \end{bmatrix}, \quad R_4 = \begin{bmatrix} 0 & 0 & & & & \\ \frac{1}{4} & \frac{3}{4} & & & & \\ & \frac{1}{2} & \frac{1}{2} & & & \\ & & \ddots & \ddots & & \\ & & & \frac{1}{2} & \frac{1}{2} & \\ & & & & \frac{3}{4} & \frac{1}{4} \end{bmatrix},$$

and $\otimes$ is the Kronecker product.

4. The CN preconditioner

After discretizing the Kolmogorov backward equation (2.1) by the fourth order accurate BVM, we obtain an ODE system (3.6). The size of the coefficient matrix A in (3.6) is

$$(m+1)N = (m+1)[(n+1)J + n + 2],$$

which is large especially when a small timestep is in use. However, it still retains plenty of sparsity. Hence we opt for the generalized minimal residual

(GMRES) method [15] and design an appropriate preconditioner for better convergence.

The CN preconditioner was firstly proposed by Hou and Sun [13]. Later, in [20], Yang *et al.* extended Hou-Sun's scheme and applied the GMRES method with a right CN preconditioner to solve a BVM-discretized problem for pricing options in a jump-diffusion model. The CN preconditioner is motivated by the fact that the CN timestepping scheme itself is a one-step and second order accurate BVM with the formula

$$-p_{k-1} + p_k = \frac{hQ^\intercal}{2}(p_{k-1} + p_k), \quad k = 1, 2, \ldots, m.$$

If we use the BVM version of the CN timestepping scheme, the resulting ODE system after discretizing (2.1) becomes

$$(L_c \otimes I_N - hR_c \otimes Q^\intercal)\mathbf{p} = \mathbf{b}, \tag{4.1}$$

where

$$L_c = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \quad \text{and} \quad R_c = \begin{bmatrix} 0 & 0 & & \\ 1/2 & 1/2 & & \\ & \ddots & \ddots & \\ & & 1/2 & 1/2 \end{bmatrix}.$$

Inspired by [20], we use the coefficient matrix in (4.1)

$$P \equiv L_c \otimes I_N - hR_c \otimes Q^\intercal \tag{4.2}$$

as the preconditioner for our ODE system (3.6). In particular, we use (4.2) as a right preconditioner in the GMRES method because we can derive theoretical properties of P. We remark that the CN preconditioner in (4.2) can be cheaply inverted step by step as it is only a one-step method.

We need the following two lemmas to guarantee the invertibility of (4.2) before using it as a preconditioner.

Lemma 1. *The eigenvalues of Q in* (2.2) *have nonpositive real parts.*

Proof: Let γ be an eigenvalue of Q. Suppose that q_{ij} is the (i, j)-th entry of Q for $i, j = 1, 2, \ldots, N$. From the expression of Q in (2.2), we find that

$$q_{ii} < 0 \quad \text{and} \quad q_{ij} > 0, \; j \neq i,$$

for each $i = 1, 2, \ldots, N$. Moreover, we have

$$|q_{ii}| = -q_{ii} = \sum_{j=1,\ j\neq i}^{N} q_{ij}$$

since the sum of each row of Q is zero according to (2.3). Applying the Gershgorin circle theorem [12] to Q, we have $\gamma \subset \bigcup_{i=1}^{N} D_i$, where

$$D_i = \left\{ z \in \mathbb{C} : \ |z - q_{ii}| \leq \sum_{j=1,\ j\neq i}^{N} |q_{ij}| \right\} = \{z \in \mathbb{C} : \ |z - q_{ii}| \leq |q_{ii}|\}.$$

Since all D_i's are closed disks lying on the left-half plane, γ has a nonpositive real part. □

Lemma 2. [13] *Let* $\gamma(Q^{\intercal})$ *denote an eigenvalue of* $Q^{\intercal}$*. If* $h \cdot \gamma(Q^{\intercal}) \neq 2$*, then the matrix* P *defined in* (4.2) *is nonsingular.*

We note that Q is real. Therefore, an eigenvalue of Q is also an eigenvalue of $Q^{\intercal}$ and hence Lemma 1 holds for $Q^{\intercal}$ as well. According to Lemmas 1 and 2, we have the following theorem.

Theorem 1. *The matrix* P *in* (4.2) *is nonsingular.*

The next theorem is about the convergence of the GMRES method when CN preconditioner is used.

Theorem 2. [20, Theorem 3.3] *Let* P *in* (4.2) *be the right preconditioner for solving* (3.6)*. We have*

$$\|AP^{-1}\mathbf{b} - \mathbf{b}\|_{\infty} = O(h^3).$$

Theorem 2 can be used to show that the residual of the first iteration in the GMRES method with right CN preconditioner decreases with the size of the timestep. See [20] for more details about the CN preconditioner.

For the operation cost per iteration, the main work of the GMRES method with right preconditioning is the matrix-vector multiplication $AP^{-1}\mathbf{z}$. Here the term $P^{-1}\mathbf{z}$ can be obtained by solving $(m+1)$ sparse linear systems of size N-by-N and it requires $O(Nm)$ operations [13]. If we replace P with circulant or skew-circulant preconditioners, the same term can be obtained by performing fast Fourier transforms and the computational cost is of order $O(Nm \log m)$ [2, 3, 6].

5. Numerical results

In this section, we try to find the transient probabilities $p(T)$, where T is a finite number, in the single-queue queueing system with two heterogeneous servers under a variant vacation policy [23]. Recall that n is the number of queueing spaces in the system, J is the number of vacations the two servers are allowed to take, λ is the Poisson distribution parameter for the arriving customers, μ_1 and μ_2 are distinct exponentially distributed service rates for the two servers, θ is the exponential distribution rate of the vacation time. We assume that the initial state of the system is $(0,0)$ and set

$$p_0 = [1, 0, \ldots, 0]^\intercal \in \mathbb{R}^N.$$

Two sets of parameters are put to the test:

- **Example 1.** $T = 10$, $n = 40$, $J = 30$, $\lambda = 2$, $\mu_1 = 1$, $\mu_2 = 2$ and $\theta = 1.5$;
- **Example 2.** $T = 10$, $n = 120$, $J = 10$, $\lambda = 5$, $\mu_1 = 2$, $\mu_2 = 4$ and $\theta = 3$;

From (2.3), the transition rate matrix Q has zero row sum and therefore is singular. Together with the consistency condition of the ETR_2, we know that the Strang-type preconditioner [2, 6] will also be singular [3]. In the numerical experiments, we only try out the P-circulant preconditioner [2] and the Strang-type skew-circulant preconditioner [3] for comparison purpose.

When solving the ODE system (3.6) by the GMRES method with a right preconditioner, we use a stopping criterion of 10^{-7}. In Table 1, we report the unpreconditioned iterations and the preconditioned iterations for P-circulant preconditioner, Strang-type skew-circulant preconditioner, and CN preconditioner under the labels "No", "P-circ", "skew" and "CN" respectively. The preconditioned iterations are fewer than the unpreconditioned ones under the effect of preconditioning. Out of the three chosen preconditioners, the CN preconditioner outperforms the other two with smaller iteration numbers. In addition to fewer iterations, the CN preconditioner also has the cheapest computational cost in each iteration as discussed before.

We also sketch the residuals of the first five iterations in the GMRES method with CN preconditioner in Figure 1. We see that the residuals are rapidly decreasing and in particular, the drops in both examples are more obvious in the beginning, contributing to a smaller iteration number.

Table 1: Iteration numbers of GMRES method with no preconditioner, P-circulant preconditioner, Strang-type skew-circulant preconditioner and CN preconditioner in Examples 1 & 2.

	Example 1				Example 2			
m	No	P-circ	skew	CN	No	P-circ	skew	CN
10	58	29	26	13	105	55	39	14
20	55	19	17	11	95	32	30	14
40	62	14	12	9	93	22	19	11
80	90	12	11	7	111	16	14	9
160	158	11	10	6	164	13	11	8

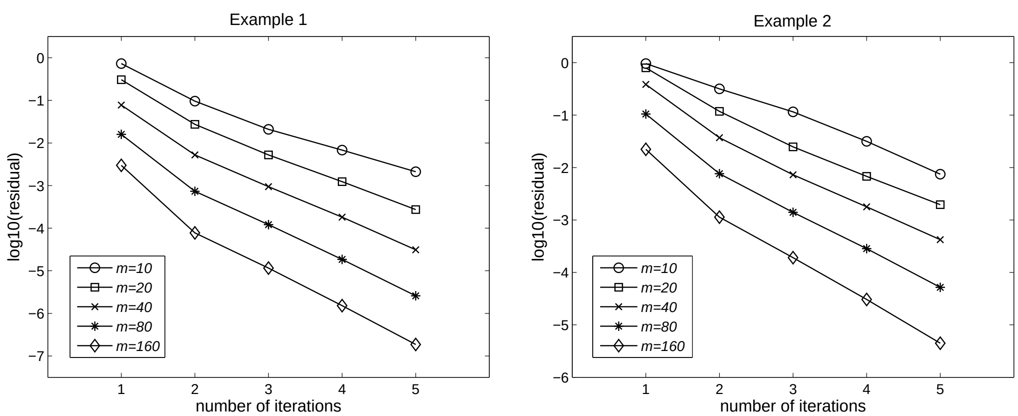

Figure 1: The residuals of the first five iterations in the GMRES method with CN preconditioner in Examples 1 and 2.

6. Concluding remarks

In this paper, we use the BVMs with CN preconditioner to find transient solutions of a vacation queueing system. Numerical results demonstrate that the CN preconditioner greatly helps decrease the iteration numbers in the GMRES method. Note that the queueing system considered in this paper has only one queue which makes the inversion of the preconditioner inexpensive. For future work, we will consider the problem with multiple queues, and employ other efficient methods to find the transient solution, as well as the steady-state solution.

Acknowledgements The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] L. Aceto, R. Pandolfi, Theoretical analysis of the stability for Extended Trapezoidal Rules, Nonlinear Anal. 71 (2009) e2521–e2534.

[2] D. Bertaccini, A circulant preconditioner for the systems of LMF-based ODE codes, SIAM J. Sci. Comput. 22 (2000) 767–786.

[3] D. Bertaccini, M. Ng, Block $\{\omega\}$-circulant preconditioners for the systems of differential equations, Calcolo 40 (2003) 71–90.

[4] L. Brugnano, D. Trigiante, A parallel preconditioning technique for boundary value methods, Appl. Numer. Math. 13 (1993) 277–290.

[5] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998.

[6] R. Chan, M. Ng, X. Jin, Strang-type preconditioners for systems of LMF-based ODE codes, IMA J. Numer. Anal. 21 (2001) 451–462.

[7] R. Chan, K. Ma, W. Ching, Boundary value methods for solving transient solutions of Markovian queueing networks, Appl. Math. Comput. 172 (2006) 690–700.

[8] R. Chan, X. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, PA, 2007.

[9] W. Ching, Iterative Methods for Queuing and Manufacturing Systems, Springer-Verlag London Ltd., London, 2001.

[10] B. Doshi, Queueing systems with vacations—a survey, Queueing Systems Theory Appl. 1 (1986) 29–66.

[11] B. Doshi, Single server queues with vacations, in: H. Takagi (Ed.), Stochastic Analysis of Computer and Communication Systems, North-Holland, Amsterdam, 1990, pp. 217–265.

[12] G. Golub, C. Van Loan, Matrix Computations, 3rd ed., The John Hopkins University Press, Baltimore, 1996.

[13] J. Hou, H. Sun, A preconditioner from Crank-Nicolson scheme for systems of LMF-based ODE code, in: D. Ding, X. Jin, H. Sun (Eds.), Recent Advances in Computational Mathematics, Higher Education Press, Beijing & International Press of Boston, Somerville, 2008, pp. 139–149.

[14] B. Krishna Kumar, S. Pavai Madheswari, An $M/M/2$ queueing system with heterogeneous servers and multiple vacations, Math. Comput. Modelling 41 (2005) 1415–1429.

[15] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, Philadelphia, 2000.

[16] W. J. Stewart, Introduction to the Numerical Solution of Markov Chains, Princeton University Press, Princeton, NJ, 1994.

[17] H. Takagi, Queueing Analysis: A Foundation of Performance Evaluation Volume 1: Vacation and Priority Systems, North-Holland, Amsterdam, 1991.

[18] J. Teghem Jr., Control of the service process in a queueing system, European J. Oper. Res. 23 (1986) 141–158.

[19] N. Tian, Z. Zhang, Vacation Queueing Models: Theory and Applications, Springer-Verlag, New York, 2006.

[20] S. Yang, S. Lee, H. Sun, Boundary value methods with the Crank-Nicolson preconditioner for pricing options in the jump-diffusion model, Int. J. Comput. Math. 88 (2011) 1730–1748.

[21] D. Yue, J. Yu, W. Yue, A Markovian queue with two heterogeneous servers and multiple vacations, J. Ind. Manag. Optim. 5 (2009) 453–465.

[22] D. Yue, W. Yue, A heterogeneous two-server network system with balking and a Bernoulli vacation schedule, J. Ind. Manag. Optim. (2010) 501–516.

[23] D. Yue, W. Yue, R. Tian, Analysis of two-server queues with a variant vacation policy, The 9th International Symposium on Operations Research and Its Applications, Jiuzhaigou, Chengdu, China, 2010, pp. 483–491.