JOURNAL OF APPLIED MATHEMATICS AND DECISION SCIENCES, 6(1), 43–50

# Calculation of Steady-State Probabilities of M/M Queues: Further Approaches

DAVID K. SMITH[†]
*School of Mathematical Sciences, University of Exeter,*
*Exeter EX4 4QE United Kingdom*

**Abstract.** This paper draws attention to the risk of rounding error in the numerical evaluation of steady-state probabilities for the M/M family of queues. A method for avoiding the risk is presented which is easy to program for calculation in practice.

**Keywords:** Accuracy, computing, queue models.

## 1. Introduction

The M/M family of queues is generally the first to be introduced to students of operational research and queueing theory. The assumptions made in the model are that customers arrive randomly at a service point at a known rate $\lambda$ per unit time, wait until a server is available and then require a random time with an exponential distribution with mean $1/\mu$ for service. It is possible to calculate the steady state probabilities for the number of customers in the queue system for many variants of this family of queues, and the results appear in many textbooks.

However, little notice appears to have been taken of the practicalities of calculating these probabilities numerically. Recently, Pasternack and Drezner([2]) presented a recursive approach which avoided some computational difficulties. In this note, we suggest a different approach which gives similar computational accuracy, and is easy to use by students who are familiar with spreadsheets. Our interest in the problem arose when circumstances were encountered where the obvious approaches to solving the queueing models proved liable to severe rounding error.

[†] Requests for reprints should be sent to David K. Smith, School of Mathematical Sciences, University of Exeter, Exeter EX4 4QE United Kingdom

## 2. Rounding Errors

In most references to computational accuracy, computer programmers are warned about the risks of rounding errors, and particularly truncation of small numbers to zero. For instance, the first chapter of Knuth[1], a text familiar to many students of computer science, works through an example of how rounding errors quickly lead to inaccurate results in a computational algorithm. Numerous examples can be cited of the need to avoid particular types of calculation, known to cause difficulties. A simple instance is the calculation of the sum and the individual terms of a geometric series, $S_N = \sum_0^N x^n$ for $|x| < 1$. Leaving aside the possibility of finding the sum analytically, the calculation may be organized from "left to right" or from "right to left". The first would calculate the first term, and then subsequent terms by multiplying by $x$; the second would calculate the last term, and then use division by $x$ to find the others. The second method relies on the accurate calculation and storage of $x^N$, which is liable to rounding errors, depending on the values of $x$ and $N$. For small values of $x$ and large $N$, say $x = 0.1$ and $N = 100$, $x^N$ may be too small to be treated as distinct from zero. This rounding may not be registered by the computer, so that the user will be ignorant of the inaccuracy of calculation. The recurrence relation will generate all terms in the sum to be zero as well, and the sum will be found to be zero. In contrast, the first method will be more accurate.

## 3. Calculating the Steady State Probabilities of Queues

This problem can be encountered when attempting to find the steady state probabilities of busy multi-server queues. For the sake of illustration, a queue with several servers and limited waiting room will be discussed. Pasternack and Drezner discussed the problems of rounding errors where the waiting room is infinite, although this can be treated as a special case of the model with finite room.

Given an $M/M/C/K$ queue (that is Poisson process arrivals, exponential service time, $C$ servers and space for $K \geq C$ customers in the system), the steady state probabilities can be readily derived to be:

$$\begin{aligned}
p_0 &= 1/Q \\
p_1 &= \frac{\lambda}{\mu} p_0 \\
p_2 &= \frac{\lambda}{\mu} \frac{p_1}{2} \\
&\ldots\ldots \\
p_C &= \frac{\lambda}{\mu} \frac{p_{C-1}}{C} \\
p_{C+1} &= \frac{\lambda}{C\mu} p_C \\
&\ldots\ldots \\
p_K &= \frac{\lambda}{C\mu} p_{K-1} \\
Q &= 1 + \sum_{r=1}^{C} \frac{\lambda^r}{r!\mu^r} + \sum_{r=C+1}^{K} \frac{\lambda^r}{C!\mu^r C^{r-C}}
\end{aligned}$$

where $\lambda$ is the arrival rate and $\mu$ the service rate. Such a result can be found in both general textbooks of operational research (e.g. Taha[3]) or specialised discussions of queueing models (e.g. Tijms[4]). In many circumstances, the order of calculations will be to find $Q$ first of all, then to take its reciprocal to obtain $p_0$, and then make use of the recurrence relationship to find all the steady state probabilities. Working in this way, very little computer storage is needed, and the recurrence formulæ can be evaluated rapidly. Essentially, using these equations, one is calculating the probability that the system is empty, and then building on that.

However, such a form of calculation is prone to errors, in circumstances where $Q$ is so large that $p_0$ is negligible. The precise values of the queue parameters when this may occur will depend on the accuracy of one's computer/calculator. For the sake of illustration, we assume that all calculations are rounded to five decimal places, although most students would have the facilities to work to greater accuracy.

For an $M/M/30/40$ queue, which might be appropriate for modelling a telephone call centre, $\lambda = 10\mu$ would give $p_0 = 0.00005$, rounded up from six-figure accuracy of 0.000 045. Use of the formulæ leads to the absurd position that the sum of all the steady state probabilities is more than 1, since all the terms in the sum will be scaled by the constant multiplier 0.000 05/0.000 045 or an increase of 11%. Increasing $\lambda$ to $15\mu$ gives $p_0 = 0.0$

to five decimal places, and so all the steady state probabiliies would be treated as zero. Other values of the queue parameters could lead to the rounded value of $p_0$ being almost twice, or about two-thirds, of its correct value.

In order to try and avoid these rounding errors, one could adopt the method employed for calculating the sum of the geometric series earlier, and start with the last term, which is the probability that the system is full. The order of calculation would be essentially the reverse of that given earlier.

$$\begin{aligned} p_K &= 1/Q' \\ p_{K-1} &= \frac{C\mu}{\lambda} p_K \\ &\cdots\cdots \\ p_0 &= \frac{\mu}{\lambda} p_1 \\ Q' &= 1 + \sum_{r=1}^{C} \frac{r!\mu^r}{\lambda^r} + \sum_{r=C+1}^{K} \frac{C!\mu^r C^{r-C}}{\lambda^r} \end{aligned}$$

In this set of equations, instead of defining the steady-state probabilities from an empty queue up to a full one, the calculation runs from a full queue to an empty one.

Sadly, the same problem occurs. Truncation error may mean that the value of $p_K$ is zero, or is rounded to be significantly different from its correct value. For instance, with $\lambda = 20\mu$, use of $Q'$ leads to 1.1% error in the evaluation of the probabilities, and with $\lambda = 15\mu$, $1/Q'$ is zero when rounded to five decimal places.

## 4. A Safer Method of Calculating Steady State Probabilities.

A third recursive way of finding the steady state probabilities exists, and this is much safer than the two widely taught methods. Minimal rounding errors will occur if the probabilities of the steady states are calculated relative to the largest such probability. For the $M/M/C/K$ queue, this is $p_K$ if $\lambda \geq C\mu$ or $p_r$ with $r = \lfloor \frac{\lambda}{\mu} \rfloor$ (the integer part of $\frac{\lambda}{\mu}$) otherwise. The case where $p_K$ is largest has been described above, but the second case presents a new situation.

Evaluating the probabilities relative to $p_r$ ($r$ defined above) provides a safe and straightforward method of calculation, when $\lambda < C\mu$. This can be stated as an algorithm:

1. Find the most probable state of the system, $r = \lfloor \frac{\lambda}{\mu} \rfloor$

2. Let $t_r = 1$

3. For $0 \leq i < r$, let $t_i$ be defined as

$$t_i = \left(\frac{(i+1)\mu}{\lambda}\right) t_{i+1}$$

with the calculation starting with $i = r - 1$ and descending to $i = 0$

4. For $r < i \leq C$, let $t_i$ be defined as

$$t_i = \left(\frac{\lambda}{(i-1)\mu}\right) t_{i-1}$$

($i$ ascending)

5. For $C < i \leq K$, let $t_i$ be defined as

$$t_i = \left(\frac{\lambda}{C\mu}\right) t_{i-1}$$

($i$ ascending)

6. Let

$$Q'' = \sum_{i=0}^{K} t_i \quad \text{and then} \quad p_i = \frac{t_i}{Q''} \quad i = 0, \ldots, K$$

Results of this process are shown in tables 1, 2 and 3. It will be seen that the calculations are much more precise than the examples quoted with $Q$ and $Q'$, and the method has accurately given non-zero probabilities in all the cases included. Further, the sum of all the steady-state probabilities ($K + 1$ terms) are very close to 1.

It may be argued that most calculations in practice will be carried to much greater accuracy than has been seen here. This is true. However, the problems of rounding and error propagation will exist to some extent, whenever steady state equations for queues of the form described here are evaluated. In many applications, the most likely state of the queueing system is for many servers to be busy. It would be dangerous to calculate the steady state probabilities on the basis of an event of small probability.

*Table 1.* Steady state probabilities: $\lambda = 10\mu$, recurrence defined by $p_{10}$ rounded to 5 decimal places

| | $\lambda = 10\mu$ | |
|---|---|---|
| Number in system | Exact probability | Prob. from recurrence |
| 0 | 0.000 045 | 0.000 05 |
| 5 | 0.037 833 | 0.037 84 |
| 10 | 0.125 110 | 0.125 11 |
| 15 | 0.034 718 | 0.034 72 |
| 20 | 0.001 866 | 0.001 87 |
| 30 | 0.000 000 17 | 0.0 |
| 40 | $2.9 \times 10^{-12}$ | 0.0 |
| Sum(all states, $i$=0 to 40) | 1.0 | 1.000 01 |

*Table 2.* Steady state probabilities: $\lambda = 15\mu$, recurrence defined by $p_{15}$ rounded to 5 decimal places

| | $\lambda = 15\mu$ | |
|---|---|---|
| Number in system | Exact probability | Prob. from recurrence |
| 0 | 0.000 000 31 | 0.0 |
| 5 | 0.001 936 | 0.001 94 |
| 10 | 0.048 610 | 0.048 61 |
| 15 | 0.102 433 | 0.102 43 |
| 20 | 0.041 809 | 0.041 81 |
| 30 | 0.000 221 | 0.000 22 |
| 40 | 0.000 000 22 | 0.0 |
| Sum(all states, $i$=0 to 40) | 1.0 | 1.000 01 |

*Table 3.* Steady state probabilities: $\lambda = 20\mu$, recurrence defined by $p_{20}$ rounded to 5 decimal places

| | $\lambda = 20\mu$ | |
|---|---|---|
| Number in system | Exact probability | Prob. from recurrence |
| 0 | 0.000 000 002 1 | 0.0 |
| 5 | 0.000 055 | 0.000 05 |
| 10 | 0.005 799 | 0.005 80 |
| 15 | 0.051 498 | 0.051 50 |
| 20 | 0.088 576 | 0.088 58 |
| 30 | 0.008 319 | 0.008 32 |
| 40 | 0.000 144 | 0.000 15 |
| Sum(all states, $i$=0 to 40) | 1.0 | 1.000 09 |

## 5. Further Examples and the Use of Spreadsheets

If the waiting room is infinite, then the queue will have a steady state only if $\lambda < C\mu$. The steady state probabilities can be found using the algorithm that has just been described. However, instead of treating $Q''$ as the sum of a finite number of terms, it will be necessary to use the analytic form of the sum of the infinite series:

$$\sum_{i=C}^{\infty} t_i = t_C \left(1 + \left(\frac{\mu}{C\lambda}\right) + \left(\frac{\mu}{C\lambda}\right)^2 + \dots\right) = \frac{t_C\lambda}{(C\lambda - \mu)}$$

With this modification, the steady state probabilities can be found for such queues.

Other properties of these queues may readily be found. The mean number of customers in the system $L$, is the sum of $rP_r$; the mean number waiting, $L_q$ is found by summing $(r - C)p_r$ for $r > C$. Little's formulae apply in their modified form, $L = \lambda(1 - p_n)W, L_q = \lambda(1 - p_n)W_q$ allowing the mean waiting anad queuing times to be calculated.

The algorithm is also well-suited to calculation in a spreadsheet, such as Excel, where there are logical tests available for calculating cell values. For given values of the four parameters $\lambda, \mu, C, K$, a column can be defined whose values are the $t_i$ values. Then the value of the individual cells can be found by nesting several logical expressions.

1. Find the state $r$ which will have $t_r = 1$ by the rule: "if $\frac{\mu}{\lambda} > C$ then $r = K$ else $r = \lfloor\frac{\mu}{\lambda}\rfloor$".

2. Find the individual values of $t_i$.

   - If $i > K$, then $t_i = 0$.
   - If $i = r$ then $t_i = 1$.
   - If $i < r$ then $t_i = t_{i+1} \times \left(\frac{\min(C,i+1)\mu}{\lambda}\right)$
   - If $i > r$ then $t_i = t_{i-1} \times \left(\frac{\lambda}{\min(C,i)\mu}\right)$

3. Summing the column gives the reciprocal of $p_r$ and the steady state probabilities follow at once.

In a spreadsheet, other queueing situations can be developed with similar structures to this. The author has used Excel with a student class to find the minimum number of servers that would maintain a given service level when the waiting room size $K$ was fixed relative to $C$. A student project on a telephone call-centre provided an unusual service rule: when the number waiting exceeds a particular value, the office manager starts to accept telephone calls. Although the duration of calls was not exponentially distributed, a spreadsheet model which mimicked the varying number of servers with the size of the queue provided a “quick-and-dirty” way of studying this queue.

Finally, similar recurrence relations arise in consideration of the probabilities of a Poisson distribution, for large means. Instead of calculating $p(X = 0) = \exp(-\lambda)$ and $p(X = i) = \left(\frac{\lambda}{i}\right) p(X = i - 1)$ one can find the mode of the distribution and calculate the probabilities relative to that.

## 6. Conclusion

This paper has shown how errors may arise when using two obvious ways of finding steady state probabilities for queues in the $M/M/C/K$ family, and has demonstrated a safer approach to such calculation. The approach is easy to code into a spreadsheet, and can be used with a wide range of elementary queues.

## References

1. Knuth DE (1968) *The Art of Computer Programming. Volume 1: Fundamental Algorithms.* Addison-Wesley, London
2. Pasternack, BA and Drezner, Z (1998) *A Note on Calculating Steady State Results for an M/M/k Queueing System when the Ratio of the Arrival Rate to the Service Rate is Large.* Journal of Applied Mathematics and Decision Sciences, **2**(2) p201–203
3. Taha HA (1992) *Operations Research: an introduction (5th edition).* Macmillan, New York
4. Tijms HC (1986) *Stochastic Modelling and Analysis: a Computational Approach.* Wiley, Chichester, UK