MTH5119 Lab 8: Simulation of Queueing Systems

In the previous two labs you learnt how to generate a sequence of independent realizations of exponentially distributed random variables. This week we will use these to study properties of queueing systems that might be used in a bank, at first with a single server, then with two servers. There are a very large variety of problems that can be analyzed by this simulation method. Remember of course that any conclusions we derive are subject to simulation error (not studied in this lab). Analytical solutions (rather than those by simulation) are possible for some simple systems, notably the single server queue with exponentially distributed arrivals and service times. Where the mean service time is $1/\beta$ and the mean *interarrival* time is $1/\alpha$, and $\alpha < \beta$, Probability III shows that the mean waiting time (mean time in the system) for a new arrival is $1/(\beta - \alpha)$, provided the system is in equilibrium, that is when effects of initial conditions have worn off. Then the *distributions* of waiting times, queue size etc do not vary with time and are called *stationary*.

The standard simulation method is to use a master clock starting at time zero with some specified initial conditions and then run the simulation for as long as possible so that long term averages such as mean waiting time per customer settle down to near their true value in a similar fashion to the *law of large numbers*. This method is illustrated in the notes and later in this Lab but you will use a special method (only for single server queues) first.

1. Using Calc→Make Patterned data→Simple Set of Numbers, put $1,2,\ldots,1000$ into C1 (**customer**).

2. Generate the interarrival times $I_1, I_2, \ldots, I_{1000}$ in C2 (**interarr**) i.e. *Poisson arrivals* using Calc→Random Data→Exponential with mean $1/\alpha = 2$ (so $\alpha = 1/2$), and hence calculate the arrival times $A_1, A_2, \ldots, A_{1000}$ in C3 (**arrive**) by $A_n = \sum_{i=1}^{n} I_i$, that is using Calc→Calculator→Partial Sums with **PARS(C2)**.

3. Generate the service times $S_1, S_2, \ldots, S_{1000}$ in C4 (**service**) using Calc→Random Data→Exponential mean $1/\beta = 1.5$ (so $\beta = 2/3$).

4. If the n^{th} customer arrives to find a server free (*idle*), then her/his waiting time W_n is just the service time S_n, and (s)he leaves at time $L_n = A_n + S_n$, otherwise (s)he waits until a server is free. Calculate $A_n + S_n$ using **C3+C4** in C5 (**a+s**). For a single server (SS) queue, (with FIFO='first in, first out' discipline), the server is *idle* and so available when customer n arrives if $A_n \geq L_{n-1}$ (with $L_0 = 0$ if queue is assumed to be empty at time $t = 0$), otherwise $L_n = L_{n-1} + S_n$ i.e the customer has to wait until the previous customer has been served. Thus

$$L_n = \max(A_n, L_{n-1}) + S_n, \quad n = 1,2,\ldots$$

   Calculate this in C6 (**leaveSS**) iteratively by creating an ANSI text file (using say Notepad) called **leave.mtb** with a single line:
   LET C6=RMAX(C3,LAG(C6))+C4
   saving it in your filespace. To start the iteration, type the first value L_1 in row 1 of **leaveSS** which, as the queue is assumed empty at the start, is just the first entry in C5. Now using File→Other files→Run an Exec 1000 times (the total number of customers, or at least as many times are as necessary before the **leave** values are unchanged). Now calculate in C7 (**waitSS**) the waiting time (time in system) $W_n = L_n - A_n$ using **C6-C3**.

Finally, estimate the mean waiting time by the average over all customers (column mean). Are these simulation estimates close to the true value?.

5. The previous system yields analytical results, but now we will change to a queueing system which has no analytical solution! We will vary the *arrival mechanism* to be a system of appointment times with normally distributed error or deviations. Put in C8 (**error**) 1000 values from a standard Normal distribution, and in C9 (**arriveN**):
   C8+2*C1 to simulate actual arrival times based on customer number with a random normal error. This allows a very generous 2 minutes between expected arrival times, the same as for the previous Poisson arrivals, despite the mean service time of 1.5 minutes. It is important that these arrival times are in ascending order for the FIFO discipline, that is we number the customers in order of actual arrival rather than appointment time, so use Data→Sort in C9, by C9, put in C9. Now run through step 4, replacing C6 by C11 and C3 by C9 in **leave.mtb**, for this new single server system, similarly putting the results in C10-C12 (**aN+s, leaveNSS, waitNSS** respectively). What is the change in mean waiting time from the Poisson arrivals model?

6. We will now study the effect of introducing another server. MINITAB cannot calculate queueing statistics as for the single server, so we will estimate the mean waiting time for only the first 16 customers under both the above arrival mechanisms. Obviously introducing another server will reduce mean waiting times, queue sizes and is necessary for equilibrium i.e. to achieve stationarity if $\alpha \geq \beta$, (For k servers, Poisson arrivals and exponential service times, the condition for equilibrium is $\rho = \alpha/\beta < k$.) The queueing discipline will be single queue: head of queue (FIFO) gets served as soon as either server is free, which can easily be shown to give the shortest mean waiting time (*ceteris paribus*). To make the comparison of arrival mechanisms more interesting we will halve the mean interarrival time to 1, so that $\alpha = 1$ (keeping β and consequently C4 the same). Regenerate C2 with mean interarrival time 1, then recalculate C3 and C5 as before, (C5 remember is the leaving time for that customer if they are served immediately ie a server is idle on their arrival, and the waiting time is then just the service time C4). For Normally distributed arrival times on an appointments system, recalculate C9 as **C8+C1+2**, and then reorder using Data→Sort as before, then recalculate C10.

7. Name columns C13-C16 **S1in**, **S1out**, **S2in**, **S2out**. These refer to the times at which each server starts and completes service, both being idle at the beginning.

8. Trace through the development of the system for the first 16 customers only for each arrival pattern, noting the waiting time (queueing time + service time) in C17 and C18 for each customer. See Notes 6 for an example. Compare the arrival patterns by average waiting time (over the first 16 customers). Which is best?

9. Using File→Save Project, save your output for Cwk4.