Stochastic Systems
2013, Vol. 3, No. 1, 96–146
DOI: 10.1214/11-SSY029

MANY-SERVER QUEUES WITH CUSTOMER ABANDONMENT: NUMERICAL ANALYSIS OF THEIR DIFFUSION MODEL

By J. G. Dai* and Shuangchi He†

Cornell University‡ *and National University of Singapore*

We use a multidimensional diffusion process to approximate the dynamics of a queue served by many parallel servers. Waiting customers in this queue may abandon the system without service. To analyze the diffusion model, we develop a numerical algorithm for computing its stationary distribution. A crucial part of the algorithm is choosing an appropriate reference density. Using a conjecture on the tail behavior of the limit queue length process, we propose a systematic approach to constructing a reference density. With the proposed reference density, the algorithm is shown to converge quickly in numerical experiments. These experiments demonstrate that the diffusion model is a satisfactory approximation for many-server queues, sometimes for queues with as few as twenty servers.

1. Introduction. The focus of this paper is the numerical analysis of a multidimensional diffusion process that approximates the dynamics of a queue with many parallel servers. A many-server queue serves as a building block modeling operations of a large-scale service system. Such a service system could be a call center with hundreds of service agents, a hospital department with tens or hundreds of inpatient beds, or a computer cluster with many processors. When the customers of a service system are human beings, some of them may abandon the system before their service begins. The phenomenon of customer abandonment is ubiquitous because no one would wait for service indefinitely. As argued by Garnett et al. in [11], one must model customer abandonment explicitly in order for an operational model to be relevant for decision making. We model customer abandonment

Received April 2011.

*The work of J. G. Dai was supported in part by NSF grants CMMI-0727400, CMMI-0825840, CMMI-1030589, and CNS-1248117.

†The work of S. He was supported in part by MOE AcRF grant R-266-000-063-133 and NUS Global Asia Institute grant R-716-000-006-133.

‡On leave from Georgia Institute of Technology.

AMS 2000 subject classifications: Primary 60K25, 60J70, 65R20; secondary 68M20.

Keywords and phrases: Diffusion process, stationary distribution, phase-type distribution, many-server queue, heavy traffic, customer abandonment, quality- and efficiency-driven regime.

by assigning each customer a patience time. When a customer's waiting time for service exceeds his patience time, he abandons the queue without service.

The exact analysis of such a many-server queue has been largely limited to the $M/M/n+M$ model (also known as the Erlang-A model) that has a Poisson arrival process and exponential service and patience time distributions. See, e.g., [11]. However, as pointed out by Brown et al. in [1], the service time distribution in a call center appears to follow a log-normal distribution. In [34], the patience time distribution in a call center has been observed to be far from exponential, too. With a general service or patience time distribution, there is no finite-dimensional Markovian representation of the queue. Except computer simulation, there is no method that is able to exactly analyze such a queue either analytically or numerically. To deal with this challenge, the following strategies are adopted in this paper.

First, the service time distribution is restricted to be phase-type. As phase-type distributions can approximate any positive-valued distribution, such a queueing model is still relevant to practical systems. We focus on a $GI/Ph/n+GI$ queue with n identical servers. The first GI indicates that the customer interarrival times are independent and identically distributed (iid) following a general distribution, the Ph indicates that the service times are iid following a phase-type distribution, and the $+GI$ indicates that the patience times are iid following a general distribution. Second, we are particularly interested in a queue operated in the *quality- and efficiency-driven (QED)* regime. In this regime, the queue has a large number of servers and the arrival rate is high; the arrival rate and the service capacity are approximately balanced so that the mean waiting time is relatively short compared with the mean service time. Such a system has high server utilization as well as short customer waiting times and a small fraction of customer abandonment. Therefore, both quality and efficiency can be achieved in this regime. See [11] for more details on the QED regime. Third, rather than analyzing the many-server queue itself, we propose and analyze an approximate model. In this model, a multidimensional diffusion process is used to represent the scaled customer numbers in all service phases. We also develop a numerical algorithm to solve the stationary distribution of the diffusion process. This distribution is used to estimate the performance of the many-server queue. Numerical examples in Section 4 demonstrate that the diffusion model is accurate, even if the queue has as few as twenty servers.

Except for certain simple cases, the stationary distribution of a diffusion process has no explicit formula. The algorithm proposed in this paper is a variant of the one developed by Dai and Harrison in [5], which is used to compute the stationary distribution of a semimartingale reflecting Brownian motion (SRBM). As in [5], the starting point of our algorithm is the basic adjoint relationship that characterizes the stationary distribution of a diffusion process. With an appropriate reference density, the algorithm produces a stationary density that satisfies this relationship.

We set up a Hilbert space using the reference density. In this space, the stationary density of the diffusion process is orthogonal to an infinite-dimensional subspace H. A finite-dimensional subspace H_k is used to approximate H and a function orthogonal to H_k is numerically computed by solving a system of linear equations. This function is used to approximate the stationary density. There are two sources of numerical error from computation: *approximation error* and *round-off error*. Approximation error arises because H_k is an approximation of H. As H_k increases to H, this error decreases to zero. Round-off error occurs because the solution to the system of linear equations has error due to the finite precision of a computer. As H_k increases to H, the dimension of the linear system gets higher and the coefficient matrix becomes closer to singular. As a consequence, the round-off error increases. The condition number of the matrix is used as a proxy for the round-off error. Balancing approximation and round-off error is an important issue in our algorithm.

A properly chosen reference density is essential for the convergence of the algorithm. By convergence, we mean that the approximation error converges to zero as H_k increases to H. More importantly, a "good" reference density can make H_k converge to H quickly so that the resulting approximation error and round-off error are small simultaneously even though the dimension of H_k is moderate. To ensure the convergence of the algorithm, the reference density should have a comparable or slower decay rate than the stationary density. Since the stationary density is unknown, we make a conjecture on the tail behavior of the limit queue length process of many-server queues with customer abandonment. We conjecture that the limit queue length process has a Gaussian tail and the tail depends on the service time distribution only through its first two moments. This tail is used to construct a product-form reference density. The algorithm appears to converge quickly with this reference density, producing stable and accurate results. For comparison purposes, we also test the algorithm with a "naively" chosen reference density in Section 5.1. The algorithm fails to converge in that case.

Besides the proposed diffusion model, the major contribution of this work is the systematic approach that exploits the asymptotic tail behavior of queue length processes to constructing a reference density for the generic algorithm developed in [5]. Using a finite element implementation of their algorithm with our choice of the reference density, we demonstrate that the diffusion model is an adequate and tractable approximation for many-server queues.

Our diffusion model is obtained by replacing certain scaled renewal processes by Brownian motions. The replacement procedure is rooted in the limit theorems for many-server queues in heavy traffic. More specifically, the diffusion model is motivated by the diffusion limits proved in [6] and [26]. The theory of diffusion approximation for many-server queues can be traced back to the seminal paper [13] by Halfin and Whitt, where a diffusion limit was established for $GI/M/n$ queues. Garnett et al. proved a diffusion limit in [11] for $M/M/n + M$ queues that allows for customer abandonment. Whitt generalized this result in [32] to $G/M/n+M$ queues. Puhalskii and Reiman established a diffusion limit in [25] for $GI/Ph/n$ queues. Their result was extended in [6] to $G/Ph/n+GI$ queues with customer abandonment by Dai et al. Recently, Reed and Tezcan proved a diffusion limit for $GI/M/n + GI$ queues in [26]. In their framework, a refined limit process is obtained by scaling the patience time hazard rate function.

Harrison and Nguyen derived Brownian models for multiclass open queueing networks in [14]. Their diffusion models are SRBMs and are rooted in the conventional heavy traffic limit theorems pioneered by Iglehart and Whitt for serial networks in [16] and by Reiman for single-class networks in [28]. See [33] for a survey of limit theorems in the literature. For a two-dimensional SRBM living in a rectangle, Dai and Harrison proposed an algorithm in [4] for computing its stationary distribution. In [5], they extended the algorithm to an SRBM living in an orthant. The notion of a reference density was first introduced there to deal with the unbounded state space. Their finite-dimensional space H_k is constructed by multinominals of order up to k. With this choice of H_k, the algorithm sometimes appears numerically unstable. In such a case, the round-off error dominates the algorithm output while the approximation error is still significant. In [30], Shen et al. extended the algorithm in [4] to a hypercube state space of an arbitrary dimension. They employed a finite element method to construct H_k to avoid numerical instability. Their algorithm sometimes converges slowly because it does not use a reference density. A linear programming algorithm for computing the stationary distribution of a diffusion process was proposed by Saure et al. in [29]. Both SRBMs in an orthant and a diffusion approximation of many-server queues with two priority classes were investigated in their paper. Like the role of a reference density, the rescaling of variables is essential to the convergence of their algorithm.

The remainder of the paper is organized as follows. The diffusion model for $GI/Ph/n+GI$ queues is presented in Section 2. In Section 3, we begin with recapitulating the generic algorithm in [5] and then discuss how to choose a reference density by exploiting the tail behavior of a diffusion process. In Section 4, it is demonstrated via numerical examples that the diffusion model is a good approximation of many-server queues. Section 5 is dedicated to a few implementation issues arising from the algorithm. The paper is concluded in Section 6. We leave the finite element implementation of the algorithm to the appendix.

Notation. The symbols $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$ are used to denote the sets of positive integers, real numbers, and nonnegative real numbers, respectively. For $d, m \in \mathbb{N}$, $\mathbb{R}^d$ denotes the d-dimensional Euclidean space and $\mathbb{R}^{d\times m}$ denotes the space of $d\times m$ real matrices. We use $C_b^2(\mathbb{R}^d)$ to denote the set of real-valued functions on $\mathbb{R}^d$ that are twice continuously differentiable with bounded first and second derivatives. For $z, w \in \mathbb{R}$, we set $z^+ = \max\{z, 0\}$, $z^- = \max\{-z, 0\}$, and $z \wedge w = \min\{z, w\}$. All vectors are envisioned as column vectors. For a d-dimensional vector $x \in \mathbb{R}^d$, we use x_j for its jth entry and $\operatorname{diag}(x)$ for the $d \times d$ diagonal matrix with jth diagonal entry x_j. For a matrix M, M' denotes its transpose, M_{ij} denotes its (i, j)th entry, and $|M| = (\sum_{i,j} M_{ij}^2)^{1/2}$. We reserve I for the $d \times d$ identity matrix, e for the d-dimensional vector with all entries 1, and e^j for the d-dimensional vector with its jth entry 1 and all other entries 0. Given two functions φ and $\hat{\varphi}$ from $\mathbb{N}$ to $\mathbb{R}$, we write $\hat{\varphi}(n) = O(\varphi(n))$ as $n \to \infty$ if there exists a constant $\kappa > 0$ and some $n_0 \in \mathbb{N}$ such that $|\hat{\varphi}(n)| \le \kappa|\varphi(n)|$ for all $n > n_0$.

2. A diffusion model for many-server queues.

In this section, we introduce the $GI/Ph/n+GI$ queue and elaborate how to use a multidimensional diffusion process to approximate its dynamics. Diffusion processes and the basic adjoint relationship are reviewed in Section 2.1. The dynamics of the $GI/Ph/n+GI$ queue is studied in Section 2.2. We present the diffusion model in Section 2.3.

2.1. *Diffusion processes.*

Let d be a positive integer and $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a filtered probability space with filtration $\mathbb{F} = \{\mathcal{F}_t : t \ge 0\}$. Consider a d-dimensional diffusion process $X = \{X(t) : t \ge 0\}$ that satisfies the following stochastic differential equation

$$X(t) = X(0) + \int_0^t b(X(s))\,\mathrm{d}s + \int_0^t \sigma(X(s))\,\mathrm{d}B(s). \tag{2.1}$$

In (2.1), the drift coefficient b is a function from $\mathbb{R}^d$ to $\mathbb{R}^d$, the diffusion coefficient σ is a function from $\mathbb{R}^d$ to $\mathbb{R}^{d\times m}$, and $B = \{B(t) : t \ge 0\}$ is an m-dimensional standard Brownian motion with respect to $\mathbb{F}$. We assume that both b and σ are Lipschitz continuous, i.e., there exists a constant $c_1 > 0$ such that

$$|b(x) - b(y)| + |\sigma(x) - \sigma(y)| \le c_1 |x - y| \quad \text{for all } x, y \in \mathbb{R}^d. \tag{2.2}$$

Under condition (2.2), the stochastic differential equation (2.1) has a unique strong solution, i.e., there exists a unique process X on $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ such that (a) X is adapted to $\mathbb{F}$, (b) for each sample path $\omega \in \Omega$, $X(t, \omega)$ is continuous in t, and (c) for each $t \ge 0$, the stochastic differential equation (2.1) holds with probability 1. See [24] for more details. We also assume that σ is uniformly elliptic, i.e., there exists a constant $c_2 > 0$ such that

$$y' \Sigma(x) y \ge c_2 y' y \quad \text{for all } x, y \in \mathbb{R}^d, \tag{2.3}$$

where

$$\Sigma(x) = \sigma(x) \sigma'(x). \tag{2.4}$$

We are interested in the diffusion process that models the dynamics of a many-server queue. Such a process will be identified in Section 2.3 and the coefficients b and σ will be mapped out explicitly in terms of primitive parameters of the queue.

A probability distribution π on $\mathbb{R}^d$ is said to be a stationary distribution of X if $X(t)$ follows distribution π for each $t > 0$ whenever $X(0)$ has distribution π. Condition (2.3) is required to ensure the uniqueness of the stationary distribution. See [7] for more details. In the present paper, we assume that X has a unique stationary distribution π and that π has a density g with respect to the Lebesgue measure on $\mathbb{R}^d$. For a general diffusion process, there is no explicit formula for π, so we develop a numerical algorithm. As in [5], the starting point of the algorithm is the *basic adjoint relationship*

$$\int_{\mathbb{R}^d} \mathcal{G} f(x) \, \pi(\mathrm{d}x) = 0 \quad \text{for all } f \in C_b^2(\mathbb{R}^d), \tag{2.5}$$

where $\mathcal{G}$ is the generator of X defined by

$$\mathcal{G} f(x) = \sum_{j=1}^d b_j(x) \frac{\partial f(x)}{\partial x_j} + \frac{1}{2} \sum_{j=1}^d \sum_{\ell=1}^d \Sigma_{j\ell}(x) \frac{\partial^2 f(x)}{\partial x_j \partial x_\ell} \quad \text{for each } f \in C_b^2(\mathbb{R}^d) \tag{2.6}$$

and Σ is the covariance matrix given by (2.4). The following theorem is a consequence of Proposition 9.2 in [8].

THEOREM 1. *Let π be a probability distribution on $\mathbb{R}^d$ that satisfies (2.5). Then, π is a stationary distribution of X.*

In this paper, we conjecture that a stronger version of Theorem 1 is true.

CONJECTURE 1. *Let π be a signed measure on $\mathbb{R}^d$ that satisfies (2.5) and $\pi(\mathbb{R}^d) = 1$. Then, π is a nonnegative measure and consequently it is a stationary distribution of X.*

Our algorithm is to construct a function g on $\mathbb{R}^d$ such that

$$\int_{\mathbb{R}^d} g(x)\,\mathrm{d}x = 1 \quad \text{and} \quad \int_{\mathbb{R}^d} \mathcal{G}f(x)g(x)\,\mathrm{d}x = 0 \quad \text{for all } f \in C_b^2(\mathbb{R}^d). \tag{2.7}$$

Assuming that Conjecture 1 is true, g must be the unique stationary density of X. As a special case, the nonnegativity of the signed measure π that satisfies (2.5) for the piecewise Ornstein–Uhlenbeck (OU) process was proposed as an open problem in [3]. The piecewise OU process will be introduced in Section 2.3.

2.2. *The $GI/Ph/n + GI$ queue in the QED regime.* Let us focus on a queue with many parallel servers. In this queue, the service time distribution is restricted to be phase-type. All positive-valued distributions can be approximated by phase-type distributions.

Let p be a d-dimensional nonnegative vector whose entries sum to 1, ν be a d-dimensional positive vector, and P be a $d \times d$ sub-stochastic matrix. We assume that the diagonal entries of P are zero and that P is transient, namely, $I - P$ is invertible. Consider a continuous-time Markov chain with $d+1$ phases (or states) where phases $1, \dots, d$ are transient and phase $d+1$ is absorbing. For $j = 1, \dots, d$, the Markov chain starts in phase j with probability p_j. The amount of time that it stays in phase j is exponentially distributed with mean $1/\nu_j$. When it leaves phase j, the Markov chain enters phase $\ell = 1, \dots, d$ with probability $P_{j\ell}$ or enters phase $d+1$ with probability $1 - \sum_{\ell=1}^d P_{j\ell}$. The *phase-type distribution* with parameters (p, ν, P) is the distribution of the time from starting until absorption for this Markov chain. In particular, when P is a zero matrix, the associated phase-type distribution is a *hyperexponential distribution* with d phases.

In the $GI/Ph/n + GI$ queue, there are n identical servers working in parallel. The customer arrival process is a renewal process. Upon arrival, a customer enters service immediately if an idle server is available. Otherwise, he waits in a buffer with infinite waiting room that holds a first-in-first-out queue. The service times form a sequence of iid random variables, following a phase-type distribution. When a server finishes serving a customer, the server takes the leading customer from the buffer. When the buffer is empty, the server begins to idle. Each customer has a patience time. The patience times are iid following a general distribution. When a customer's waiting time in the buffer exceeds his patience time, the customer abandons the system with no service.

Let λ be the arrival rate and $1/\mu$ be the mean service time. The system is assumed to be operated in the QED regime, i.e., both the arrival rate λ and the number of servers n are large, while the traffic intensity $\rho = \lambda/(n\mu)$ is close to 1. Because customer abandonment is allowed, it is not necessary to assume $\rho < 1$ for the system to reach a steady state. For future purposes, we put

$$\beta = \sqrt{n}(1-\rho). \tag{2.8}$$

Assume that the phase-type service time distribution has parameters (p, ν, P). Each service time can be decomposed into a number of phases. When a customer is in service, he must be in one of the d phases. Let $Z_j(t)$ denote the number of customers in phase j service at t. In the steady state, customers in service are approximately distributed among the d phases following distribution γ, which is given by

$$\gamma = \mu R^{-1} p \quad \text{and} \quad R = (I - P') \operatorname{diag}(\nu). \tag{2.9}$$

One can check that $\sum_{j=1}^{d} \gamma_j = 1$ and γ_j is interpreted to be the fraction of phase j service load on the n servers.

Suppose that all customers, including those customers waiting in the buffer at time zero, sample their first service phases following distribution p upon arrival. One can stratify customers in the waiting buffer according to their first service phases. For $j = 1, \dots, d$, we use $W_j(t)$ to denote the number of waiting customers at time t whose service begins with phase j. Then,

$$Y_j(t) = Z_j(t) + W_j(t) \tag{2.10}$$

is the number of phase j customers in system, either waiting or in service. Let $Y(t)$ be the corresponding d-dimensional random vector and

$$\tilde{Y}(t) = \frac{1}{\sqrt{n}}(Y(t) - n\gamma). \tag{2.11}$$

We will approximate $\tilde{Y} = \{\tilde{Y}(t) : t \geq 0\}$ by a d-dimensional diffusion process.

The $GI/Ph/n + GI$ queue is driven by several primitive processes. Let $E = \{E(t) : t \geq 0\}$ be the arrival process, where $E(t)$ is the number of customer arrivals by time t. For $j = 1, \dots, d$, let $S_j = \{S_j(t) : t \geq 0\}$ be a Poisson process with rate ν_j, and $\phi_j = \{\phi_j(i) : i \in \mathbb{N}\}$ be a sequence of iid d-dimensional random vectors such that $\phi_j(i)$ takes e^ℓ with probability $P_{j\ell}$ and takes a zero vector with probability $1 - \sum_{\ell=1}^d P_{j\ell}$. Similarly, let $\phi_0 = \{\phi_0(i) : i \in \mathbb{N}\}$ be a sequence of iid d-dimensional random vectors such that $\phi_0(i)$ takes e^ℓ with probability p_ℓ. For $j = 0, \dots, d$, define the routing process $\Phi_j = \{\Phi_j(k) : k \in \mathbb{N}\}$ by

$$\Phi_j(k) = \sum_{i=1}^k \phi_j(i).$$

We assume that $Y(0), E, S_1, \dots, S_d, \Phi_0, \dots, \Phi_d$ are mutually independent.

For $j = 1, \dots, d$, let $T_j(t)$ be the cumulative amount of service effort received by customers in phase j service by time t. Clearly,

$$T_j(t) = \int_0^t Z_j(s)\,\mathrm{d}s \quad \text{for } t \geq 0. \tag{2.12}$$

Thus, $S_j(T_j(t))$ is equal in distribution to the cumulative number of phase j service completions by time t. (For more details, please refer to Section 4.1 of [6] on a perturbed system.) Let $L_j(t)$ be the cumulative number of phase j customers who have abandoned the system by time t, and let $L(t)$ be the corresponding d-dimensional vector. One can check that the d-dimensional process $Y = \{Y(t) : t \geq 0\}$ satisfies the following equation

$$Y(t) = Y(0) + \Phi_0(E(t)) + \sum_{j=1}^d \Phi_j(S_j(T_j(t))) - S(T(t)) - L(t), \tag{2.13}$$

where $S(T(t)) = (S_1(T_1(t)), \dots, S_d(T_d(t)))'$.

To derive the diffusion model, consider a scaled version of (2.13). For $t \geq 0$ and $j = 1, \dots, d$, we define several scaled processes by

$$\begin{aligned}
\tilde{E}(t) &= \frac{1}{\sqrt{n}}(E(t) - \lambda t), \quad \tilde{S}(t) = \frac{1}{\sqrt{n}}(S(nt) - n\nu t),\\
\tilde{Z}(t) &= \frac{1}{\sqrt{n}}(Z(t) - n\gamma), \quad \tilde{L}(t) = \frac{1}{\sqrt{n}} L(t),\\
\tilde{\Phi}_0(t) &= \frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} (\phi_0(i) - p), \quad \tilde{\Phi}_j(t) = \frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} (\phi_j(i) - p^j),
\end{aligned}$$

where p^j is the jth column of P'. By (2.8)–(2.13), we have

$$\begin{aligned}\tilde{Y}(t) = \tilde{Y}(0) - \beta\mu p t + p\tilde{E}(t) + \tilde{\Phi}_0\Big(\frac{E(t)}{n}\Big) + \sum_{j=1}^{d} \tilde{\Phi}_j\Big(\frac{S_j(T_j(t))}{n}\Big) \\ - (I - P')\tilde{S}\Big(\frac{T(t)}{n}\Big) - R\int_0^t \tilde{Z}(s)\,\mathrm{d}s - \tilde{L}(t).\end{aligned} \tag{2.14}$$

2.3. *The diffusion model.* In the diffusion model, the scaled primitive processes in (2.14) are replaced by Brownian motions. These approximations can be justified by the functional central limit theorem. Let B_E be a one-dimensional driftless Brownian motion with variance $\lambda c_a^2/n$, where c_a^2 is the squared coefficient of variation of the interarrival time distribution. Let $B_0, \ldots, B_d, B_S$ be d-dimensional driftless Brownian motions with covariance matrices $H^0, \ldots, H^d, \operatorname{diag}(\nu)$, respectively, where

$$H^0_{k\ell} = \begin{cases} p_k(1-p_\ell) & \text{if } k = \ell, \\ -p_k p_\ell & \text{otherwise} \end{cases}$$

and

$$H^j_{k\ell} = \begin{cases} P_{jk}(1-P_{j\ell}) & \text{if } k = \ell, \\ -P_{jk} P_{j\ell} & \text{otherwise} \end{cases}$$

for $j = 1, \ldots, d$. We assume that $\tilde{Y}(0), B_E, B_0, \ldots, B_d, B_S$ are mutually independent. In the diffusion model, the above Brownian motions take the places of the scaled primitive processes $\tilde{E}, \tilde{\Phi}_0, \ldots, \tilde{\Phi}_d, \tilde{S}$, respectively.

Let $Q(t)$ be the queue length (i.e., the number of waiting customers) at time t and

$$\tilde{Q}(t) = \frac{1}{\sqrt{n}} Q(t).$$

Then, $Q(t) = (e'Y(t) - n)^+$ or equivalently,

$$\tilde{Q}(t) = (e'\tilde{Y}(t))^+. \tag{2.15}$$

When n is large, these waiting customers are approximately distributed among the d phases according to distribution p (see Lemma 2 in [6]), i.e.,

$$W_j(t) \approx p_j Q(t) \quad \text{for } j = 1, \ldots, d.$$

It follows from (2.10) that

$$Z(t) \approx Y(t) - pQ(t).$$

By (2.11) and (2.15), this approximation has a scaled version

$$\tilde{Z}(t) \approx \tilde{Y}(t) - p(e'\tilde{Y}(t))^{+}. \tag{2.16}$$

The following approximations are also exploited in the diffusion model:

$$\frac{E(t)}{n} \approx \frac{\lambda t}{n} = \rho\mu t, \qquad \frac{T(t)}{n} \approx (\rho \wedge 1)\gamma t, \qquad \frac{S_j(T_j(t))}{n} \approx (\rho \wedge 1)\nu_j\gamma_j t. \tag{2.17}$$

Let $G(t)$ be the cumulative number of abandoned customers by time t and

$$\tilde{G}(t) = \frac{1}{\sqrt{n}} G(t).$$

These abandoned customers are also approximately distributed among the d phases by distribution p, i.e.,

$$\tilde{L}(t) \approx p\tilde{G}(t). \tag{2.18}$$

To approximate the process $\tilde{G} = \{\tilde{G}(t) : t \geq 0\}$, we exploit the idea of scaling the patience time hazard rate function, which was first proposed by Reed and Ward in [27] for single-server queues and was extended to many-server queues by Reed and Tezcan in [26].

We assume that the patience time distribution F satisfies

$$F(0) = 0 \tag{2.19}$$

and it has a bounded hazard rate function h, given by

$$h(t) = \frac{f_F(t)}{1 - F(t)} \quad \text{for } t \geq 0,$$

where f_F is the density of F. For a queue without customer abandonment, each patience time is assumed to be infinite and h is a zero function. In the diffusion model, the scaled abandonment process is approximated by

$$\tilde{G}(t) \approx \int_0^t \int_0^{(e'\tilde{Y}(s))^{+}} h\biggl(\frac{\sqrt{n}u}{\lambda}\biggr)\,\mathrm{d}u\,\mathrm{d}s \quad \text{for } t \geq 0. \tag{2.20}$$

The patience time distribution is built into this approximation through its hazard rate function. The intuition was explained in [27]: Consider the $Q(s)$ waiting customers in the buffer at time s. In general, only a small fraction of customers can abandon the system when the queue is working in the QED regime. Then by time s, the ith customer from the back of the queue has been waiting around i/λ time units. Approximately, this customer will abandon the queue during the next δ time units with probability $h(i/\lambda)\delta$. Then at time s, the instantaneous abandonment rate of the queue is around $\sum_{i=1}^{Q(s)} h(i/\lambda)$. By (2.11) and (2.15), the scaled abandonment rate can be approximated by

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{Q(s)} h\Big(\frac{i}{\lambda}\Big) \approx \int_0^{\tilde{Q}(s)} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u = \int_0^{(e'\tilde{Y}(s))^+} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u, \tag{2.21}$$

from which (2.20) follows. For a many-server queue in the QED regime, the arrival rate is on the order of $O(n)$ and the queue length is typically on the order of $O(n^{1/2})$. The patience time distribution in a neighborhood of zero is considered in the instantaneous abandonment rate in (2.21). This approximation can be justified for $GI/M/n+GI$ queues by Propositions 9.1 and 9.2 in [26]. These two propositions can be extended to $GI/Ph/n+GI$ queues with minor modifications to the proofs.

Using the Brownian replacement and the approximations in (2.16)–(2.20), we obtain the following stochastic differential equation

$$\begin{aligned} X(t) = X(0) - \beta\mu p t + pB_E(t) + B_0(\rho\mu t) + \sum_{j=1}^{d} B_j((\rho\wedge 1)\nu_j\gamma_j t) \\ - (I-P')B_S((\rho\wedge 1)\gamma t) - R\int_0^t (X(s) - p(e'X(s))^+)\,\mathrm{d}s \\ - p\int_0^t\int_0^{(e'X(s))^+} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u\,\mathrm{d}s, \end{aligned} \tag{2.22}$$

where the initial condition is taken to be $X(0) = \tilde{Y}(0)$. This stochastic differential equation is the *diffusion model* for the $GI/Ph/n+GI$ queue in the QED regime.

We may write (2.22) into the standard form

$$X(t) = X(0) + \int_0^t b(X(s))\,\mathrm{d}s + \int_0^t \sigma(X(s))\,\mathrm{d}B(s),$$

where for each $x \in \mathbb{R}^d$, the drift coefficient b is

$$b(x) = -\beta\mu p - R(x - p(e'x)^+) - p\int_0^{(e'x)^+} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u, \tag{2.23}$$

the diffusion coefficient σ is a $d \times d$ constant matrix satisfying

$$\begin{aligned}\Sigma(x) &= \sigma(x)\sigma'(x)\\ &= \rho\mu(c_a^2pp' + H^0)\\ &\quad + (\rho \wedge 1)\biggl(\sum_{j=1}^{d} \nu_j\gamma_j H^j + (I - P')\operatorname{diag}(\nu)\operatorname{diag}(\gamma)(I - P)\biggr),\end{aligned} \tag{2.24}$$

and B is a d-dimensional standard Brownian motion. One can check that $\Sigma(x)$ is positive definite and thus satisfies (2.3). Because h is bounded, both b and σ are Lipschitz continuous. Hence, a strong solution to (2.22) exists.

If the patience times are exponentially distributed with mean $1/\alpha$, the hazard rate function is constant with $h(t) = \alpha$ for all $t \geq 0$. In this case, the diffusion model becomes

$$\begin{aligned}\hat{X}(t) = \hat{X}(0) - \beta\mu pt + pB_E(t) + B_0(\rho\mu t) + \sum_{j=1}^{d} B_j((\rho \wedge 1)\nu_j\gamma_j t)\\ - (I - P')B_S((\rho \wedge 1)\gamma t) - R\int_0^t (\hat{X}(s) - p(e'\hat{X}(s))^+)\,\mathrm{d}s\\ - p\alpha\int_0^t (e'\hat{X}(s))^+\,\mathrm{d}s\end{aligned} \tag{2.25}$$

with $\hat{X}(0) = \tilde{Y}(0)$. When $\alpha = 0$, this stochastic differential equation approximates a queue without abandonment. In Section 3.2, $\hat{X} = \{\hat{X}(t) : t \geq 0\}$ will be used as an auxiliary process for choosing a reference density for the diffusion model.

When ρ is equal to 1, $\hat{X}$ is identical to the diffusion limit for $G/Ph/n+GI$ queues in Theorem 2 in [6]. This limit process is the strong solution of

$$\begin{aligned}\check{X}(t) = \check{X}(0) - \beta\mu pt + pB_E(t) + B_0(\mu t) + \sum_{j=1}^{d} B_j(\nu_j\gamma_j t)\\ - (I - P')B_S(\gamma t) - R\int_0^t (\check{X}(s) - p(e'\check{X}(s))^+)\,\mathrm{d}s\\ - p\alpha\int_0^t (e'\check{X}(s))^+\,\mathrm{d}s.\end{aligned} \tag{2.26}$$

More specifically, as the number of servers n goes large, the d-dimensional process $\tilde{Y}$ in (2.14) converges to $\check{X} = \{\check{X}(t) : t \geq 0\}$ in distribution under certain conditions. This diffusion limit allows for a general patience time distribution and α is not merely the rate of an exponential distribution. It is now defined by

$$\alpha = \lim_{t \downarrow 0} t^{-1} F(t), \tag{2.27}$$

which is the patience time density at zero. In Section 3.2, we will investigate the tail behavior of $\check{X}$ and use it to build a reference density for $\hat{X}$ in (2.25).

The above two processes, $\hat{X}$ in (2.25) and $\check{X}$ in (2.26), have the same drift coefficient

$$b(x) = -\beta\mu p - R(x - p(e'x)^+) - p\alpha(e'x)^+$$

that is a piecewise linear function of $x \in \mathbb{R}^d$. Both of them are d-dimensional *piecewise OU processes.*

3. A numerical algorithm for the stationary distribution.

In this section, we propose an algorithm for the stationary density of the diffusion model. The generic algorithm, introduced in Section 3.1, follows the one developed in [5]. The convergence of the algorithm is controlled by the reference density. In Section 3.2, we discuss how to choose a reference density by investigating the tail behavior of the diffusion model. The finite element implementation of the algorithm follows [30]. We leave it to the appendix.

3.1. *The generic algorithm.*

Consider the diffusion process X in (2.1). To compute its stationary density g on $\mathbb{R}^d$, we adopt a notion called the reference density that was first introduced in [5]. A *reference density* for g is a positive function r on $\mathbb{R}^d$ such that

$$\int_{\mathbb{R}^d} r(x)\,\mathrm{d}x < \infty \tag{3.1}$$

and

$$\int_{\mathbb{R}^d} q^2(x) r(x)\,\mathrm{d}x < \infty, \tag{3.2}$$

where

$$q(x) = \frac{g(x)}{r(x)} \quad \text{for each } x \in \mathbb{R}^d$$

is called the *ratio function*. Such a function r exists because g itself satisfies both (3.1) and (3.2). For the rest of Section 3.1, we assume that a reference density r has been determined and remains fixed, and that

$$\int_{\mathbb{R}^d} b_j^2(x) r(x)\,\mathrm{d}x < \infty \quad \text{and} \quad \int_{\mathbb{R}^d} \Sigma_{j\ell}^2(x) r(x)\,\mathrm{d}x < \infty \tag{3.3}$$

for $j, \ell = 1, \dots, d$. Since both b and σ are Lipschitz continuous, condition (3.3) is satisfied if

$$\int_{\mathbb{R}^d} |x|^4 \, r(x) \, \mathrm{d}x < \infty. \tag{3.4}$$

Let $L^2(\mathbb{R}^d, r)$ be the space of all square-integrable functions on $\mathbb{R}^d$ with respect to the measure that has density r, i.e.,

$$L^2(\mathbb{R}^d, r) = \left\{ f \in \mathcal{B}(\mathbb{R}^d) : \int_{\mathbb{R}^d} f^2(x) r(x) \, \mathrm{d}x < \infty \right\}$$

where $\mathcal{B}(\mathbb{R}^d)$ is the set of Borel-measurable functions on $\mathbb{R}^d$. We define an inner product on $L^2(\mathbb{R}^d, r)$ by

$$\langle f, \hat{f} \rangle = \int_{\mathbb{R}^d} f(x) \hat{f}(x) r(x) \, \mathrm{d}x \quad \text{for } f, \hat{f} \in L^2(\mathbb{R}^d, r).$$

With this inner product, $L^2(\mathbb{R}^d, r)$ is a Hilbert space and the induced norm is given by

$$\|f\| = \langle f, f \rangle^{1/2} \quad \text{for each } f \in L^2(\mathbb{R}^d, r). \tag{3.5}$$

Condition (3.2) is equivalent to $q \in L^2(\mathbb{R}^d, r)$ and assumption (3.3) ensures that $\mathcal{G}f \in L^2(\mathbb{R}^d, r)$ for all $f \in C_b^2(\mathbb{R}^d)$. In this space, the basic adjoint relationship in (2.7) is equivalent to

$$\langle \mathcal{G}f, q \rangle = 0 \quad \text{for all } f \in C_b^2(\mathbb{R}^d). \tag{3.6}$$

If we are able to obtain q by (3.6) with a fixed reference density, the stationary density can be computed via $g(x) = q(x) r(x)$ for $x \in \mathbb{R}^d$.

Let

$$H = \text{the closure of } \{\mathcal{G}f : f \in C_b^2(\mathbb{R}^d)\} \tag{3.7}$$

where the closure is taken in the norm in (3.5). As a subspace of $L^2(\mathbb{R}^d, r)$, H is orthogonal to q. Let c be a constant function with $c(x) = 1$ for all $x \in \mathbb{R}^d$. Clearly, $c \in L^2(\mathbb{R}^d, r)$ but $c \notin H$ because

$$\langle c, q \rangle = \int_{\mathbb{R}^d} g(x) \, \mathrm{d}x = 1. \tag{3.8}$$

Let

$$\bar{c} = \arg\min_{f \in H} \|c - f\| \tag{3.9}$$

be the projection of c onto H. Then, $c-\bar{c}$ must be orthogonal to H. Assuming that Conjecture 1 holds and that X has a unique stationary density, we have $q = \kappa_c(c-\bar{c})$ for some normalizing constant $\kappa_c \in \mathbb{R}$. By (3.8),

$$\kappa_c^{-1} = \langle c, c-\bar{c}\rangle = \langle c-\bar{c}, c-\bar{c}\rangle + \langle \bar{c}, c-\bar{c}\rangle = \|c-\bar{c}\|^2 .$$

Hence, the ratio function is

$$q = \frac{c-\bar{c}}{\|c-\bar{c}\|^2}. \tag{3.10}$$

To obtain q using (3.10), we need compute $\bar{c}$, the projection of c onto H. The space H is linear and infinite-dimensional (i.e., a basis of H contains infinitely many functions). In general, solving (3.9) in an infinite-dimensional space is impossible. In the algorithm, we use a finite-dimensional subspace H_k to approximate H.

Suppose that there is a sequence of finite-dimensional subspaces $\{H_k : k \in \mathbb{N}\}$ of H such that $H_k \to H$ in $L^2(\mathbb{R}^d, r)$ as $k \to \infty$. Here, $H_k \to H$ in $L^2(\mathbb{R}^d, r)$ means that for each $f \in H$, there exists a sequence of functions $\{\varphi_k : k \in \mathbb{N}\}$ with $\varphi_k \in H_k$ such that $\|\varphi_k - f\| \to 0$ as $k \to \infty$. Let

$$\bar{c}_k = \operatorname*{arg\,min}_{f \in H_k} \|c - f\| \tag{3.11}$$

be the projection of c onto H_k. By Proposition 7 in [5], we have the following approximation result.

Proposition 1. *Let r be a positive function on $\mathbb{R}^d$ that satisfies (3.1) and (3.2). Let $\{H_k : k \in \mathbb{N}\}$ be a sequence of finite-dimensional subspaces of H such that $H_k \to H$ in $L^2(\mathbb{R}^d, r)$ as $k \to \infty$. Let c be the constant function with $c(x) = 1$ for all $x \in \mathbb{R}^d$ and $\bar{c}_k$ be the projection of c on H_k given by (3.11). Assume that Conjecture 1 is true. Then,*

$$\|q_k - q\| \to 0 \quad \textit{as } k \to \infty,$$

where $q_k = (c-\bar{c}_k)/\|c-\bar{c}_k\|^2$. Moreover, if r is bounded on $\mathbb{R}^d$, then

$$\int_{\mathbb{R}^d} (g_k(x) - g(x))^2 \,\mathrm{d}x \to 0 \quad \textit{as } k \to \infty,$$

where $g_k(x) = q_k(x) r(x)$ for each $x \in \mathbb{R}^d$.

As in [5], we choose

$$H_k = \{\mathcal{G} f : f \in C_k\} \tag{3.12}$$

for some finite-dimensional space C_k. We will discuss how to construct C_k using a finite element method in the appendix. For notational convenience, we omit the subscript k when k is fixed. The finite-dimensional function space is thus denoted by C. Let m_C be the dimension of C and $\{f_i : i = 1, \dots, m_C\}$ be a basis of C. We assume that the family $\{\mathcal{G} f_i : i = 1, \dots, m_C\}$ is linearly independent in $L^2(\mathbb{R}^d, r)$. Then,

$$\bar{c}_k = \sum_{i=1}^{m_C} u_i \mathcal{G} f_i \quad \text{for some } u_i \in \mathbb{R} \text{ and } i = 1, \dots, m_C. \tag{3.13}$$

Using the fact $\langle \mathcal{G} f_i, c - \bar{c}_k \rangle = 0$ for $i = 1, \dots, m_C$, we obtain a system of linear equations

$$Au = v \tag{3.14}$$

where

$$A_{i\ell} = \langle \mathcal{G} f_i, \mathcal{G} f_\ell \rangle, \quad u = (u_1, \dots, u_{m_C})', \quad v_i = \langle \mathcal{G} f_i, c \rangle. \tag{3.15}$$

By the linear independence assumption, the $m_C \times m_C$ matrix A is positive definite. Thus, $u = A^{-1} v$ is the unique solution to (3.14). Once the vector u is obtained, we can compute the projection $\bar{c}_k$ by (3.13). Finally, the stationary density can be approximated via

$$g(x) \approx g_k(x) = r(x) \frac{c(x) - \bar{c}_k(x)}{\|c - \bar{c}_k\|^2} \quad \text{for each } x \in \mathbb{R}^d.$$

In [5], the authors employed multinominals of orders up to k to construct the space C_k. This choice appears to be numerically unstable. The approximation error is significant when k is small, say, $k \leq 5$. As k increases, the round-off error in solving (3.14) increases and ultimately dominates the approximation error. Although their implementation produces accurate estimates for the stationary means of SRBMs, it sometimes produces poor estimates for the stationary distributions. In this paper, we construct the space C_k using the finite element method as in [30]. This implementation yields more stable output. Please refer to the appendix.

3.2. *Choosing a reference density.* The reference density controls the convergence of the proposed algorithm. As long as the positive function r satisfies (3.1) and (3.2), the output of the proposed algorithm will converge to the stationary density (see Proposition 1 in Section 3.1 and Proposition 3 in the appendix). For choosing an appropriate reference density, some considerations are as follows.

First, to be a reference density, the candidate function r must satisfy (3.2) even though the stationary density g is unknown. This requires that r have a comparable or slower decay rate than g. When g is bounded, its decay rate is sufficient to determine a function r that satisfies (3.2).

Second, the most computational effort in our algorithm is constructing and solving the system of linear equations (3.14). By Proposition 3 in the appendix, the finite-dimensional space H_k approximates the infinite-dimensional space H better as k increases, thus reducing the approximation error. On the other hand, as the dimension of H_k increases, constructing and solving (3.14) requires more computation time and memory space. The condition number of the matrix A in (3.14) also gets worse as the dimension of H_k becomes large. This yields higher round-off error. A "good" reference density should balance these two types of error. With such a reference density, it is possible to have small approximation error even if the dimension of H_k is moderate.

Intuitively, when r is "close" to the stationary density g, both the ratio function q and the projection $\bar{c}$ are "close" to constant functions. We can thus expect that the space H_k with a moderate dimension is able to produce a satisfactory approximation. All these observations motivate us to explore the tail behavior of the diffusion model.

3.2.1. *Tail behavior of the limit queue length process.* Let us focus on the limiting tail behavior of the queue length process in a many-server queue. It will be used to estimate the tail of the diffusion model.

Consider a sequence of $GI/GI/n+GI$ queues in the QED regime. In each queue, the service times are iid following a general distribution. If all patience times are infinite, they are $GI/GI/n$ queues without customer abandonment. We assume that these queues, each indexed by the number of servers n, have the same service and patience time distributions. Let λ_n be the arrival rate of the nth system. To mathematically define the QED regime, we assume that

$$\lim_{n\to\infty} \frac{\lambda_n}{n} > 0 \tag{3.16}$$

and

$$\lim_{n\to\infty} \sqrt{n}(1-\rho_n) = \check{\beta} \quad \text{for some } \check{\beta}\in\mathbb{R}, \tag{3.17}$$

where $\rho_n = \lambda_n/(n\mu)$ is the traffic intensity of the nth system.

Assume that all these queues are in their steady states. Let $N_n(\infty)$ be the stationary number of customers in the nth system and

$$\tilde{N}_n(\infty) = \frac{1}{\sqrt{n}}(N_n(\infty) - n).$$

For $GI/GI/n$ queues in the QED regime, the limit queue length in the steady state was studied in [10] by Gamarnik and Momčilović, where the service time distribution is assumed to be lattice-valued on a finite support. The authors first showed that $\tilde{N}_n(\infty)$ converges to a random variable $\check{N}(\infty)$ in distribution as $n \to \infty$, and then proved that

$$\lim_{z\to\infty} \frac{1}{z} \log \mathbb{P}[\check{N}(\infty) > z] = -\frac{2\check{\beta}}{c_a^2 + c_s^2}, \tag{3.18}$$

where c_a^2 and c_s^2 are the squared coefficients of variation of the interarrival and service time distributions, respectively. In (3.18), the decay rate does not depend on the service time distribution beyond its first two moments. Recently, this result has been extended by Gamarnik and Goldberg in [9] to $GI/GI/n$ queues with a general service time distribution.

Assume that the piecewise OU process $\check{X}$ in (2.26) has a stationary distribution. Let $\check{X}(\infty)$ be the corresponding d-dimensional random vector in the steady state. When $\alpha = 0$ and $d = 1$, $\check{X}$ is the diffusion limit for $GI/M/n$ queues without customer abandonment. In this case, the service time distribution is exponential and $\check{N}(\infty) = \check{X}(\infty)$. It was proved in [13] that the stationary density of $\check{X}(\infty)$ has a closed-form expression

$$\check{g}(z) = \begin{cases} a_1 \exp\left(-\dfrac{(z+\check{\beta})^2}{1+c_a^2}\right) & \text{if } z < 0, \\ a_2 \exp\left(-\dfrac{2\check{\beta}z}{1+c_a^2}\right) & \text{if } z \geq 0, \end{cases} \tag{3.19}$$

where a_1 and a_2 are normalizing constants making $\check{g}$ continuous at zero. The decay rate of $\check{g}$ in (3.19) is consistent with (3.18). Both formulas suggest that $\check{N}(\infty)$ has an exponential tail on the right side.

For a $GI/GI/n + GI$ queue with many servers and customer abandonment, the limiting tail behavior of $\tilde{N}_n(\infty)$ remains unknown except for very simple cases. When $\alpha > 0$ and $d = 1$, the diffusion limit $\check{X}$ in (2.26) is a one-dimensional piecewise OU process. It admits a piecewise Gaussian stationary density

$$\check{g}(z) = \begin{cases} a_3 \exp\left(-\dfrac{(z+\check{\beta})^2}{1+c_a^2}\right) & \text{if } z < 0, \\ a_4 \exp\left(-\dfrac{\alpha(z+\alpha^{-1}\mu\check{\beta})^2}{\mu(1+c_a^2)}\right) & \text{if } z \geq 0, \end{cases} \tag{3.20}$$

where a_3 and a_4 are normalizing constants that make $\breve{g}$ continuous at zero. See [2]. In particular, it was proved in [11] that the stationary density of the diffusion limit of $M/M/n+M$ queues follows (3.20) with $c_a^2 = 1$. In contrast to the exponential tail in (3.18), the stationary density in (3.20) has a Gaussian tail on the right.

Observing (3.18) and (3.20), we conjecture that for $GI/GI/n+GI$ queues in the QED regime, the limiting tail behavior of $\tilde{N}_n(\infty)$ depends on the service time distribution only through its first two moments and on the patience time distribution only through its density at zero.

CONJECTURE 2. *Consider a sequence of $GI/GI/n+GI$ queues that satisfies (2.19), (3.16), and (3.17). Assume that the patience time distribution has a positive density at zero, i.e., $\alpha > 0$ in (2.27). Assume further that the interarrival and service time distributions satisfy the T_0 assumptions (i)–(iii) in Section 2.1 of [9]. Then, (a) $N_n(\infty)$ exists for each n; (b) the sequence of random variables $\{\tilde{N}_n(\infty) : n \in \mathbb{N}\}$ converges to a random variable $\check{N}(\infty)$ in distribution; (c) $\check{N}(\infty)$ satisfies*

$$\lim_{z\to\infty} \frac{1}{z^2} \log \mathbb{P}[\check{N}(\infty) > z] = -\frac{\alpha}{\mu(c_a^2 + c_s^2)}.$$

The intuition below may help understand why the conjectured decay rate is Gaussian. When $\check{N}(\infty) > z$ for some $z > 0$, there are more than $n^{1/2}z$ waiting customers in the associated queue, where each waiting customer is "racing" to abandon the system. At any time, the instantaneous abandonment rate is approximately proportional to the queue length. In such a system, the customer departure process, including both service completions and customer abandonments, behaves as if the system is a queue with infinite servers. It is known that in an infinite-server queue, the limit process for the scaled number of customers has a Gaussian stationary distribution. See, e.g., Theorem 4.1 in [15] for $M/M/\infty$ queues and the corollary of Theorem 3 in [31] for $GI/Ph/\infty$ queues. Thus, one can expect that the tail of the limit queue length for queues with abandonment is also Gaussian, which decays much faster than the exponential tail for queues without abandonment.

3.2.2. *A reference density for the piecewise OU process in (2.25).* Let us build a reference density for the piecewise OU process $\hat{X}$ in (2.25). It is the diffusion model for the $GI/Ph/n+M$ queue with an exponential patience time distribution. In Section 3.2.3, we will use an auxiliary $GI/Ph/n+M$ queue to build a reference density for the diffusion model (2.22).

In the QED regime, the traffic intensity ρ is close to 1. The tail behavior of $\hat{X}$ in (2.25) is thus expected to be comparable to that of the diffusion limit $\check{X}$ in (2.26). In the steady state, the diffusion limit $\check{X}$ satisfies

$$\check{N}(\infty) = e'\check{X}(\infty).$$

The discussion in Section 3.2.1 has given us ample evidence of the tail behavior of $\mathbb{P}[\check{N}(\infty) > z]$ as $z \to \infty$. Although the left tail $\mathbb{P}[\check{N}(\infty) < -z]$ as $z \to \infty$ remains unknown when $d > 1$, our numerical experiments suggest that this tail is not sensitive to the service time distribution beyond its mean. Thus, we use the left tail for a queue with an exponential service time distribution to construct the reference density. We propose to use a product reference density

$$r(x) = \prod_{j=1}^{d} r_j(x_j) \quad \text{for } x \in \mathbb{R}^d. \tag{3.21}$$

When $\alpha = 0$ and $\rho < 1$ in (2.25), there is no abandonment in the queue. Based on (3.18) and (3.19), we choose

$$r_j(z) = \begin{cases} \exp\left(-\dfrac{(z+\gamma_j\beta)^2}{1+c_a^2}\right) & \text{if } z < 0, \\ \exp\left(-\dfrac{2\beta z}{c_a^2+c_s^2} - \dfrac{\gamma_j^2\beta^2}{1+c_a^2}\right) & \text{if } z \geq 0, \end{cases} \tag{3.22}$$

where β is given by (2.8). The function r_j has an exponential tail on the right and a Gaussian tail on the left. The reference density given by (3.21) and (3.22) satisfies condition (3.4). In (3.22), we set the shift term for $z < 0$ to be $\gamma_j\beta$ according to the following observation. In the associated queue, β is the scaled mean number of idle servers and γ_j is the fraction of phase j service load. In the steady state, one can expect that $\tilde{Y}_j(t)$, the centered and scaled number of phase j customers, is around $-\gamma_j\beta$.

When $\alpha > 0$ in (2.25), the associated queue has abandonment. By (3.20) and Conjecture 2, we choose

$$r_j(z) = \begin{cases} \exp\left(-\dfrac{(z+\gamma_j\beta)^2}{1+c_a^2}\right) & \text{if } z < 0, \\ \exp\left(-\dfrac{\alpha(z+p_j\alpha^{-1}\mu\beta)^2}{\mu(c_a^2+c_s^2)} + \dfrac{p_j^2\alpha^{-1}\mu\beta^2}{c_a^2+c_s^2} - \dfrac{\gamma_j^2\beta^2}{1+c_a^2}\right) & \text{if } z \geq 0, \end{cases} \tag{3.23}$$

whose two tails are both Gaussian but have different decay rates. This reference density also satisfies (3.4). In (3.23), the shift term for $z \geq 0$ is taken to be $p_j\mu\beta/\alpha$ because of the observation below. When $\rho \geq 1$, the throughput of the queue is nearly $n\mu$. Let q_0 be the scaled queue length *in equilibrium*, i.e., the arrival and departure rates of the system are balanced when the queue length is around $n^{1/2}q_0$. Because in this case the abandonment rate is $\alpha n^{1/2}q_0$, we must have $\lambda = n\mu + \alpha n^{1/2}q_0$, or $q_0 = -\mu\beta/\alpha$ by (2.8). Since the fraction of phase j waiting customers is around p_j, $\tilde{Y}_j(t)$ is around $-p_j\mu\beta/\alpha$ as the queue reaches the steady state.

3.2.3. *A reference density for the diffusion model.* With a general patience time distribution, the tail behavior of X in (2.22) is unknown. In some cases, the diffusion limit in (2.26) can still help us find a reference density. The principle is again to ensure that the candidate function has a comparable or slower decay rate than the stationary density of X. For that, we build an auxiliary queue that shares the same arrival process and service times with the $GI/Ph/n + GI$ queue, but the auxiliary queue may have no abandonment or have an exponential patience time distribution. Let $\hat{X}$ be the diffusion process in (2.25) for the auxiliary queue. If $\hat{X}$ has a slower decay rate than X, a reference density of $\hat{X}$ must be a reference density of X, too.

When $\rho < 1$, the auxiliary queue is a $GI/Ph/n$ queue, so $\alpha = 0$ in (2.25). Intuitively, the queue length decays faster in the $GI/Ph/n + GI$ queue than in the auxiliary queue since the latter has no abandonment. As a consequence, $\hat{X}$ has a slower decay rate than X and the reference density given by (3.21) and (3.22) for $\hat{X}$ can be used for the current model.

When $\rho > 1$, the auxiliary queue is a $GI/Ph/n + M$ queue. Let $\alpha > 0$ be the rate of the exponential patience time distribution, which is to be determined in order for $\hat{X}$ to have an appropriate decay rate. For that, we need investigate the abandonment process of the $GI/Ph/n + GI$ queue.

Let $h^{(\ell)}$ be the ℓth order derivative of the hazard rate function h. Assume that h is m times continuously differentiable in a neighborhood of zero for some nonnegative integer m, and that among $\ell = 0, \dots, m$, there is at least one $h^{(\ell)}(0) \neq 0$. We follow the convention that $h^{(0)}(0) = h(0)$. Let ℓ_0 be the smallest nonnegative integer such that $h^{(\ell_0)}(0) \neq 0$. For $z > 0$ in a small neighborhood of zero, the ℓ_0th degree Taylor's approximation of h is

$$h(z) \approx \frac{h^{(\ell_0)}(0) z^{\ell_0}}{\ell_0!}, \tag{3.24}$$

which, along with (2.15) and (2.20), implies that the scaled abandonment process can be approximated by

$$\tilde{G}(t) \approx \frac{n^{\ell_0/2} h^{(\ell_0)}(0)}{\lambda^{\ell_0}(\ell_0 + 1)!} \int_0^t \tilde{Q}(s)^{\ell_0+1}\,\mathrm{d}s.$$

This approximation implies that the abandonment process depends on the hazard rate function primarily through $h^{(\ell_0)}(0)$, the nonzero derivative at the origin with the lowest order. It also implies that the scaled abandonment rate at time t is approximately

$$\int_0^{\tilde{Q}(t)} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u \approx \frac{n^{\ell_0/2}h^{(\ell_0)}(0)}{\lambda^{\ell_0}(\ell_0+1)!}\tilde{Q}(t)^{\ell_0+1}. \tag{3.25}$$

With a general hazard rate function, the scaled queue length in equilibrium q_0 satisfies

$$\lambda = n\mu + \sqrt{n}\int_0^{q_0} h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u. \tag{3.26}$$

If (3.25) holds, it turns out to be

$$\lambda \approx n\mu + \frac{n^{(\ell_0+1)/2}h^{(\ell_0)}(0)}{\lambda^{\ell_0}(\ell_0+1)!}q_0^{\ell_0+1},$$

which gives us

$$q_0 \approx \frac{1}{\sqrt{n}}\Big(\frac{\lambda^{\ell_0}(\ell_0+1)!(\lambda-n\mu)}{h^{(\ell_0)}(0)}\Big)^{1/(\ell_0+1)}. \tag{3.27}$$

The scaled queue length process fluctuates around this equilibrium length. Correspondingly, the instantaneous abandonment rate changes around an equilibrium level, too. This observation motivates us to take

$$\alpha = \frac{n^{\ell_0/2}h^{(\ell_0)}(0)}{\lambda^{\ell_0}(\ell_0+1)!}q_0^{\ell_0} \tag{3.28}$$

for the auxiliary $GI/Ph/n+M$ queue. With this setting, the original queue and the auxiliary queue have comparable abandonment rates when the scaled queue length is close to q_0. For any $q_1 > q_0$, when the scaled queue length is q_1 in both queues, the abandonment rate in the auxiliary queue is lower because

$$\alpha q_1 < \frac{n^{\ell_0/2}h^{(\ell_0)}(0)}{\lambda^{\ell_0}(\ell_0+1)!}q_1^{\ell_0+1}.$$

Hence, when the queue length is longer than q_0, it decays slower in the auxiliary queue than in the original queue. Consequently, the decay rate of $\hat{X}$ is slower than that of X and the reference density of $\hat{X}$ can work for the diffusion model.

The above discussion suggests a product reference density in (3.21) with

$$r_j(z) = \begin{cases} \exp\left(- \dfrac{(z+\gamma_j\beta)^2}{1+c_a^2} \right) & \text{if } z < 0, \\ \exp\left(- \dfrac{\alpha(z-p_jq_0)^2}{\mu(c_a^2+c_s^2)} + \dfrac{\alpha p_j^2 q_0^2}{\mu(c_a^2+c_s^2)} - \dfrac{\gamma_j^2\beta^2}{1+c_a^2} \right) & \text{if } z \geq 0, \end{cases} \tag{3.29}$$

where q_0 follows (3.27) and α follows (3.28).

The above reference density fails when $\rho = 1$ and $\ell_0 > 0$, because $q_0 = 0$ by (3.27) and thus α is zero in (3.28). In this case, we can still choose a reference density by (3.21) and (3.29) but using a traffic intensity ρ that is slightly larger than 1. Because the tail of the queue length becomes heavier as ρ increases, a reference density for the diffusion model with $\rho > 1$ must have a comparable or slower decay rate than the stationary density of the model with $\rho = 1$.

The reference density given by (3.21) and (3.29) that exploits the lowest-order nonzero derivative at the origin may fail when the hazard rate function has a rapid change near the origin. In this case, the Taylor's approximation in (3.24) may not be satisfactory when the queue length is not short enough. Such an example is discussed in Section 4.4. In addition, the above procedure cannot determine a reference density when all $h^{(\ell)}(0)$'s are zero, i.e., the hazard rate function is zero in a neighborhood of the origin. This issue will be explored in the future.

4. Numerical examples. Several numerical examples are presented in this section. In each example, we compute the stationary distribution of the number of customers in a many-server queue using the diffusion model and the proposed algorithm. We assume that the customer arrivals follow a Poisson process and the service times follow a two-phase hyperexponential distribution with mean 1, i.e., the system is an $M/H_2/n+GI$ queue with $c_a^2 = 1$ and $\mu = 1$. In such a queue, there are two types of customers. The service times of either type are iid following an exponential distribution, whereas the mean service times of these two types are different. We approximate this queue by a two-dimensional diffusion process X. The results computed using the diffusion model are compared with the results obtained either by the matrix-analytic method or by simulation. Please refer to [20] and [22] for the implementation of the matrix-analytic method. All simulation results are obtained by averaging 20 runs and in each run, the queue is simulated for 1.0×10^5 time units.

In the proposed algorithm, all numerical integration is implemented using a Gauss–Legendre quadrature rule. See [19]. When computing $A_{i\ell}$ or v_i in

(3.15), the integrand is evaluated at 8 points in each dimension. In the numerical examples, the tail probability

$$\mathbb{P}[X_1(\infty) + X_2(\infty) > z] = \int_{\{x\in\mathbb{R}^2 : x_1+x_2>z\}} g(x)\,\mathrm{d}x \quad \text{for some } z \in \mathbb{R} \tag{4.1}$$

is also evaluated, where $X(\infty) = (X_1(\infty), X_2(\infty))'$ is a two-dimensional random vector having probability density g. The integral in (4.1) is computed by adding up the integrals over the finite elements that intersect with the set $\{x \in \mathbb{R}^2 : x_1 + x_2 > z\}$, and the integral over each finite element is again computed using a Gauss–Legendre quadrature formula. Because the indicator function has jumps inside certain finite elements, we use 64 points in each dimension when evaluating the integrand over each finite element.

4.1. *Example 1: an $M/H_2/n + M$ queue.* Consider an $M/H_2/n + M$ queue that has an exponential patience time distribution. We are interested in such a queue because its customer-count process $N = \{N(t) : t \geq 0\}$ is a quasi-birth-death process, where $N(t)$ is the number of customers in system at time t. The stationary distribution of that can be computed by the matrix-analytic method.

In this example, we take $\alpha = 0.5$ for the rate of the exponential patience time distribution and take

$$p = (0.9351, 0.0649)' \quad \text{and} \quad \nu = (9.354, 0.072)'$$

for the hyperexponential service time distribution. The mean service time of the second-type customers is more than 100 times longer than that of the first type. Although over 93% of customers are of the first type, the fraction of its workload is merely 10%, i.e., $\gamma = (0.1, 0.9)'$. Such a distribution has a large squared coefficient of variation $c_s^2 = 24$.

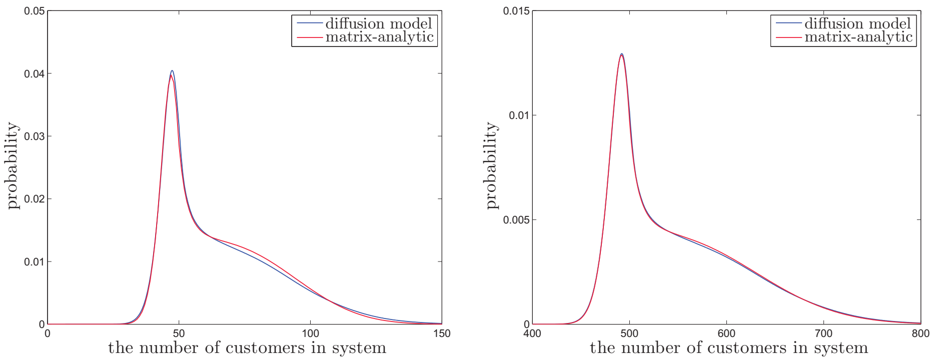
Because $h(t) = \alpha$ for all $t \geq 0$, X in (2.22) is a two-dimensional piecewise OU process. Because the service time distribution is hyperexponential, P is a zero matrix and thus $R = \mathrm{diag}(\nu)$. By (2.23) and (2.24), the drift coefficient of X is

$$b(x) = \begin{pmatrix} -p_1\mu\beta - \nu_1(x_1 - p_1(x_1+x_2)^+) - p_1\alpha(x_1+x_2)^+ \\ -p_2\mu\beta - \nu_2(x_2 - p_2(x_1+x_2)^+) - p_2\alpha(x_1+x_2)^+ \end{pmatrix} \tag{4.2}$$

and the covariance matrix of the diffusion coefficient is

$$\Sigma(x) = \begin{pmatrix} p_1\mu(\rho + (\rho \wedge 1)) & 0 \\ 0 & p_2\mu(\rho + (\rho \wedge 1)) \end{pmatrix} \tag{4.3}$$

for all $x \in \mathbb{R}^2$.



(A) $\rho = 1.141$ *and* $n = 50$

(B) $\rho = 1.045$ *and* $n = 500$

FIG 1: *The stationary distribution of the customer number in the* $M/H_2/n+M$ *queue, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method.*

We consider three scenarios, in all of which the queue is overloaded. In the first two scenarios, there are $n = 50$ and 500 servers, respectively. The arrival rates are $\lambda = 57.071$ and 522.36, or equivalently, $\rho = 1.141$ and 1.045. By (2.8), $\beta = -1$ in both scenarios. The third scenario, with $n = 20$ servers, will be presented shortly.

To compute the stationary distribution of X, we use a product reference density given by (3.21) and (3.23). To generate basis functions by the finite element method, we set the truncation rectangle $K = [-7, 32] \times [-7, 32]$, which is obtained by (A.1) with $\varepsilon_0 = 10^{-7}$, and use a lattice mesh in which all finite elements are 0.5×0.5 squares.

Once the stationary density of X is obtained, one can approximately produce the distribution of $N(\infty)$, the stationary number of customers in system. Note that the probability density of $X_1(\infty) + X_2(\infty)$ is given by

$$g_N(z) = \int_{-\infty}^{+\infty} g(x_1, z - x_1)\,\mathrm{d}x_1 \quad \text{for } z \in \mathbb{R}.$$

The distribution of $N(\infty)$ can be approximated by

$$\mathbb{P}[N(\infty) = i] \approx \frac{1}{\sqrt{n}} g_N\Big(\frac{i-n}{\sqrt{n}}\Big) \quad \text{for } i = 0, 1, \ldots.$$

For the first two scenarios, the distributions of $N(\infty)$ obtained by the diffusion model are illustrated in Figure 1. In the same figure, the stationary distributions computed by the matrix-analytic method are plotted, too.

We see good agreement. Comparing the two scenarios, we also find out that the diffusion model is more accurate when the number of servers n is larger. This observation is consistent with the many-server limit theorem for $G/Ph/n+GI$ queues in [6].

The matrix-analytic method can be used in this example because the three-dimensional process $\{(Q(t), Z_1(t), Z_2(t)) : t \geq 0\}$ forms a continuous-time Markov chain and the customer-count process N is a quasi-birth-death process. Clearly, $N(t) = Q(t) + Z_1(t) + Z_2(t)$. At time t, N is said to be at level ℓ if $N(t) = \ell$. In this example, level ℓ consists of $\ell + 1$ states if $\ell \leq n$ and it contains $n+1$ states if $\ell > n$. In the matrix-analytic method, the transition rate matrices between adjacent levels are used to compute the stationary distribution of N iteratively. Each iteration requires $O(n^3)$ arithmetic operations. For this queue, the transition rate matrices at different levels are different because the abandonment rate depends on the queue length. For implementation purposes, we assume in the algorithm that at level $\ell > \ell_0$ for some $\ell_0 \gg n$, the abandonment rate at level ℓ is $\alpha(\ell_0 - n)$ rather than $\alpha(\ell - n)$. In other words, the transition rate matrices at level ℓ are invariant with respect to ℓ when $\ell > \ell_0$. We take $\ell_0 = n + 2000$ in all numerical examples. The extra error caused by this modification is negligible, because in this queue, the queue length is on the order of $O(n^{1/2})$ and the chance of the customer number exceeding ℓ_0 is extremely rare.

To investigate the diffusion model quantitatively, we list some steady-state performance measures in Table 1. They include the mean queue length, the fraction of abandoned customers, and the probabilities that the number of customers exceeds certain levels. Using the diffusion model,

$$\text{the mean queue length} \approx \sqrt{n} \int_{\mathbb{R}^2} (x_1 + x_2)^+ g(x)\,\mathrm{d}x$$

and

$$\text{the mean number of idle servers} \approx \sqrt{n} \int_{\mathbb{R}^2} (x_1 + x_2)^- g(x)\,\mathrm{d}x.$$

It follows from the latter approximation that

$$\text{the abandonment fraction} \approx 1 - \frac{\mu}{\lambda}\Big(n - \sqrt{n} \int_{\mathbb{R}^2} (x_1 + x_2)^- g(x)\,\mathrm{d}x\Big).$$

In the table, the tail probability $\mathbb{P}[N(\infty) > \ell]$ is approximated by

$$\mathbb{P}[N(\infty) > \ell] \approx \mathbb{P}\Big[X_1(\infty) + X_2(\infty) > \frac{1}{\sqrt{n}}(\ell - n)\Big] \quad \text{for } \ell = 0, 1, \ldots.$$

Table 1

Performance measures of the $M/H_2/n+M$ queue, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method

(A) $\rho = 1.141$ *and* $n = 50$

	Diffusion	Matrix-analytic
Mean queue length	17.27	17.16
Abandonment fraction	0.1512	0.1503
$\mathbb{P}[N(\infty) > 45]$	0.8675	0.8523
$\mathbb{P}[N(\infty) > 50]$	0.6785	0.6726
$\mathbb{P}[N(\infty) > 100]$	0.08700	0.07436
$\mathbb{P}[N(\infty) > 130]$	0.008662	0.003299

(B) $\rho = 1.045$ *and* $n = 500$

	Diffusion	Matrix-analytic
Mean queue length	54.17	54.05
Abandonment fraction	0.05181	0.05173
$\mathbb{P}[N(\infty) > 470]$	0.9701	0.9694
$\mathbb{P}[N(\infty) > 500]$	0.6838	0.6818
$\mathbb{P}[N(\infty) > 600]$	0.2244	0.2229
$\mathbb{P}[N(\infty) > 750]$	0.008233	0.006395

and $\mathbb{P}[X_1(\infty) + X_2(\infty) > (\ell - n)/\sqrt{n}]$ is computed via (4.1). In both scenarios, the diffusion model produces satisfactory numerical estimates.

The computational complexity of the proposed algorithm, whether in computation time or in memory space, does not change with the number of servers n. In contrast, the matrix-analytic method becomes computationally expensive when n is large. In particular, the memory usage becomes a serious constraint when a huge number of iterations are required. For the $n = 500$ scenario in this example, it took around 1 hour to finish the matrix-analytic computation and the peak memory usage is nearly 5 GB. Using the diffusion model and the proposed algorithm, it took less than 1 minute and the peak memory usage is less than 200 MB on the same computer. See Section 5.4 for more discussion on the computational complexity.

Although the diffusion model is motivated and derived from the theory of many-server queues, it is still relevant for a queue with a modest number of servers. In the third scenario, there are $n = 20$ servers and the arrival rate is $\lambda = 22.24$. Thus, $\rho = 1.112$ and $\beta = -0.5$. In the proposed algorithm, we keep the same truncation rectangle and lattice mesh as in the previous two scenarios, and the reference density is again from (3.21) and (3.23). As illustrated in Figure 2, the diffusion model can still capture the exact stationary distribution for a queue with as few as twenty servers.

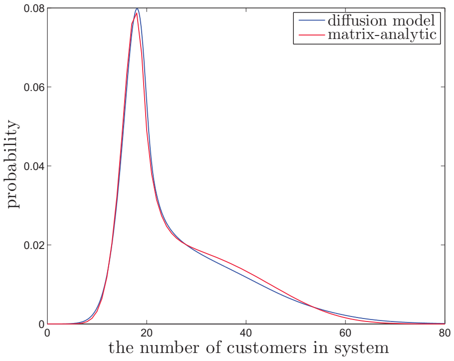


FIG 2: *The stationary distribution of the customer number in the $M/H_2/n+M$ queue, with $\rho = 1.112$ and $n = 20$, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method.*

4.2. *Example 2: an $M/H_2/n$ queue.* In this example, an $M/H_2/n$ queue without abandonment is considered. The hyperexponential service time distribution has

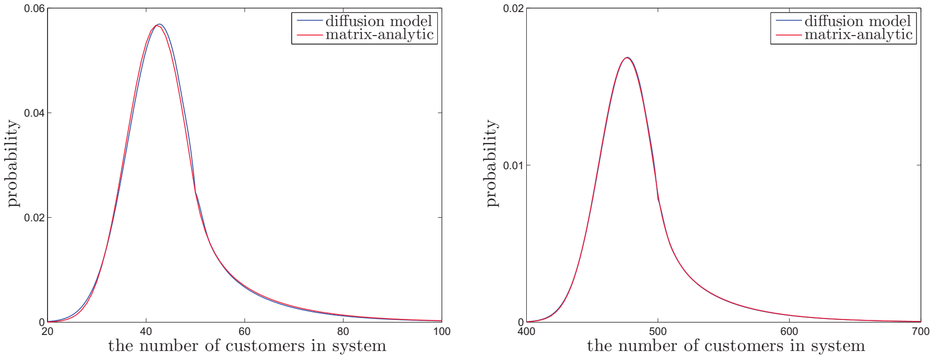
$$p = (0.5915, 0.4085)' \quad \text{and} \quad \nu = (5.917, 0.454)'.$$

Thus, $c_s^2 = 3$ and $\gamma = (0.1, 0.9)'$. Because there is no abandonment, we must take $\rho < 1$ in order for the system to reach the steady state.

In the diffusion model (2.22), the hazard rate function h is constantly zero. The drift and diffusion coefficients of X are given by (4.2) and (4.3) with $\alpha = 0$. The first scenario has $n = 50$ servers and the second scenario has $n = 500$ servers. The respective arrival rates are $\lambda = 42.929$ and 477.64. Hence, $\rho = 0.8586$ and 0.9553, both yielding $\beta = 1$. The product reference density is given by (3.21) and (3.22). With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is set by (A.1) to be $K = [-7, 35] \times [-7, 35]$, which is divided into 0.5×0.5 finite elements.

The stationary distribution of the number of customers in system is shown in Figure 3. In both scenarios, the diffusion model produces a good approximation of the result by the matrix-analytic method. As in the previous example, the diffusion model is more accurate when the system scale is larger. Several performance measures in the steady state are listed in Table 2. As in Table 1, satisfactory agreement can be found between the two approaches.

The third scenario has $n = 20$ servers with arrival rate $\lambda = 17.76$. Then, $\rho = 0.8882$ and $\beta = 0.5$. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is taken to be $K = [-7, 79] \times [-7, 79]$. The lattice mesh consists of 0.5×0.5 finite elements. The distribution of $N(\infty)$ is shown in Figure 4. For a queue without



(A) $\rho = 0.8586$ *and* $n = 50$

(B) $\rho = 0.9553$ *and* $n = 500$

FIG 3: *The stationary distribution of the customer number in the $M/H_2/n$ queue, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.22), and (ii) by the matrix-analytic method.*

TABLE 2
Performance measures of the $M/H_2/n$ queue, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.22), and (ii) by the matrix-analytic method

(A) $\rho = 0.8586$ *and* $n = 50$

	Diffusion	Matrix-analytic
Mean queue length	2.267	2.419
$\mathbb{P}[N(\infty) > 40]$	0.6908	0.6578
$\mathbb{P}[N(\infty) > 50]$	0.2072	0.2012
$\mathbb{P}[N(\infty) > 70]$	0.03395	0.03655
$\mathbb{P}[N(\infty) > 100]$	0.003537	0.003494

(B) $\rho = 0.9553$ *and* $n = 500$

	Diffusion	Matrix-analytic
Mean queue length	8.753	8.800
$\mathbb{P}[N(\infty) > 450]$	0.9038	0.9005
$\mathbb{P}[N(\infty) > 500]$	0.2285	0.2263
$\mathbb{P}[N(\infty) > 600]$	0.01910	0.01908
$\mathbb{P}[N(\infty) > 700]$	0.002241	0.001903

abandonment, the diffusion model is still useful when the number of servers is modest.

4.3. *Example 3: an $M/H_2/n + E_k$ queue.* Consider an $M/H_2/n + E_k$ queue, where $k > 1$ is a positive integer and $+E_k$ signifies an Erlang-k patience time distribution. In this queue, each patience time is the sum of k

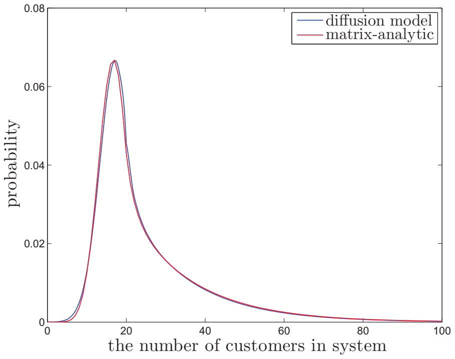


FIG 4: *The stationary distribution of the customer number in the $M/H_2/n$ queue, with $\rho = 0.8882$ and $n = 20$, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.22), and (ii) by the matrix-analytic method.*

stages and the stages are iid having an exponential distribution with mean $1/\theta$. In the following numerical experiments, we take $k = 2$ or 3 for the Erlang-k distribution and set $\theta = k$. As a result, the mean patience time is 1. The hyperexponential service time distribution is taken to be identical to that in Section 4.2.

The hazard rate function of the Erlang-k distribution is

$$h(t) = \frac{\theta^k t^{k-1}}{(k-1)! \sum_{\ell=0}^{k-1} \frac{\theta^\ell t^\ell}{\ell!}} \quad \text{for } t \geq 0.$$

For the diffusion model, it follows from (2.23) that the drift coefficient is

$$b(x) = \begin{pmatrix} -p_1\mu\beta - \nu_1(x_1 - p_1(x_1+x_2)^+) - p_1\eta(x_1+x_2)^+ \\ -p_2\mu\beta - \nu_2(x_2 - p_2(x_1+x_2)^+) - p_2\eta(x_1+x_2)^+ \end{pmatrix} \tag{4.4}$$

where

$$\eta(z) = \int_0^z h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u = \theta z - \frac{\lambda}{\sqrt{n}} \log\left(\sum_{m=0}^{k-1} \frac{n^{m/2}\theta^m z^m}{m!\lambda^m}\right) \quad \text{for } z \geq 0.$$

The first two scenarios has $n = 50$ and 500 servers, respectively. Their respective arrival rates are $\lambda = 42.929$ and 477.64. Hence, $\rho = 0.8586$ and 0.9553, both leading to $\beta = 1$. In the proposed algorithm, the reference density is chosen according to (3.21) and (3.22). The truncation rectangle is taken to be $K = [-7, 35] \times [-7, 35]$ and is divided into 0.5×0.5 finite elements. Some performance estimates can be found in Table 3.

TABLE 3

Performance measures of the $M/H_2/n + E_k$ queue with $\rho < 1$ and $k = 2$ or 3, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.22), and (ii) by simulation

(A) $\rho = 0.8586$ *and* $n = 50$

	$+E_2$		$+E_3$	
	Diffusion	Simulation	Diffusion	Simulation
Mean queue length	0.9820	1.061	1.201	1.302
Abandonment fraction	0.007974	0.008592	0.005629	0.006115
$\mathbb{P}[N(\infty) > 35]$	0.8881	0.8745	0.8896	0.8762
$\mathbb{P}[N(\infty) > 40]$	0.6755	0.6399	0.6798	0.6448
$\mathbb{P}[N(\infty) > 50]$	0.1671	0.1581	0.1788	0.1707
$\mathbb{P}[N(\infty) > 60]$	0.03238	0.03353	0.04420	0.04584

(B) $\rho = 0.9553$ *and* $n = 500$

	$+E_2$		$+E_3$	
	Diffusion	Simulation	Diffusion	Simulation
Mean queue length	4.960	5.048	6.455	6.569
Abandonment fraction	0.001689	0.001729	0.0007611	0.0007931
$\mathbb{P}[N(\infty) > 450]$	0.9003	0.8964	0.9022	0.8984
$\mathbb{P}[N(\infty) > 480]$	0.4759	0.4643	0.4859	0.4746
$\mathbb{P}[N(\infty) > 500]$	0.1995	0.1966	0.2151	0.2124
$\mathbb{P}[N(\infty) > 550]$	0.02798	0.02841	0.04412	0.04458

The third and fourth scenarios are for the case $\rho > 1$. They have $n = 50$ and 500 servers, and arrival rates $\lambda = 57.071$ and 522.36, respectively. Then, $\rho = 1.141$ and 1.045, both having $\beta = -1$. For these two scenarios, we adopt the reference density in (3.21) and (3.29). When $k = 2$, each patience time has two stages. The hazard rate function of the patience time distribution has $h(0) = 0$ and $h^{(1)}(0) = \theta^2$, so $\ell_0 = 1$ in (3.27) and (3.28). Because α in (3.28) depends on n, both the reference density and the truncation rectangle change with n. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is set to be $K = [-7, 13] \times [-7, 13]$ for $n = 50$ and to be $K = [-7, 16] \times [-7, 16]$ for $n = 500$. When $k = 3$, a patience time consists of three stages. In this case, $h(0) = h^{(1)}(0) = 0$ and $h^{(2)}(0) = 8\theta^3$, so $\ell_0 = 2$. We set $K = [-7, 11] \times [-7, 11]$ for $n = 50$ and $K = [-7, 15] \times [-7, 15]$ for $n = 500$. All truncation rectangles are partitioned into 0.5×0.5 finite elements. The performance estimates are listed in Table 4.

To evaluate the diffusion model, we list corresponding simulation estimates of the performance measures in both tables. As in the previous examples, the diffusion model produces adequate approximations.

It seems that the matrix-analytic method can be used in this example as the customer-count process is again a quasi-birth-death process. But this is

TABLE 4

Performance measures of the $M/H_2/n+E_k$ queue with $\rho > 1$ and $k = 2$ or 3, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.29), and (ii) by simulation

(A) $\rho = 1.141$ *and* $n = 50$

	$+E_2$		$+E_3$	
	Diffusion	Simulation	Diffusion	Simulation
Mean queue length	15.03	14.94	19.44	19.31
Abandonment fraction	0.1332	0.1334	0.1303	0.1305
$\mathbb{P}[N(\infty) > 45]$	0.9568	0.9490	0.9704	0.9645
$\mathbb{P}[N(\infty) > 50]$	0.8780	0.8648	0.9169	0.9066
$\mathbb{P}[N(\infty) > 70]$	0.3325	0.3121	0.5037	0.4761
$\mathbb{P}[N(\infty) > 90]$	0.008153	0.009354	0.03033	0.03422

(B) $\rho = 1.045$ *and* $n = 500$

	$+E_2$		$+E_3$	
	Diffusion	Simulation	Diffusion	Simulation
Mean queue length	76.50	76.20	119.5	119.1
Abandonment fraction	0.04438	0.04437	0.04340	0.04337
$\mathbb{P}[N(\infty) > 480]$	0.9857	0.9846	0.9946	0.9940
$\mathbb{P}[N(\infty) > 500]$	0.9390	0.9363	0.9770	0.9756
$\mathbb{P}[N(\infty) > 600]$	0.3115	0.3051	0.6733	0.6645
$\mathbb{P}[N(\infty) > 700]$	0.0009757	0.0009658	0.04260	0.04358

impractical because the computational complexity is too high. Consider the case that the patience time distribution is Erlang-2. Let $V_1(t)$ and $V_2(t)$ be the respective numbers of waiting customers whose patience times are in the first and second stages at time t. For this queue, the four-dimensional process $\{(V_1(t), V_2(t), Z_1(t), Z_2(t)) : t \geq 0\}$ is a continuous-time Markov chain. At level ℓ, there are $\ell+1$ states if $\ell \leq n$ and there are $(n+1)(\ell-n+1)$ states if $\ell > n$. The number of states at level ℓ is formidable when ℓ is large. Even if we may truncate the state space using the technique described in Section 4.1, the number of states is still too large to apply the matrix-analytic method. In fact, except simulation and the diffusion model, we are not aware of any numerical methods that are able to produce the estimates in Tables 3 and 4.

4.4. *Example 4: an $M/H_2/n+H_2$ queue.* In this example, we consider a patience time distribution that changes rapidly near the origin. Assume that the patience times follow a two-phase hyperexponential distribution with

$$\hat{p} = (0.9, 0.1)' \quad \text{and} \quad \hat{\nu} = (1, 200)'.$$

Hence, there are two types of patience times: 90% of them are exponentially distributed with mean 1 and 10% are exponentially distributed with mean 0.005. We take the same service time distribution as in Sections 4.2 and 4.3.

The hazard rate function of the hyperexponential patience time distribution is

$$h(t) = \frac{\hat{p}_1\hat{\nu}_1 \exp(-\hat{\nu}_1 t) + \hat{p}_2\hat{\nu}_2 \exp(-\hat{\nu}_2 t)}{\hat{p}_1 \exp(-\hat{\nu}_1 t) + \hat{p}_2 \exp(-\hat{\nu}_2 t)} \quad \text{for } t \geq 0.$$

The drift coefficient of X in (2.22) also follows (4.4) with

$$\begin{aligned}\eta(z) &= \int_0^z h\Big(\frac{\sqrt{n}u}{\lambda}\Big)\,\mathrm{d}u \\ &= -\frac{\lambda}{\sqrt{n}} \log\Big(\hat{p}_1 \exp\Big(-\frac{\sqrt{n}}{\lambda}\hat{\nu}_1 z\Big) + \hat{p}_2 \exp\Big(-\frac{\sqrt{n}}{\lambda}\hat{\nu}_2 z\Big)\Big) \quad \text{for } z \geq 0.\end{aligned}$$

In this example, we have

$$h(0) = \hat{p}_1\hat{\nu}_1 + \hat{p}_2\hat{\nu}_2 = 20.9 \quad \text{and} \quad h^{(1)}(0) = -\hat{p}_1\hat{p}_2(\hat{\nu}_1 - \hat{\nu}_2)^2 = -3564.1.$$

Thus, $\ell_0 = 0$ and h has a steep slope near the origin. As the zeroth degree Taylor's approximation in (3.24) may bring on too much error, the reference density exploiting the lowest-order nonzero derivative at the origin could be erroneous.

To choose an appropriate reference density, an auxiliary queue is used again. As in Section 3.2.3, the auxiliary queue is an $M/H_2/n + M$ queue that shares the same arrival process and service times with the $M/H_2/n+H_2$ queue. Let $\alpha > 0$ be the rate of the exponential patience time distribution. We take $\alpha = \hat{\nu}_1 \wedge \hat{\nu}_2$ so that the patience times in the auxiliary queue all belong to the type with the longer mean. If the queue lengths are equal, the abandonment rate in the auxiliary queue must be lower than that in the original queue. Therefore, the queue length in the former decays slower. A reference density designed for the auxiliary queue should also work for the original queue. This observation leads to a reference density that follows (3.21) and (3.29), but in this example, we take $\alpha = \hat{\nu}_1 \wedge \hat{\nu}_2$ and solve (3.26) to find q_0.

Two scenarios with $n = 50$ and 500 servers are investigated. The respective arrival rates are $\lambda = 57.071$ and 522.36. Thus, $\rho = 1.141$ and 1.045 and both scenarios have $\beta = -1$. By solving (3.26), we have $q_0 = 0.165$ for the first scenario and $q_0 = 0.0059$ for the second scenario. The reference density follows (3.21) and (3.29) with $\alpha = \hat{\nu}_1 = 1$. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is $K = [-7, 9] \times [-7, 9]$, partitioned into 0.5×0.5 finite elements. The performance estimates obtained by the diffusion model are compared with the simulation results in Table 5. As in the previous examples, the diffusion model is still accurate.

TABLE 5

Performance measures of the $M/H_2/n+H_2$ queue, computed (i) by the finite element algorithm for the diffusion model with the reference density in (3.21) and (3.29), and (ii) by simulation

(A) $\rho = 1.141$ *and* $n = 50$

	Diffusion	Simulation
Mean queue length	4.869	4.845
Abandonment fraction	0.1504	0.1499
$\mathbb{P}[N(\infty) > 40]$	0.9749	0.9728
$\mathbb{P}[N(\infty) > 50]$	0.6377	0.6111
$\mathbb{P}[N(\infty) > 60]$	0.1895	0.1737
$\mathbb{P}[N(\infty) > 70]$	0.02568	0.02142

(B) $\rho = 1.045$ *and* $n = 500$

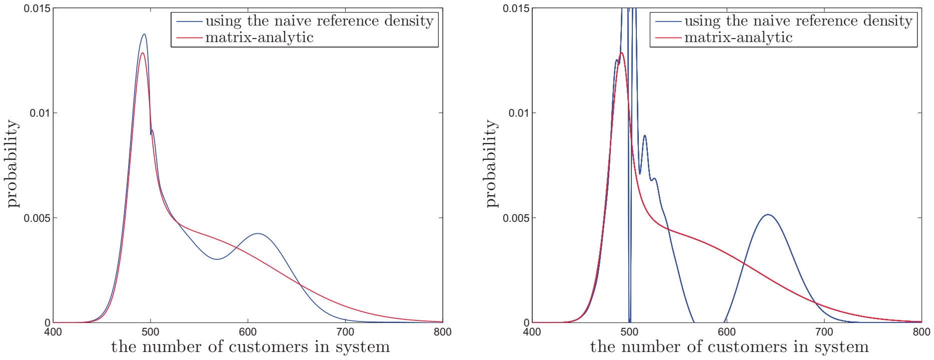
	Diffusion	Simulation
Mean queue length	6.359	6.413
Abandonment fraction	0.05517	0.05512
$\mathbb{P}[N(\infty) > 480]$	0.8929	0.8881
$\mathbb{P}[N(\infty) > 500]$	0.4822	0.4720
$\mathbb{P}[N(\infty) > 520]$	0.1074	0.1050
$\mathbb{P}[N(\infty) > 550]$	0.006616	0.006248

5. Implementation issues.

In this section, we discuss several practical issues arising from the algorithm implementation. As an illustration, the scenario of the $M/H_2/n+M$ queue with $n = 500$ servers in Section 4.1 is investigated throughout Sections 5.1–5.4. The influence of the reference density on algorithm output, the mesh and quadrature order selection, as well as the computational complexity of the proposed algorithm, are studied there. An $M/M/n+M$ queue is considered in Section 5.5. In this example, by comparing the algorithm output with the exact performance measures of the queue and the analytical results of the diffusion model, we show that the error in the performance estimates is mostly from the approximate model. In other words, the error caused by the finite element algorithm is usually negligible.

5.1. *Influence of the reference density.*

The reference density controls the convergence of the algorithm. It must satisfy both (3.1) and (3.2). If condition (3.1) does not hold, the sequence of subspaces $\{H_k : k \in \mathbb{N}\}$ generated by the finite element method may not converge to H in $L^2(\mathbb{R}^d, r)$ (see Proposition 3 in the appendix). Without condition (3.2), the ratio function q may not be in $L^2(\mathbb{R}^d, r)$. In either case, the output of the algorithm may significantly deviate from the exact stationary density. To demonstrate this issue, let us consider a "naive" reference density.



(A) $K = [-7, 10] \times [-7, 10]$

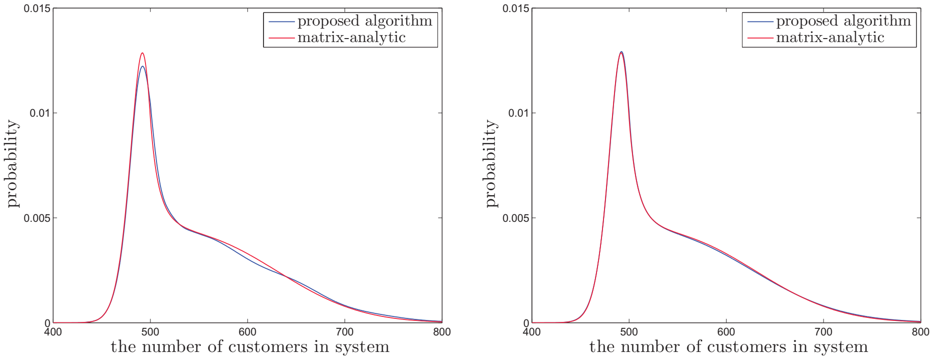
(B) $K = [-7, 32] \times [-7, 32]$

FIG 5: *The stationary distribution of the customer number in the $M/H_2/n + M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the finite element algorithm with the* "naive" reference density*, and (ii) by the matrix-analytic method.*

To produce the "naive" reference density, we consider a queue that has the same arrival process and patience time distribution as the $M/H_2/n+M$ queue. This new queue has an exponential service time distribution and its mean service time is equal to that of the $M/H_2/n + M$ queue. For this $M/M/n + M$ queue, the diffusion model is a one-dimensional piecewise OU process whose stationary density follows (3.20). The "naive" reference density is a product reference density in (3.21) with each r_j being the stationary density in (3.20). In other words, the "naive" reference density is obtained by pretending the service time distribution to be exponential.

Let us apply the "naive" reference density to the finite element algorithm. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is set to be $K = [-7, 10] \times [-7, 10]$ and is partitioned into 0.5×0.5 finite elements. As shown in Figure 5a, the output of the proposed algorithm noticeably deviates from the exact stationary distribution. This is in sharp contrast to the algorithm output in Figure 1b, which uses the proposed reference density given by (3.21) and (3.23) and produces a perfect agreement with the exact results. To further confirm that the "naive" reference density cannot work, we also test the truncation rectangle $K = [-7, 32] \times [-7, 32]$, which is used in Section 4.1 along with the proposed reference density. In this case, the matrix A in (3.14) is close to singular and its condition number is 3.52×10^{190}. Figure 5b manifests severe error in the algorithm output.

The hyperexponential service time distribution of the queue has $c_s^2 = 24$. Comparing (3.20) with (3.23), we can tell that the decay rate of the "naive"



(A) 1.0 × 1.0 *finite elements* (B) 0.25 × 0.25 *finite elements*

FIG 6: *The stationary distribution of the customer number in the $M/H_2/n + M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the finite element algorithm using* different meshes *with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method.*

reference density is much larger than that of the proposed reference density. If Conjecture 2 is true, one can expect that the "naive" reference density decays much faster than the stationary density and condition (3.2) does not hold. In this case, the ratio function q is not in $L^2(\mathbb{R}^d, r)$. As a consequence, the algorithm fails to produce any adequate estimate of the ratio function.

5.2. *Mesh selection.* When both the reference density and the truncation hypercube are fixed, using a finer mesh may produce smaller approximation error. However, a finer mesh yields more basis functions, which in turn lead to a larger condition number for the matrix A in (3.14). If the condition number of A is too large, the round-off error in solving (3.14) becomes considerable. So a finer mesh does not necessarily yield a more accurate output.

Let us test different meshes for the second scenario in Section 4.1. We keep the same settings for the algorithm except the size of finite elements. The output with 1.0 × 1.0 finite elements is plotted in Figure 6a. With this mesh, the algorithm does not perform well at the interval where the stationary distribution has a rapid change. We need a finer mesh to improve the accuracy. In this case, the condition number of A is 5.70×10^{20}. Recall that to produce the curve in Figure 1b, we use a mesh consisting of 0.5×0.5 finite elements. With this mesh, the condition number of A is 1.15×10^{23}. When the element size is further reduced to 0.25×0.25, the condition number of A grows to 7.13×10^{27}. As illustrated in Figure 6b, the output of

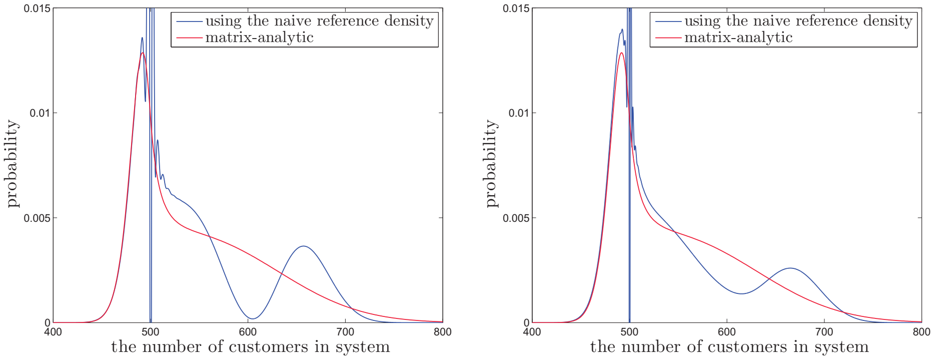
Table 6
Performance measures of the $M/H_2/n+M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the finite element algorithm using different meshes *with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method*

	0.5×0.5	0.25×0.25	Matrix-analytic
Mean queue length	54.17	54.17	54.05
Abandonment fraction	0.05181	0.05182	0.05173
$\mathbb{P}[N(\infty) > 470]$	0.9701	0.9702	0.9694
$\mathbb{P}[N(\infty) > 500]$	0.6838	0.6835	0.6818
$\mathbb{P}[N(\infty) > 600]$	0.2244	0.2241	0.2229
$\mathbb{P}[N(\infty) > 750]$	0.008233	0.008246	0.006395

the algorithm fits the exact stationary distribution well. When we compare Figures 1b and 6b, however, there is barely any noticeable difference between the algorithm outputs. To confirm that this mesh is not superior to the one with 0.5×0.5 finite elements, we list several performance estimates in Table 6. In this table, the results in Table 1b are duplicated for comparison. The difference between the algorithm outputs using these two meshes is negligible. Considering the modeling error from the diffusion model (which will be discussed in Section 5.5), we can assert that using 0.5×0.5 finite elements is sufficient to produce an accurate approximation for this queue.

Given an appropriate reference density and the associated truncation hypercube, the above discussion indicates an approach to selecting a mesh. Beginning with two meshes, with one finer than the other, we compare the algorithm outputs using these two meshes. If obvious difference is observed, the coarser mesh should be discarded and a further finer mesh should be evaluated. Continue this procedure until the difference between the outputs of two meshes is negligible. Then, the coarser one of the remaining two is selected as an appropriate mesh.

We would also demonstrate that with a wrong reference density, a finer mesh cannot make the algorithm yield an adequate output. Let us go back to the example in Section 5.1 with the "naive" reference density. We set the truncation rectangle to be $K = [-7, 32] \times [-7, 32]$ and the size of finite elements to be 0.25×0.25. The output is shown in Figure 7a. Although the curve by the "naive" reference density appears smoother than the one in Figure 5b with 0.5×0.5 finite elements, the output still fails to capture the exact stationary distribution. This time, the condition number of A is 3.91×10^{195}. There is no doubt that such an ill-conditioned matrix will bring about huge round-off error in solving (3.14). In contrast, with the same mesh and truncation rectangle, the algorithm yields accurate results in Figure 6b when the proposed reference density is used. A mesh with 0.125×0.125 finite elements is also investigated and the algorithm output



(A) 0.25×0.25 *finite elements*

(B) 0.125×0.125 *finite elements*

FIG 7: *The stationary distribution of the customer number in the $M/H_2/n + M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the finite element algorithm using the* "naive" reference density *and* different meshes, *and (ii) by the matrix-analytic method.*

using the "naive" reference density is plotted in Figure 7b. The condition number of A increases to 6.35×10^{198} and the algorithm misses the target as well.

5.3. *Gauss–Legendre quadrature.* Before solving the linear system (3.14), we must generate the matrix A and the vector v whose entries are given by (3.15). A Gauss–Legendre quadrature rule is followed to compute the integral for each entry. The integral is taken over a two-dimensional rectangle and the quadrature rule evaluates the integrand at m points in each dimension. The results are more accurate when a larger m is used. In Section 4, we take $m = 8$ in the numerical examples. Here, we briefly discuss the impact of the order m.

Several performance estimates are listed in Table 7. We keep the same settings for the algorithm except the quadrature order in each dimension and the size of finite elements. For the data in the first three columns, the size of finite elements is set to be 0.5×0.5. Clearly, the Gauss–Legendre quadrature of order $m \geq 4$ is sufficiently accurate for our purposes. To check the joint impact of the mesh and the quadrature order, we also explore the mesh using 0.25×0.25 squares, along with the quadrature order $m = 16$. Compared with other estimates in the table that use coarser meshes and lower quadrature orders, the algorithm output changes little.

5.4. *Computational complexity.* Let d, the dimension of the diffusion model, be fixed. The size of A is $m_C \times m_C$ where m_C is the dimension of the

Table 7

Performance measures of the $M/H_2/n+M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the finite element algorithm using different quadrature orders *and* different meshes *with the reference density in (3.21) and (3.23), and (ii) by the matrix-analytic method*

	$m = 4$	$m = 8$	$m = 16$	$m = 16$	Matrix-
	0.5×0.5	0.5×0.5	0.5×0.5	0.25×0.25	analytic
Mean queue length	54.17	54.17	54.17	54.17	54.05
Abandonment fraction	0.05181	0.05181	0.05181	0.05182	0.05173
$\mathbb{P}[N(\infty) > 470]$	0.9701	0.9701	0.9701	0.9702	0.9694
$\mathbb{P}[N(\infty) > 500]$	0.6833	0.6838	0.6839	0.6835	0.6818
$\mathbb{P}[N(\infty) > 600]$	0.2245	0.2244	0.2244	0.2241	0.2229
$\mathbb{P}[N(\infty) > 750]$	0.008235	0.008233	0.008232	0.008246	0.006395

Table 8

Computation time (in seconds) of the finite element algorithm using different meshes

	1.0×1.0	0.5×0.5	0.25×0.25	0.125×0.125
Dimension m_C	5776	23716	96100	386884
Constructing A and v	6.63	27.3	109	455
Solving (3.14)	0.0780	0.359	2.29	18.2

function space C given by (A.5) in the appendix. The matrix A is sparse. There are at most 6^d nonzero entries in each row or column. Hence, it takes $O(m_C)$ arithmetic operations to construct A. Gaussian elimination can be used to solve the linear system (3.14). When the basis functions are properly ordered, the nonzero entries of A are confined to a diagonally bordered band of width $O(m_C^{(d-1)/d})$. Hence, solving (3.14) requires $O(m_C^{(2d-1)/d})$ arithmetic operations as $m_C \to \infty$.

The computation time (measured by seconds) for various meshes can be found in Table 8, where we list both the time for constructing A and v and the time for solving (3.14). When computing A and v, we follow a Gauss–Legendre quadrature rule with $m = 8$ points in each dimension. The truncation rectangle is set to be $K = [-7, 32] \times [-7, 32]$. We change the size of finite elements to have different meshes. The dimension m_C increases by around four times as the width of each finite element is reduced by half. The proposed algorithm is tested on a laptop with a 2.66 GHz Intel Core 2 Duo processor and 8 GB memory. Both A and v are produced by a program written in C++. The linear system (3.14) is solved by Matlab. These two parts are connected via a MEX interface that comes with Matlab.

5.5. *Modeling error vs numerical error.* Performance estimates produced by the proposed algorithm contain both modeling and numerical error. Modeling error is present because the diffusion model is an approximation of the many-server queue. Numerical error (including both approximation and

TABLE 9

Performance measures of the $M/M/n+M$ queue, with $\rho = 1.045$ and $n = 500$, computed (i) by the Erlang-A formula (Exact), (ii) by (3.19) for the diffusion model (Analytical), (iii) by the finite element algorithm designed for $M/H_2/n+M$ queues using the reference density in (3.21) and (3.23) with $c_a^2 = c_s^2 = 1$ (Numerical A), and (iv) by the finite element algorithm designed for $M/H_2/n+M$ queues using the reference density in (3.21) and (3.23) with $c_a^2 = 1$ and $c_s^2 = 24$ (Numerical B)

	Exact	Diffusion		
		Analytical	Numerical A	Numerical B
Mean queue length	46.4309	46.4080	46.4072	46.4072
Abandonment fraction	0.0444433	0.0444214	0.0444216	0.0444215
$\mathbb{P}[N(\infty) > 480]$	0.987203	0.986631	0.986629	0.986629
$\mathbb{P}[N(\infty) > 500]$	0.930219	0.929269	0.929128	0.929123
$\mathbb{P}[N(\infty) > 550]$	0.444188	0.439271	0.439282	0.439283
$\mathbb{P}[N(\infty) > 600]$	0.0464741	0.0423851	0.0423975	0.0423963

round-off error) is from computation. For a many-server queue, the modeling error analysis of diffusion approximations has not been well studied in the literature. A comprehensive error analysis for the numerical algorithm is also beyond the scope of this paper. Instead, we evaluate modeling and numerical error through an example.

Consider an $M/M/n+M$ queue with $n = 500$ servers. The arrival rate is $\lambda = 522.36$, the mean service time is $1/\mu = 1$, and the mean patience time is $1/\alpha = 2$. The stationary distribution of the number of customers in system is given by the Erlang-A formula (see [21]). Using that, we can compute the exact performance measures of the queue. The diffusion model for this queue is a one-dimensional piecewise OU process, whose stationary distribution is given by (3.19) with $c_a^2 = 1$. Using (3.19), we can derive analytical expressions for the performance estimates. Hence, the modeling error can be evaluated by comparing the analytical results with the exact performance measures. See Table 9. To measure the two types of error, more digits are displayed for each quantity in this table.

The corresponding performance estimates by the proposed algorithm are also listed in Table 9. When computing the stationary density, the $M/M/n+M$ queue is regarded as an $M/H_2/n+M$ queue that has the same mean service time for the two types of customers. In other words, the queue is approximated by a two-dimensional diffusion process as in the previous numerical experiments. Here, we set

$$p = (0.9, 0.1)' \quad \text{and} \quad \nu = (1, 1)'.$$

The reference density follows (3.21) and (3.23) with $d = 2$ and $c_a^2 = c_s^2 = 1$. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is $K = [-7, 9] \times [-7, 9]$ and is divided into 0.5×0.5 finite elements. See the column labeled "Numerical

A" for the performance estimates. The numerical error from computation is the difference between each performance estimate by the algorithm and the corresponding analytical result. As the estimates from the algorithm are very close to the analytical results, we can clearly see that in this example the modeling error dominates and the numerical error is negligible. Note that the modeling error increases as the number of servers gets smaller. If the system has no more than several hundred servers, we can expect that the error in performance estimates is mostly from the approximate diffusion model rather than from computation.

In this example, the algorithm is also tested with another reference density. This time we adopt a reference density given by (3.21) and (3.23) with $d = 2$, $c_a^2 = 1$, but $c_s^2 = 24$. The decay rate of this reference density is much slower than that of the stationary density. With $\varepsilon_0 = 10^{-7}$, the truncation rectangle is now set to be $K = [-7, 32] \times [-7, 32]$, still divided into 0.5×0.5 finite elements. As the truncation rectangle is larger, more basis functions are used for this reference density. The algorithm output is displayed in the column labeled "Numerical B". Although the computation time is longer, the performance estimates still conform with the analytical results very well. This is in sharp contrast to the example in Section 5.1: The algorithm fails when a "naive" reference density that has a much faster decay rate is used. Comparing Figures 5a and 5b, we also see that with a wrong reference density, the algorithm may yield more error when a larger truncation rectangle is used. The current numerical example, however, indicates that the algorithm output is not sensitive to the reference density as long as its decay rate is not too fast. With such a reference density, the algorithm can produce stable output as long as the reference density is sufficiently small outside the selected truncation rectangle.

6. Concluding remarks.

In this paper, we proposed a diffusion model for many-server queues with customer abandonment. A finite element algorithm was developed for computing the stationary distribution of the model. An essential part of the algorithm is a reference density that controls the convergence of the algorithm. To construct the reference density, we conjectured that the limit queue length process has a certain Gaussian tail. Using this conjecture, we proposed a systematic approach to choosing a reference density. With the proposed reference density, the output of the algorithm is stable and accurate. Numerical examples indicate that the diffusion model is a good approximation for many-server queues.

Some more considerations for the algorithm are as follows. Assume that the stationary density g is twice differentiable in $\mathbb{R}^d$ and vanishes at infinity.

Using the basic adjoint relationship (2.7) and applying integration by parts twice, we have

$$\mathcal{G}^* g(x) = 0 \quad \text{for all } x \in \mathbb{R}^d$$

where $\mathcal{G}^*$ is the adjoint operator of the generator $\mathcal{G}$. Fix a finite domain $K \subset \mathbb{R}^d$ large enough. One can solve the stationary density g by the Dirichlet problem

$$\begin{cases} \mathcal{G}^* g(x) = 0 & \text{for } x \text{ in the interior of } K, \\ g(x) = 0 & \text{for } x \text{ on the boundary of } K. \end{cases}$$

Such a Dirichlet problem can be solved via the finite difference method. Alternatively, for each test function f, one may apply integration by parts once to the basic adjoint relationship to obtain an equation that involves the first order derivatives of g and the first order derivatives of f. From this weak formulation, fixing a large enough finite domain K and assuming that g is zero on the boundary of K, one may apply a standard Galerkin finite element method to compute the stationary density g on K. See, e.g., [18]. Both the finite difference method and the Galerkin method do not need a reference density. A future research topic is to compare the efficiency and accuracy of these two algorithms with the proposed algorithm in this paper.

The dimension of the function space C grows exponentially in d, the dimension of the diffusion model. As a consequence, both computation time and memory usage increase exponentially in d. When d is not small, the curse of dimensionality is a serious challenge for the proposed algorithm as well as for any other algorithms. To reduce the dimension of C, one possible approach is to investigate a reference density that potentially shares more common features with the stationary density. Such a reference density may enable us to compute the stationary density with a moderate number of basis functions when d is not small. Another possible direction to reduce the computational complexity of the algorithm is to investigate a low-rank matrix approximation for the linear system (3.14). The technique of random sampling may be explored. See [17] for more details.

APPENDIX: THE FINITE ELEMENT IMPLEMENTATION

In this appendix, we construct a sequence of function spaces $\{C_k : k \in \mathbb{N}\}$ using the finite element method. Each C_k is finite-dimensional and is used to generate the space H_k in (3.12). This finite element implementation follows the algorithm developed in [30]. Since the state space of the SRBM in [30] is bounded, neither a reference density nor state space truncation is used there.

Consider the d-dimensional diffusion process X in (2.1). As the state space of X is unbounded, it is necessary to truncate it to apply the finite element method. Let $\{K_k : k \in \mathbb{N}\}$ be a sequence of compact sets in $\mathbb{R}^d$. For each $f \in C_k$, we assume that $f(x) = 0$ for $x \in \mathbb{R}^d \backslash K_k$. For notational convenience, the subscript k is omitted when it is fixed, so K is the compact support of the finite-dimensional space C. In our implementation, we restrict K to be a d-dimensional hypercube

$$K = [-\zeta_1, \xi_1] \times \cdots \times [-\zeta_d, \xi_d],$$

where both ζ_j and ξ_j are positive constants for $j = 1, \ldots, d$. Once the reference density is determined, we can set the truncation hypercube by the following procedure: First, pick a small $\varepsilon_0 > 0$; then, choose a hypercube K such that

$$\text{(A.1)} \qquad \int_{\mathbb{R}^d \backslash K} r(x) \,\mathrm{d}x < \varepsilon_0.$$

When ε_0 is small enough, the influence of the reference density outside K is negligible in computing the stationary density.

We partition K into a finite number of subdomains. Such a partition is called a *mesh* and each subdomain is called a *finite element*. Since K is a hypercube, it is convenient to use a lattice mesh, where each finite element is again a hypercube. In this case, each corner point of a finite element is called a *node*. In dimension j, we divide the interval $[-\zeta_j, \xi_j]$ into n_j subintervals by partition points

$$-\zeta_j = y_j^0 < y_j^1 < \cdots < y_j^{n_j} = \xi_j.$$

Then, K is divided into $\prod_{j=1}^d n_j$ finite elements. For future reference, we label the nodes following the way that node $(i_1, \ldots, i_d)$ corresponds to spatial coordinate $(y_1^{i_1}, \ldots, y_d^{i_d})$, and define

$$h_j^\ell = y_j^{\ell+1} - y_j^\ell \quad \text{for } \ell = 0, \ldots, n_j - 1 \text{ and } j = 1, \ldots, d.$$

If Δ denotes such a mesh, we define

$$|\Delta| = \max\{h_j^\ell : \ell = 0, \ldots, n_j - 1;\ j = 1, \ldots, d\}$$

and

$$\text{(A.2)} \quad \eta_\Delta = \max\left\{\frac{h_{j_1}^{\ell_1}}{h_{j_2}^{\ell_2}} : \ell_1, \ell_2 = 0, \ldots, n_j - 1;\ j_1, j_2 = 1, \ldots, d;\ \ j_1 \neq j_2\right\}.$$

The space C is generated using the above mesh. We use the cubic Hermite basis functions to construct a basis of C, as in [30]. The one-dimensional Hermite basis functions for $-1 \leq z \leq 1$ are given by

$$\phi(z) = (|z|-1)^2(2\,|z|+1) \quad \text{and} \quad \psi(z) = z(|z|-1)^2. \tag{A.3}$$

In dimension $j = 1, \ldots, d$ and for $\ell = 1, \ldots, n_j - 1$, let

$$\phi_j^\ell(z) = \begin{cases} \phi\left(\dfrac{z - y_j^\ell}{h_j^{\ell-1}}\right) & \text{if } y_j^{\ell-1} \leq z \leq y_j^\ell, \\ \phi\left(\dfrac{z - y_j^\ell}{h_j^{\ell}}\right) & \text{if } y_j^{\ell} \leq z \leq y_j^{\ell+1}, \\ 0 & \text{otherwise} \end{cases}$$

and

$$\psi_j^\ell(z) = \begin{cases} h_j^{\ell-1}\psi\left(\dfrac{z - y_j^\ell}{h_j^{\ell-1}}\right) & \text{if } y_j^{\ell-1} \leq z \leq y_j^\ell, \\ h_j^{\ell}\psi\left(\dfrac{z - y_j^\ell}{h_j^{\ell}}\right) & \text{if } y_j^{\ell} \leq z \leq y_j^{\ell+1}, \\ 0 & \text{otherwise.} \end{cases}$$

Let $x = (x_1, \ldots, x_d)'$ be a vector in K. At node $(i_1, \ldots, i_d)$, the basis functions of C are the tensor-product Hermite basis functions

$$f_{i_1,\ldots,i_d,\chi_1,\ldots,\chi_d}(x) = \prod_{j=1}^{d} g_{i_j,\chi_j}(x_j) \tag{A.4}$$

where χ_j is either 0 or 1 and

$$g_{i_j,\chi_j}(z) = \begin{cases} \phi_j^{i_j}(z) & \text{if } \chi_j = 0, \\ \psi_j^{i_j}(z) & \text{if } \chi_j = 1. \end{cases}$$

Therefore, each node has 2^d tensor-product basis functions and the space C has a total of

$$m_C = 2^d \prod_{j=1}^{d} (n_j - 1) \tag{A.5}$$

basis functions.

The space C is not a subspace of $C_b^2(\mathbb{R}^d)$. For the one-dimensional Hermite basis functions in (A.3), the second order derivative of $\phi(z)$ is not defined at $z = -1$ and 1, and the second order derivative of $\psi(z)$ is not defined at $z = -1$, 0, and 1. As a consequence, there exists $f \in C$ for which $\mathcal{G}f$ is not defined on the boundaries of certain finite elements. Because such boundaries have Lebesgue measure zero in $\mathbb{R}^d$, for each $f \in C$, we can find a sequence of functions $\{\varphi_i : i \in \mathbb{N}\}$ in $C_b^2(\mathbb{R}^d)$ such that $\|\mathcal{G}\varphi_i - \mathcal{G}f\| \to 0$ as $i \to \infty$. Hence, $H_k \subset H$ still holds for each k.

For the linear system (3.14) to have a unique solution, the family of functions

$$\{\mathcal{G}f_{i_1,\ldots,i_d,\chi_1,\ldots,\chi_d} : i_j = 1, \ldots, n_j - 1; \chi_j = 0, 1; j = 1, \ldots, d\}$$

must be linearly independent in $L^2(\mathbb{R}^d, r)$. The following proposition provides sufficient conditions for the linear independence.

Proposition 2. *Let r be a positive function on $\mathbb{R}^d$ that satisfies (3.1). Let $\mathcal{G}$ be the operator in (2.6) such that conditions (2.2) and (2.3) hold and all entries of Σ are continuously differentiable. Then, the family of functions*

$$\{\mathcal{G}f_{i_1,\ldots,i_d,\chi_1,\ldots,\chi_d} : i_j = 1, \ldots, n_j - 1; \chi_j = 0, 1; j = 1, \ldots, d\}$$

is linearly independent in $L^2(\mathbb{R}^d, r)$, where $f_{i_1,\ldots,i_d,\chi_1,\ldots,\chi_d}$ is the basis function of C given by (A.4). Consequently, the solution to the linear system (3.14) is unique.

Proof. We use $C_0^1(K)$ to denote the set of real-valued functions on a neighborhood of K that are continuously differentiable and have compact support in K. Clearly, $C \subset C_0^1(K)$. For any $f, \hat{f} \in C_0^1(K)$, we define an inner product by

$$\langle f, \hat{f} \rangle_{D(K)} = \sum_{j=1}^{d} \int_K \frac{\partial f(x)}{\partial x_j} \frac{\partial \hat{f}(x)}{\partial x_j} \, \mathrm{d}x$$

and let $W_0^{1,2}(K)$ be the closure of $C_0^1(K)$ in the norm induced by this inner product. Then, $W_0^{1,2}(K)$ is a Hilbert space and $C \subset W_0^{1,2}(K)$.

Since $\mathcal{G}$ is a linear operator, it suffices to show that for any $f_0 \in C$, we must have $f_0 = 0$ if $\mathcal{G}f_0 = 0$ in $L^2(\mathbb{R}^d, r)$. The uniform elliptic operator $\mathcal{G}$ can be written into the divergence form as in (8.1) of [12], i.e.,

$$\mathcal{G}f(x) = \sum_{j=1}^{d} \hat{b}_j(x) \frac{\partial f(x)}{\partial x_j} + \frac{1}{2} \sum_{j=1}^{d} \sum_{\ell=1}^{d} \frac{\partial(\Sigma_{j\ell}(x) \partial f(x)/\partial x_j)}{\partial x_\ell}$$

for each $f \in C_b^2(\mathbb{R}^d)$, where

$$\hat{b}_j(x) = b_j(x) - \frac{1}{2}\sum_{\ell=1}^{d} \frac{\partial \Sigma_{j\ell}(x)}{\partial x_\ell}.$$

Let $U \subset \mathbb{R}^d$ be a connected open set that is bounded and contains K. Since $r > 0$ and $\mathcal{G} f_0$ is continuous in the interior of each finite element, we must have $\mathcal{G} f_0 = 0$ in K except on the boundaries of certain finite elements where $\mathcal{G} f_0$ is not defined. Hence, $\mathcal{G} f_0 = 0$ in U in the weak sense (see (8.2) of [12]). Note that b, Σ, and the partial derivatives of Σ are all continuous, so both $\hat{b}$ and Σ are bounded in U. Because $f_0 \in W_0^{1,2}(K)$, it follows from Corollary 8.2 in [12] that $f_0 = 0$ in K, and thus $f_0 = 0$ in $\mathbb{R}^d$. □

When using the finite element algorithm to solve the stationary density of the diffusion model (2.22), it follows from Proposition 2 that the linear system (3.14) has a unique solution.

Now consider a sequence of function spaces $\{C_k : k \in \mathbb{N}\}$. Let Δ_k be the mesh for constructing C_k. We assume that Δ_{k+1} is a refinement of Δ_k, i.e., a node or an interelement boundary in Δ_k is also a node or an interelement boundary in Δ_{k+1}. We also assume that the refinements are *regular*, i.e., $\sup\{\eta_{\Delta_k} : k \in \mathbb{N}\} < \infty$ for η_{Δ_k} defined by (A.2). The next proposition, along with Proposition 1, justifies our finite element implementation for computing the stationary distribution.

PROPOSITION 3. *Let r be a positive function on $\mathbb{R}^d$ that satisfies (3.1). Let $\{\Delta_k : k \in \mathbb{N}\}$ be a sequence of lattice meshes such that each Δ_{k+1} is a refinement of Δ_k and the refinements are regular. Let K_k be the d-dimensional finite hypercube that is the domain of Δ_k, and C_k be the function space generated by Δ_k using the tensor-product Hermite basis functions in (A.4). Let H be the infinite-dimensional space in (3.7) and H_k be the finite-dimensional subspace in (3.12), where the generator $\mathcal{G}$ satisfies (2.2) and (3.3). Assume that*

$$|\Delta_k| \to 0 \quad \text{and} \quad K_k \uparrow \mathbb{R}^d \quad \text{as } k \to \infty.$$

Then,

$$H_k \to H \ \text{in } L(\mathbb{R}^d, r) \quad \text{as } k \to \infty.$$

PROOF. Given a compact set $K \subset \mathbb{R}^d$, let $C_b^2(K)$ be the set of real-valued functions on a neighborhood of K that are twice continuously differentiable with bounded first and second order derivatives in K. For each $f \in C_b^2(K)$, define a norm $\|\cdot\|_{H^2(K)}$ by

$$\|f\|^2_{H^2(K)} = \int_K \left(f^2(x) + \max_{j=1,\dots,d} \left(\frac{\partial f(x)}{\partial x_j} \right)^2 + \max_{j,\ell=1,\dots,d} \left(\frac{\partial^2 f(x)}{\partial x_j \partial x_\ell} \right)^2 \right) r(x) \, \mathrm{d}x.$$

Because both b and Σ are bounded in K, there exists $\kappa_0(K) > 0$ such that

$$\text{(A.6)} \qquad \int_K (\mathcal{G} f(x))^2 r(x) \, \mathrm{d}x \le \kappa_0(K) \, \|f\|^2_{H^2(K)} \quad \text{for all } f \in C_b^2(K).$$

Let $\bar{C}_b^2(K)$ be the closure of $C_b^2(K)$ in the above norm. A standard procedure can be used to define the first and second order derivatives for each $f \in \bar{C}_b^2(K)$. Then, the operator $\mathcal{G}$ can be extended to $\bar{C}_b^2(K)$ and inequality (A.6) holds for all $f \in \bar{C}_b^2(K)$.

To prove the proposition, it suffices to prove that for any $f_0 \in C_b^2(\mathbb{R}^d)$, there exists a sequence of functions $\{\varphi_k \in C_k : k \in \mathbb{N}\}$ such that

$$\|\mathcal{G}\varphi_k - \mathcal{G} f_0\| \to 0 \quad \text{as } k \to \infty.$$

Fix $\varepsilon > 0$. Because $K_k \uparrow \mathbb{R}^d$ as $k \to \infty$, by (3.3) and the Cauchy–Schwartz inequality, there exists $a \in \mathbb{N}$ such that

$$\text{(A.7)} \qquad \int_{\mathbb{R}^d \setminus K_a} (\mathcal{G} f_0(x))^2 r(x) \, \mathrm{d}x < \frac{\varepsilon^2}{2}.$$

Consider the finite hypercube K_a. By (A.6), there is $\kappa_0(K_a) > 0$ such that

$$\text{(A.8)} \qquad \int_{K_a} (\mathcal{G} f(x))^2 r(x) \, \mathrm{d}x \le \kappa_0(K_a) \, \|f\|^2_{H^2(K_a)} \quad \text{for all } f \in \bar{C}_b^2(K_a).$$

A polynomial is used to approximate f_0 on K_a. By Proposition 7.1 in the appendix of [8], there exists a polynomial f_p such that

$$\|f_\mathrm{p} - f_0\|_{H^2(K_a)} < \frac{\varepsilon}{2\sqrt{2\kappa_0(K_a)}}.$$

For the lattice mesh Δ_k, let $\Lambda_{a,k}$ be the set of its nodes in the interior of K_a. For any $k \ge a$, let φ_k be a function in C_k such that $\varphi_k(x) = 0$ for all $x \in \mathbb{R}^d \setminus K_a$ and

$$\varphi_k(x) = f_\mathrm{p}(x) \quad \text{and} \quad \frac{\partial \varphi_k(x)}{\partial x_j} = \frac{\partial f_\mathrm{p}(x)}{\partial x_j}$$

for $j = 1, \dots, d$ and all $x \in \Lambda_{a,k}$. Clearly, $\varphi_k \in \bar{C}_b^2(K_a)$. Because the sequence of meshes $\{\Delta_k : k \in \mathbb{N}\}$ is regularly refined, there exists a constant $\kappa_1 > 0$ such that $\eta_{\Delta_k} < \kappa_1$ for all $k \geq a$. Using the interpolation error estimate by Theorem 6.6 in [23], we have

$$\|\varphi_k - f_{\mathrm{p}}\|_{H^2(K_a)} \leq \kappa_1^2 \kappa_2 \kappa_3 \Big(\int_{\mathbb{R}^d} r(x)\,\mathrm{d}x \Big)^{1/2} |\Delta_k|^2 ,$$

where $\kappa_2 > 0$ is a constant independent of Δ_k and f_{p}, and

$$\kappa_3 = \sup \left\{ \left| \frac{\partial^4 f_{\mathrm{p}}(x)}{\partial x_1^{m_1} \cdots \partial x_d^{m_d}} \right| : x \in K_a;\, m_1 + \cdots + m_d = 4 \right\} < \infty.$$

Hence, there exists $\delta_0 > 0$ such that

$$\|\varphi_k - f_{\mathrm{p}}\|_{H^2(K_a)} < \frac{\varepsilon}{2\sqrt{2\kappa_0(K_a)}}$$

whenever $|\Delta_k| < \delta_0$. In this case,

$$\|\varphi_k - f_0\|_{H^2(K_a)} \leq \|\varphi_k - f_{\mathrm{p}}\|_{H^2(K_a)} + \|f_{\mathrm{p}} - f_0\|_{H^2(K_a)} < \frac{\varepsilon}{\sqrt{2\kappa_0(K_a)}}.$$

By (A.8),

$$\int_{K_a} (\mathcal{G}\varphi_k(x) - \mathcal{G}f_0(x))^2 r(x)\,\mathrm{d}x \leq \kappa_0(K_a)\, \|\varphi_k - f_0\|^2_{H^2(K_a)} < \frac{\varepsilon^2}{2}. \tag{A.9}$$

It follows from (A.7) and (A.9) that as long as $k \geq a$ and $|\Delta_k| < \delta_0$, we must have $\|\mathcal{G}\varphi_k - \mathcal{G}f_0\| < \varepsilon$.

□

Acknowledgements. The authors would like to thank Ton Dieker and Vadim Linetsky for their helpful comments on solving the basic adjoint relationship. We would also like to thank the associate editor and an anonymous referee for their suggestions about the paper's organization.

REFERENCES

[1] Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005). Statistical analysis of a telephone call center: A queueing-science perspective. *J. Amer. Statist. Assoc.* **100**, 469, 36–50. MR2166068

[2] Browne, S. and Whitt, W. (1995). Piecewise-linear diffusion processes. In *Advances in Queueing: Theory, Methods, and Open Problems*, J. H. Dshalalow, Ed. CRC, Boca Raton, FL, 463–480. MR1395170

[3] Dai, J. G. and Dieker, A. B. (2011). Nonnegativity of solutions to the basic adjoint relationship for some diffusion processes. *Queueing Syst.* **68**, 3–4, 295–303. MR2834200

[4] Dai, J. G. and Harrison, J. M. (1991). Steady-state analysis of RBM in a rectangle: Numerical methods and a queueing application. *Ann. Appl. Probab.* **1**, 1, 16–35. MR1097462

[5] Dai, J. G. and Harrison, J. M. (1992). Reflected Brownian motion in an orthant: Numerical methods for steady-state analysis. *Ann. Appl. Probab.* **2**, 1, 65–86. MR1143393

[6] Dai, J. G., He, S., and Tezcan, T. (2010). Many-server diffusion limits for $G/Ph/n + GI$ queues. *Ann. Appl. Probab.* **20**, 5, 1854–1890. MR2724423

[7] Dieker, A. B. and Gao, X. (2013). Positive recurrence of piecewise Ornstein–Uhlenbeck processes and common quadratic Lyapunov functions. *Ann. Appl. Probab.*. To appear.

[8] Ethier, S. N. and Kurtz, T. G. (1986). *Markov Processes: Characterization and Convergence*. Wiley, New York. MR838085

[9] Gamarnik, D. and Goldberg, D. A. (2013). Steady-state $GI/GI/n$ queue in the Halfin–Whitt regime. *Ann. Appl. Probab.*. To appear.

[10] Gamarnik, D. and Momčilović, P. (2008). Steady-state analysis of a multiserver queue in the Halfin–Whitt regime. *Adv. in Appl. Probab.* **40**, 2, 548–577. MR2433709

[11] Garnett, O., Mandelbaum, A., and Reiman, M. (2002). Designing a call center with impatient customers. *Manufacturing & Service Operations Management* **4**, 3, 208–227.

[12] Gilbarg, D. and Trudinger, N. S. (2001). *Elliptic Partial Differential Equations of Second Order*. Springer–Verlag, Berlin. MR1814364

[13] Halfin, S. and Whitt, W. (1981). Heavy-traffic limits for queues with many exponential servers. *Oper. Res.* **29**, 3, 567–588. MR629195

[14] Harrison, J. M. and Nguyen, V. (1990). The QNET method for two-moment analysis of open queueing networks. *Queueing Syst.* **6**, 1, 1–32. MR1053666

[15] Iglehart, D. L. (1965). Limiting diffusion approximations for the many server queue and the repairman problem. *J. Appl. Probab.* **2**, 2, 429–441. MR0184302

[16] Iglehart, D. L. and Whitt, W. (1970). Multiple channel queues in heavy traffic II: Sequences, networks, and batches. *Adv. in Appl. Probab.* **2**, 2, 355–369. MR0282443

[17] Kannan, R. and Vempala, S. (2008). Spectral algorithms. *Found. Trends Theor. Comput. Sci.* **4**, 3–4, 157–288. MR2558901

[18] Kovalov, P., Linetsky, V., and Marcozzi, M. (2007). Pricing multi-asset American options: A finite element method-of-lines with smooth penalty. *J. Sci. Comput.* **33**, 3, 209–237. MR2357409

[19] Kress, R. (1998). *Numerical Analysis*. Springer–Verlag, New York. MR1621952

[20] Latouche, G. and Ramaswami, V. (1999). *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, Philadelphia, PA. MR1674122

[21] Mandelbaum, A. and Zeltyn, S. (2007). Service engineering in action: The Palm/Erlang-A queue with applications to call centers. In *Advances in Services Innovations*, D. Spath and K.-P. Fähnrich, Eds. Springer–Verlag, Berlin, 17–45.

[22] Neuts, M. F. (1981). *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Johns Hopkins University Press, Baltimore, MD. MR618123

[23] Oden, J. T. and Reddy, J. N. (1976). *An Introduction to the Mathematical Theory of Finite Elements*. Wiley, New York. MR0461950

[24] Øksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*, 6th ed. Springer–Verlag, Berlin. MR2001996 (2004e:60102)

[25] Puhalskii, A. A. and Reiman, M. I. (2000). The multiclass $GI/PH/N$ queue in the Halfin–Whitt regime. *Adv. in Appl. Probab.* **32**, 2, 564–595. Correction: **36**, 3, 971 (2004). MR1778580

[26] Reed, J. and Tezcan, T. (2012). Hazard rate scaling of the abandonment distribution for the $GI/M/n + GI$ queue in heavy traffic. *Oper. Res.* **60**, 4, 981–995. MR2979435

[27] Reed, J. E. and Ward, A. R. (2008). Approximating the $GI/GI/1 + GI$ queue with a nonlinear drift diffusion: Hazard rate scaling in heavy traffic. *Math. Oper. Res.* **33**, 3, 606–644. MR2442644

[28] Reiman, M. I. (1984). Open queueing networks in heavy traffic. *Math. Oper. Res.* **9**, 3, 441–458. MR757317

[29] Saure, D., Glynn, P., and Zeevi, A. (2009). A linear programming algorithm for computing the stationary distribution of semimartingale reflected Brownian motion. Tech. rep., Graduate School of Business, Columbia University.

[30] Shen, X., Chen, H., Dai, J. G., and Dai, W. (2002). The finite element method for computing the stationary distribution of an SRBM in a hypercube with applications to finite buffer queueing networks. *Queueing Syst.* **42**, 1, 33–62. MR1943968

[31] Whitt, W. (1982). On the heavy-traffic limit theorem for $GI/G/\infty$ queues. *Adv. in Appl. Probab.* **14**, 1, 171–190. MR644013

[32] Whitt, W. (2005). Heavy-traffic limits for the $G/H_2^*/n/m$ queue. *Math. Oper. Res.* **30**, 1, 1–27. MR2125135

[33] Williams, R. J. (1996). On the approximation of queueing networks in heavy traffic. In *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, and I. Ziedins, Eds. Oxford University Press, Oxford, UK, 35–56.

[34] Zeltyn, S. and Mandelbaum, A. (2005). Call centers with impatient customers: Many-server asymptotics of the $M/M/n+G$ queue. *Queueing Syst.* **51**, 3–4, 361–402. MR2189598

J. G. Dai
School of Operations Research
and Information Engineering
Cornell University
Ithaca, New York 14853, USA
E-mail: jim.dai@cornell.edu

Shuangchi He
Department of Industrial
and Systems Engineering
National University of Singapore
Singapore 117576
E-mail: heshuangchi@nus.edu.sg