Table 4.6 *(continued)*

Methods	*Description*
`Initialization`	Initialization method
`ProcessArrival`	Event method that executes the arrival event
`ProcessDeparture`	Event method that executes the departure event
`ReportGeneration`	Report generator

The entry point of the program and the location of the control logic is through class `Sim`, shown in Figure 4.3. Variables of classes `EventList` and `Queue` are declared. As these classes are all useful for programs other than `Sim`, their declarations are given in other files, per Java rules. A variable of the Java built-in class `Random` is also declared; instances of this class provided random-number streams. The main method controls the overall flow of the event-scheduling/time-advance algorithm.

```
class Sim {

// Class Sim variables
public static double Clock, MeanInterArrivalTime, MeanServiceTime,
        SIGMA, LastEventTime, TotalBusy, MaxQueueLength, SumResponseTime;
public static long NumberOfCustomers, QueueLength, NumberInService,
        TotalCustomers, NumberOfDepartures, LongService;

public final static int arrival = 1;
public final static int departure = 2;

public static EventList FutureEventList;
public static Queue Customers;
public static Random stream;

public static void main(String argv[]) {

  MeanInterArrivalTime = 4.5; MeanServiceTime = 3.2;
  SIGMA                = 0.6; TotalCustomers  = 1000;
  long seed            = Long.parseLong(argv[0]);
  stream = new Random(seed);              // initialize rng stream
  FutureEventList = new EventList();
  Customers = new Queue();

  Initialization();

  // Loop until first "TotalCustomers" have departed
  while(NumberOfDepartures < TotalCustomers ) {
    Event evt = (Event)FutureEventList.getMin();     // get imminent event
    FutureEventList.dequeue();                       // be rid of it
    Clock = evt.get_time();                          // advance in time
    if( evt.get_type() == arrival ) ProcessArrival(evt);
    else ProcessDeparture(evt);
    }
  ReportGeneration();
}
```

Figure 4.3 Java main program for the single-server queue simulation.

The main program method first gives values to variables describing model parameters; it creates instances of the random-number generator, event list, and customer queue; and then it calls method `Initialization` to initialize other variables, such as the statistics-gathering variables. Control then enters a loop which is exited only after `TotalCustomers` customers have received service. Inside the loop, a copy of the imminent event is obtained by calling the `getMin` method of the priority queue, and then that event is removed from the event list by a call to `dequeue`. The global simulation time `Clock` is set to the time-stamp contained in the imminent event, and then either `ProcessArrival` or `ProcessDeparture` is called, depending on the type of the event. When the simulation is finally over, a call is made to method `ReportGeneration` to create and print out the final report.

A listing for the `Sim` class method `Initialization` is given in Figure 4.4. The simulation clock, system state, and other variables are initialized. Note that the first arrival event is created by generating a local `Event` variable whose constructor accepts the event's type and time. The event time-stamp is generated randomly by a call to `Sim` class method `exponential` and is passed to the random-number stream to use with the mean of the exponential distribution from which to sample. The event is inserted into the future event list by calling method `enqueue`. This logic assumes that the system is empty at simulated time `Clock` = 0, so that no departure can be scheduled. It is straightforward to modify the code to accommodate alternative starting conditions by adding events to `FutureEventList` and `Customers` as needed.

Figure 4.5 gives a listing of `Sim` class method `ProcessArrival`, which is called to process each arrival event. The basic logic of the arrival event for a single-server queue was given in Figure 3.5 (where LQ corresponds to `QueueLength` and LS corresponds to `NumberInService`). First, the new arrival is added to the queue `Customers` of customers in the system. Next, if the server is idle (`NumberInService` = = 0) then the new customer is to go immediately into service, so `Sim` class method `ScheduleDeparture` is called to do that scheduling. An arrival to an idle queue does not update the cumulative statistics, except possibly the maximum queue length. An arrival to a busy queue does *not* cause the scheduling of a departure, but does increase the total busy time by the amount of simulation time between the current event and the one immediately preceding it (because, if the server is busy now, it had to have had at least one customer in service by the end of processing the previous event). In either case, a new arrival is responsible for scheduling the next arrival, one random interarrival time into the future. An arrival event is created with simulation time equal to the current `Clock` value plus an exponential increment, that event is inserted into the future event list, the variable `LastEventTime` recording the time of the last event processed is set to the current time, and control is returned to the `main` method of class `Sim`.

```
public static void Initialization()    {
  Clock = 0.0;
  QueueLength = 0;
  NumberInService = 0;
  LastEventTime = 0.0;
  TotalBusy = 0 ;
  MaxQueueLength = 0;
  SumResponseTime = 0;
  NumberOfDepartures = 0;
  LongService = 0;

  // create first arrival event
  Event evt =
   new Event(arrival, exponential( stream, MeanInterArrivalTime));
  FutureEventList.enqueue( evt );
}
```

Figure 4.4 Java initialization method for the single-server queue simulation.

```
public static void ProcessArrival(Event evt)  {
  Customers.enqueue(evt);
  QueueLength++;
  // if the server is idle, fetch the event, do statistics
  // and put into service
  if( NumberInService == 0) ScheduleDeparture();
  else TotalBusy += (Clock - LastEventTime);    // server is busy

  // adjust max queue length statistics
  if (MaxQueueLength < QueueLength) MaxQueueLength = QueueLength;

  // schedule the next arrival
  Event next_arrival =
   new Event(arrival, Clock+exponential(stream,MeanInterArrivalTime));
  FutureEventList.enqueue( next_arrival );
  LastEventTime = Clock;
}
```

Figure 4.5 Java arrival event method for the single-server queue simulation.

`Sim` class method `ProcessDeparture`, which executes the departure event, is listed in Figure 4.6, as is method `ScheduleDeparture`. A flowchart for the logic of the departure event was given in Figure 3.6. After removing the event from the queue of all customers, the number in service is examined. If there are customers waiting, then the departure of the next one to enter service is scheduled. Then, cumulative statistics recording the sum of all response times, sum of busy time, number of customers who used more than 4 minutes of service time, and number of departures are updated. (Note that the maximum queue length cannot change in value when a departure occurs.) Notice that customers are removed from `Customers` in

```
public static void ScheduleDeparture()  {
  double ServiceTime;
  // get the job at the head of the queue
  while (( ServiceTime = normal(stream,MeanServiceTime, SIGMA)) < 0 );
  Event depart = new Event(departure,Clock+ServiceTime);
  FutureEventList.enqueue( depart );
  NumberInService = 1;
  QueueLength--;
  }

public static void ProcessDeparture(Event e) {
  // get the customer description
  Event finished = (Event) Customers.dequeue();
  // if there are customers in the queue then schedule
  // the departure of the next one
   if( QueueLength > 0 ) ScheduleDeparture();
   else NumberInService = 0;
   // measure the response time and add to the sum
   double response = (Clock - finished.get_time());
   SumResponseTime += response;
   if( response > 4.0 ) LongService++; // record long service
   TotalBusy += (Clock - LastEventTime );
   NumberOfDepartures++;
   LastEventTime = Clock;
  }
```

Figure 4.6 Java departure event method for the single-server queue simulation.

FIFO order; hence, the response time `response` of the departing customer can be computed by subtracting the arrival time of the job leaving service (obtained from the copy of the arrival event removed from the `Customers` queue) from the current simulation time. After the incrementing of the total number of departures and the saving of the time of this event, control is returned to the main program.

Figure 4.6 also gives the logic of method `ScheduleDeparture`, called by both `ProcessArrival` and `ProcessDeparture` to put the next customer into service. The `Sim` class method `normal`, which generates normally distributed service times, is called until it produces a nonnegative sample. A new event with type `departure` is created, with event time equal to the current simulation time plus the service time just sampled. That event is pushed onto `FutureEventList`, the number in service is set to one, and the number waiting (`QueueLength`) is decremented to reflect the fact that the customer entering service is waiting no longer.

The report generator, `Sim` class method `ReportGeneration`, is listed in Figure 4.7. The summary statistics, RHO, AVGR, and PC4, are computed by the formulas in Table 4.6; then the input parameters are printed, followed by the summary statistics. It is a good idea to print the input parameters at the end of the simulation, in order to verify that their values are correct and that these values have not been inadvertently changed.

Figure 4.8 provides a listing of `Sim` class methods exponential and `normal`, used to generate random variates. Both of these functions call method `nextDouble`, which is defined for the built-in Java `Random` class generates a random number uniformly distributed on the (0,1) interval. We use `Random` here for simplicity of explanation; superior random-number generators can be built by hand, as described in Chapter 7.

```
public static void ReportGeneration() {
double RHO  = TotalBusy/Clock;
double AVGR = SumResponseTime/TotalCustomers;
double PC4  = ((double)LongService)/TotalCustomers;


System.out.print( "SINGLE SERVER QUEUE SIMULATION ");
System.out.println( "- GROCERY STORE CHECKOUT COUNTER ");
System.out.println( "\tMEAN INTERARRIVAL TIME                                  "
   + MeanInterArrivalTime );
System.out.println( "\tMEAN SERVICE TIME                                       "
   + MeanServiceTime );
System.out.println( "\tSTANDARD DEVIATION OF SERVICE TIMES                     "
   + SIGMA );
System.out.println( "\tNUMBER OF CUSTOMERS SERVED                              "
   + TotalCustomers );
System.out.println();
System.out.println( "\tSERVER UTILIZATION                                      "
   + RHO );
System.out.println( "\tMAXIMUM LINE LENGTH                                     "
   + MaxQueueLength );
System.out.println( "\tAVERAGE RESPONSE TIME                                   "
   + AVGR + "  MINUTES" );
System.out.println( "\tPROPORTION WHO SPEND FOUR ");
System.out.println( "\t MINUTES OR MORE IN SYSTEM                              "
   + PC4 );
System.out.println( "\tSIMULATION RUNLENGTH                                    "
   + Clock + " MINUTES" );
System.out.println( "\tNUMBER OF DEPARTURES                                    "
   + TotalCustomers );
 }
```

Figure 4.7 Java report generator for the single-server queue simulation.

```
public static double exponential(Random rng, double mean) {
 return -mean*Math.log( rng.nextDouble() );
}

public static double SaveNormal;
public static int NumNormals = 0;
public static final double PI = 3.1415927 ;

public static double normal(Random rng, double mean, double sigma) {
        double ReturnNormal;
        // should we generate two normals?
        if(NumNormals == 0 ) {
          double r1 = rng.nextDouble();
          double r2 = rng.nextDouble();
          ReturnNormal = Math.sqrt(-2*Math.log(r1))*Math.cos(2*PI*r2);
          SaveNormal   = Math.sqrt(-2*Math.log(r1))*Math.sin(2*PI*r2);
          NumNormals = 1;
        } else {
          NumNormals = 0;
          ReturnNormal = SaveNormal;
        }
        return ReturnNormal*sigma + mean ;
}
```

Figure 4.8 Random-variate generators for the single-server queue simulation.

The techniques for generating exponentially and normally distributed random variates, discussed in Chapter 8, are based on first generating a $U(0,1)$ random number. For further explanation, the reader is referred to Chapters 7 and 8.

The output from the grocery-checkout-counter simulation is shown in Figure 4.9. It should be emphasized that the output statistics are estimates that contain random error. The values shown are influenced by the particular random numbers that happened to have been used, by the initial conditions at time 0, and by the run length (in this case, 1000 departures). Methods for estimating the standard error of such estimates are discussed in Chapter 11.

In some simulations, it is desired to stop the simulation after a fixed length of time, say TE = 12 hours = 720 minutes. In this case, an additional event type, `stop` event, is defined and is scheduled to occur by scheduling a stop event as part of simulation initialization. When the stopping event does occur, the cumulative

```
SINGLE SERVER QUEUE SIMULATION - GROCERY STORE CHECKOUT COUNTER
        MEAN INTERARRIVAL TIME                          4.5
        MEAN SERVICE TIME                               3.2
        STANDARD DEVIATION OF SERVICE TIMES             0.6
        NUMBER OF CUSTOMERS SERVED                      1000

        SERVER UTILIZATION                              0.671
        MAXIMUM LINE LENGTH                             9.0
        AVERAGE RESPONSE TIME                           6.375 MINUTES
        PROPORTION WHO SPEND FOUR
         MINUTES OR MORE IN SYSTEM                      0.604
        SIMULATION RUNLENGTH                            4728.936 MINUTES
        NUMBER OF DEPARTURES                            1000
```

Figure 4.9 Output from the Java single-server queue simulation.

statistics will be updated and the report generator called. The main program and method `Initialization` will require minor changes. Exercise 1 asks the reader to make these changes. Exercise 2 considers balking of customers.

4.5 SIMULATION IN GPSS

GPSS is a highly structured, special-purpose simulation programming language based on the process-interaction approach and oriented toward queueing systems. A block diagram provides a convenient way to describe the system being simulated. There are over 40 standard blocks in GPSS. Entities called transactions may be viewed as flowing through the block diagram. Blocks represent events, delays, and other actions that affect transaction flow. Thus, GPSS can be used to model any situation where transactions (entities, customers, units of traffic) are flowing through a system (e.g., a network of queues, with the queues preceding scarce resources). The block diagram is converted to block statements, control statements are added, and the result is a GPSS model.

The first version of GPSS was released by IBM in 1961. It was the first process-interaction simulation language and became popular; it has been implemented anew and improved by many parties since 1961, with GPSS/H being the most widely used version in use today. Example 4.3 is based on GPSS/H.

GPSS/H is a product of Wolverine Software Corporation, Annandale, VA (Banks, Carson, and Sy, 1995; Henriksen, 1999). It is a flexible, yet powerful tool for simulation. Unlike the original IBM implementation, GPSS/H includes built-in file and screen I/O, use of an arithmetic expression as a block operand, an interactive debugger, faster execution, expanded control statements, ordinary variables and arrays, a floating-point clock, built-in math functions, and built-in random-variate generators.

The animator for GPSS/H is Proof Animation™, another product of Wolverine Software Corporation (Henriksen, 1999). Proof Animation provides a 2-D animation, usually based on a scale drawing. It can run in postprocessed mode (after the simulation has finished running) or concurrently. In postprocessed mode, the animation is driven by two files: the layout file for the static background, and a trace file that contains commands to make objects move and produce other dynamic events. It can work with any simulation package that can write the ASCII trace file. Alternately, it can run concurrently with the simulation by sending the trace file commands as messages, or it can be controlled directly by using its DLL (dynamic link library) version.

Example 4.3: Single-Server Queue Simulation in GPSS/H

Figure 4.10 exhibits the block diagram and Figure 4.11 the GPSS program for the grocery-store checkout-counter model described in Example 4.2. Note that the program (Figure 4.11) is a translation of the block diagram together with additional definition and control statements.

In Figure 4.10, the GENERATE block represents the arrival event, with the interarrival times specified by RVEXPO(1,*&IAT*). RVEXPO stands for "random variable, exponentially distributed," the 1 indicates the random-number stream to use, and *&IAT* indicates that the mean time for the exponential distribution comes from a so-called ampervariable *&IAT*. Ampervariable names begin with the "&" character; Wolverine added ampervariables to GPSS because the original IBM implementation had limited support for ordinary global variables, with no user freedom for naming them. (In the discussion that follows, all nonreserved words are shown in italics.)

The next block is a QUEUE with a queue named *SYSTIME*. It should be noted that the QUEUE block is not needed for queues or waiting lines to form in GPSS. The true purpose of the QUEUE block is to work in conjunction with the DEPART block to collect data on queues or any other subsystem. In Example 4.3,

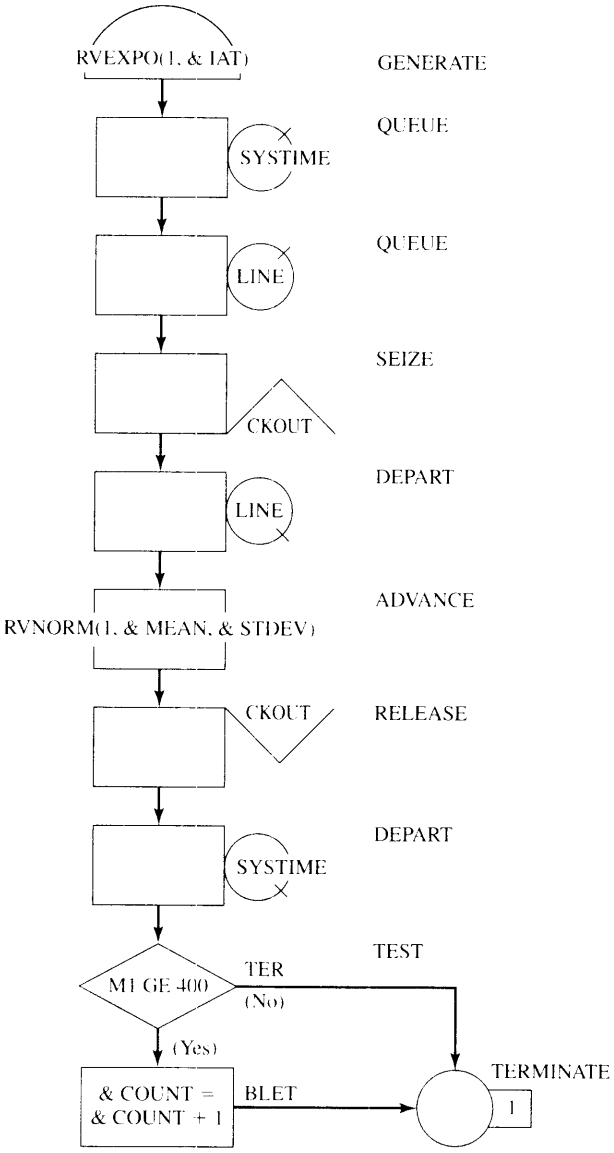


Figure 4.10 GPSS block diagram for the single-server queue simulation.

we want to measure the system response time—that is, the time a transaction spends in the system. Placing a QUEUE block at the point that transactions enter the system and placing the counterpart of the QUEUE block, the DEPART block, at the point that the transactions complete their processing causes the response times to be collected automatically. The purpose of the DEPART block is to signal the end of data collection for an individual transaction. The QUEUE and DEPART block combination is not necessary for queues to be modeled, but rather is used for statistical data collection.

The next QUEUE block (with name *LINE*) begins data collection for the waiting line before the cashier. The customers may or may not have to wait for the cashier. Upon arrival to an idle checkout counter, or after

```
      SIMULATE
*
*     Define Ampervariables
*
      INTEGER     &LIMIT
      REAL        &IAT,&MEAN,&STDEV,&COUNT
      LET         &IAT=4.5
      LET         &MEAN=3.2
      LET         &STDEV=.6
      LET         &LIMIT=1000
*
*     Write Input Data to File
*
      PUTPIC      FILE=OUT,LINES=5,(&IAT,&MEAN,&STDEV,&LIMIT)
      Mean interarrival time                  **.** minutes
      Mean service time                       **.** minutes
      Standard deviation of service time      **.** minutes
      Number of customers to be served        *****
*
*     GPSS/H Block Section
*
      GENERATE    RVEXPO(1,&IAT)  Exponential arrivals
      QUEUE       SYSTIME         Begin response time data collection
      QUEUE       LINE            Customer joins waiting line
      SEIZE       CHECKOUT        Begin checkout at cash register
      DEPART      LINE            Customer starting service leaves queue
      ADVANCE     RVNORM(1,&MEAN,&STDEV)   Customer's service time
      RELEASE     CHECKOUT        Customer leaves checkout area
      DEPART      SYSTIME         End response time data collection
      TEST GE     M1,4,TER        Is response time GE 4 minutes?
      BLET        &COUNT=&COUNT+1 If so, add 1 to counter
TER   TERMINATE   1
*
      START       &LIMIT          Simulate for required number
*
*     Write Customized Output Data to File
*
      PUTPIC      FILE=OUT,LINES=7,(FR(CHECKOUT)/1000,QM(LINE),_
      QT(SYSTIME),&COUNT/N(TER),AC1,N(TER))
      Server utilization                    .***
      Maximum line length                   **
      Average response time                 **.**    minutes
      Proportion who spend four minutes      .***
           or more in the system
      Simulation runlength                   ****.** minutes
      Number of departures                   ****
*
      END
```

Figure 4.11 GPSS/H program for the single-server queue simulation.

advancing to the head of the waiting line, a customer captures the cashier, as represented by the SEIZE block with the resource named *CHECKOUT*. Once the transaction representing a customer captures the cashier represented by the resource CHECKOUT, the data collection for the waiting-line statistics ends, as represented by the DEPART block for the queue named LINE. The transaction's service time at the cashier is

represented by an ADVANCE block. RVNORM indicates "random variable, normally distributed." Again, random-number stream 1 is being used, the mean time for the normal distribution is given by ampervariable *&MEAN*, and its standard deviation is given by ampervariable *&STDEV*. Next, the customer gives up the use of the facility *CHECKOUT* with a RELEASE block. The end of the data collection for response times is indicated by the DEPART block for the queue *SYSTIME*.

Next, there is a TEST block that checks to see whether the time in the system, M1, is greater than or equal to 4 minutes. (Note that M1 is a reserved word in GPSS/H; it automatically tracks transaction total time in system.) In GPSS/H, the maxim is "if true, pass through." Thus, if the customer has been in the system four minutes or longer, the next BLET block (for block LET) adds one to the counter *&COUNT*. If not true, the escape route is to the block labeled *TER*. That label appears before the TERMINATE block whose purpose is the removal of the transaction from the system. The TERMINATE block has a value "1" indicating that one more transaction is added toward the limiting value (or "transactions to go").

The control statements in this example are all of those lines in Figure 4.11 that precede or follow the block section. (There are eleven blocks in the model from the GENERATE block to the TERMINATE block.) The control statements that begin with an "*" are comments, some of which are used for spacing purposes. The control statement SIMULATE tells GPSS/H to conduct a simulation; if it is omitted, GPSS/H compiles the model and checks for errors only. The ampervariables are defined as integer or real by control statements INTEGER and REAL. It seems that the ampervariable *&COUNT* should be defined as an integer; however, it will be divided later by a real value. If it is integer, the result of an integer divided by a real value is truncation, and that is not desired in this case. The four assignment statements (LET) provide data for the simulation. These four values could have been placed directly in the program; however, the preferred practice is to place them in ampervariables at the top of the program so that changes can be made more easily or the model can be modified to read them from a data file.

To ensure that the model data is correct, and for the purpose of managing different scenarios simulated, it is good practice to echo the input data. This is accomplished with a PUTPIC (for "put picture") control statement. The five lines following PUTPIC provide formatting information, with the asterisks being markers (called picture formatting) in which the values of the four ampervariables replace the asterisks when PUTPIC is executed. Thus, "**.**" indicates a value that may have two digits following the decimal point and up to two before it.

The START control statement controls simulation execution. It starts the simulation, sets up a "termination-to-go" counter with initial value its operand (*&LIMIT*), and controls the length of the simulation.

After the simulation completes, a second PUTPIC control statement is used to write the desired output data to the same file *OUT*. The printed statistics are all gathered automatically by GPSS. The first output in the parenthesized list is the server utilization. FR(*CHECKOUT*)/1000 indicates that the fractional utilization of the facility *CHECKOUT* is printed. Because FR(*CHECKOUT*) is in parts per thousand, the denominator is provided to compute fractional utilization. QM(*LINE*) is the maximum value in the queue *LINE* during the simulation. QT(*SYSTIME*) is the average time in the queue *SYSTIME*. *&COUNT*/N(*TER*) is the number of customers who had a response time of four or more minutes divided by the number of customers that went through the block with label *TER*, or N(*TER*). AC1 is the clock time, whose last value gives the length of the simulation.

The contents of the custom output file *OUT* are shown in Figure 4.12. The standard GPSS/H output file is displayed in Figure 4.13. Although much of the same data shown in the file *OUT* can be found in the standard GPSS/H output, the custom file is more compact and uses the language of the problem rather than GPSS jargon. There are many other reasons that customized output files are useful. For example, if 50 replications of the model are to be made and the lowest, highest, and average value of a response are desired, this can be accomplished by using control statements, with the results in a very compact form, rather than extracting the desired values from 50 standard output files.

```
Mean interarrival time                    4.50 minutes
Mean service time                         3.20 minutes
Standard deviation of service time        3.60 minutes
Number of customers to be served          1000

Server utilization                        0.676
Maximum line length                       7
Average response time                     6.33 minutes
Proportion who spend four minutes         0.646
     or more in the system
Simulation runlength                      4767.27 minutes
Number of departures                      1000
```

Figure 4.12 Customized GPSS/H output report for the single-server queue simulation.

```
RELATIVE CLOCK: 4767.2740 ABSOLUTE CLOCK: 4767.2740

BLOCK CURRENT      TOTAL BLOCK CURRENT      TOTAL
1                   1003 TER                1000
2                   1003
3           3       1003
4                   1000
5                   1000
6                   1000
7                   1000
8                   1000
9                   1000
10                   646

          --AVG-UTIL-DURING--
FACILITY   TOTAL   AVAIL   UNAVL    ENTRIES    AVERAGE CURRENT  PERCENT  SEIZING  PREEMPTING
            TIME    TIME    TIME              TIME/XACT  STATUS   AVAIL     XACT       XACT
CHECKOUT   0.676                       1000      3.224   AVAIL

  QUEUE  MAXIMUM   AVERAGE   TOTAL     ZERO    PERCENT   AVERAGE   $AVERAGE  QTABLE   CURRENT
         CONTENTS  CONTENTS  ENTRIES   ENTRIES   ZEROS   TIME/UNIT TIME/UNIT NUMBER   CONTENTS
SYSTIME        8     1.331    1003        0               6.325     6.235                3
   LINE        7     0.655    1003      334     33.3      3.111     4.665                3

RANDOM   ANTITHETIC    INITIAL     CURRENT    SAMPLE    CHI-SQUARE
STREAM     VARIATES    POSITION    POSITION    COUNT    UNIFORMITY
     1          OFF      100000      103004     3004      0.83
```

Figure 4.13 Standard GPSS/H output report for the single-server queue simulation.

4.6 SIMULATION IN SSF

The Scalable Simulation Framework (SSF) is an Application Program Interface (API) that describes a set of capabilities for object-oriented, process-view simulation. The API is sparse and was designed to allow implementations to achieve high performance (e.g. on parallel computers). SSF APIs exist for both C++ and in Java.

and implementations exist in both languages. SSF has a wide user base—particularly in network simulation by using the add-on framework SSFNet (`www.ssfnet.org`). Our chapter on network simulation uses SSFNet.

The SSF API defines five base classes. `process` is a class that implements threads of control; the `action` method of a derived class contains the execution body of the thread. The `Entity` class is used to describe simulation objects. It contains state variables, processes, and communication endpoints. The `inChannel` and `outChannel` classes are communication endpoints. The `Event` class defines messages sent between entities. One model entity communicates with another by "writing" an `Event` into an `outChannel`; at some later time, it is available at one or more `inChannels`. A `process` that expects input on an `inChannel` can suspend, waiting for an event on it. These points, and others, will be elaborated upon as we work through an SSF implementation of the single-server queue.

Source code given in Figure 4.14 expresses the logic of arrival generation in SSF for the single-server queue example. The example is built on two SSF processes. One of these generates jobs and adds them to the system; the other services the enqueued jobs. Class `SSQueue` is a class that contains the whole simulation experiment. It uses the auxiliary classes `Random` (for random-number generation) and `Queue` (to implement FIFO queueing of general objects). `SSQueue` defines experimental constants ("public static final" types) and contains SSF communication endpoints `out` and `in`, through which the two processes communicate. `SSQueue` also defines an inner class `arrival`, which stores the identity and arrival time of each job.

Class `Arrivals` is an SSF process. Its constructor stores the identity of the entity that owns it, and creates a random-number generator that is initialized with the seed passed to it. For all but the initial call, method `action` generates and enqueues a new arrival, then blocks (via SSF method `waitFor`) for an inter-arrival time; on the first call, it by-passes the job-generation step and blocks for an initial interarrival time. The call to `waitFor` highlights details needing explanation. An `SSQueue` object calls the `Arrival` constructor and is saved as the "owner." This class contains an auxiliary method `exponential`, which samples an exponential random variable with specified mean by using a specified random-number stream. It also contains methods `d2t` and `t2d` that translate between a discrete "tick"-based integer clock and a double-precision floating-point representation. In the `waitFor` call, we use the same code seen earlier to sample the exponential in double-precision format, but then use `d2t` to convert it into the simulator's integer clock format. The specific conversion factor is listed as a `SSQueue` constant, 10^9 ticks per unit time.

SSF interprocess communication is used sparingly in this example. Because service is nonpreemptive, when a job's service completes, the process providing service can examine the list of waiting customers (in variable `owner.Waiting`) to see whether it needs to give service to another customer. Thus, the only time the server process needs to be told that there is a job waiting is when a job arrives to an empty system. This is reflected in `Arrivals.action` by use of its owner's `out` channel.

A last point of interest is that `Arrivals` is, in SSF terminology, a "simple" process. This means that every statement in `action` that might suspend the process would be the last statement executed under normal execution semantics. The `Arrivals` class tells SSF that it is simple by overriding a default method `isSimple` to return the value true, rather than the default value (false). The key reason for using simple processes is performance—they require that no state be saved, only the condition under which the process ought to be reanimated. And, when it is reanimated, it starts executing at the first line of `action`.

Figure 4.15 illustrates the code for the `Server` process. Like process `Arrival`, its constructor is called by an instance of `SSQueue` and is given the identity of that instance and a random-number seed. Like `Arrival`, it is a simple process. It maintains state variable `in_service` to remember the specifics of a job in service and state variable `service_time` to remember the value of the service time

```
// SSF MODEL OF JOB ARRIVAL PROCESS
class SSQueue extends Entity {

 private static Random rng;
 public static final double MeanServiceTime = 3.2;
 public static final double SIGMA = 0.6;
 public static final double MeanInterarrivalTime = 4.5;
 public static final long ticksPerUnitTime = 1000000000;
 public long generated=0;
 public Queue Waiting;
 outChannel out;
 inChannel in;

 public static long TotalCustomers=0, MaxQueueLength=0, TotalServiceTime=0;
 public static long LongResponse=0, SumResponseTime=0, jobStart;

 class arrival {
  long id, arrival_time;
  public arrival(long num, long a) { id=num; arrival_time = a; }
 }

class Arrivals extends process {
   private Random rng;
   private SSQueue owner;
   public Arrivals (SSQueue _owner, long seed) {
        super(_owner); owner = _owner;
        rng = new Random(seed);
   }
   public boolean isSimple() { return true; }
   public void action() {
      if ( generated++ > 0 ) {
        // put a new Customer on the queue with the present arrival time
        int Size = owner.Waiting.numElements();
        owner.Waiting.enqueue( new arrival(generated, now()));
        if( Size == 0) owner.out.write( new Event() ); // signal start of burst
      }
      waitFor(owner.d2t( owner.exponential(rng, owner.MeanInterarrivalTime)) );
   }
 }
}
```

Figure 4.14 SSF Model of Job-Arrival Process.

sampled for the job in service. When the SSF kernel calls `action`, either a job has completed service, or the `Arrival` process has just signaled `Server` though the `inChannel`. We distinguish the cases by looking at variable `in_service`, which will be nonnull if a job is in service, just now completed. In this case, some statistics are updated. After this task is done, a test is made for customers waiting for service. The first waiting customer is dequeued from the waiting list and is copied into the `in_service` variable; the process then samples a service time and suspends through a `waitFor` call. If no customer was waiting, the process suspends on a `waitOn` statement until an event from the `Arrival` process awakens it.

SSF bridges the gap between models developed in pure Java and models developed in languages specifically designed for simulation. It provides the flexibility offered by a general-programming language, yet has essential support for simulation.

```
// SSF MODEL OF SINGLE SERVER QUEUE : ACCEPTING JOBS
class Server extends process {
 private Random rng;
 private SSQueue owner ;
 private arrival in_service;
 private long service_time;

 public Server(SSQueue _owner, long seed) {
       super(_owner);
       owner = _owner;
       rng = new Random(seed);
 }
 public boolean isSimple() { return true; }

 public void action() {
   //  executes due to being idle and getting a job, or by service time expiration.
   //  if there is a job awaiting service, take it out of the queue
   //  sample a service time, do statistics, and wait for the service epoch

   // if in_service is not null, we entered because of a job completion
   if( in_service != null ) {
       owner.TotalServiceTime += service_time;
       long in_system = (now() -in_service.arrival_time);
       owner.SumResponseTime += in_system;
       if( owner.t2d(in_system) > 4.0 ) owner.LongResponse++;
       in_service = null;
       if( owner.MaxQueueLength < owner.Waiting.numElements() + 1 )
            owner.MaxQueueLength = owner.Waiting.numElements() + 1;
       owner.TotalCustomers++;
   }
   if( owner.Waiting.numElements() > 0 ) {
       in_service = (arrival)owner.Waiting.dequeue();
       service_time = -1;
       while ( service_time < 0.0 )
         service_time = owner.d2t(owner.normal( rng, owner.MeanServiceTime, owner.SIGMA));
                                                                    // model service time
       waitFor( service_time );
   } else {
       waitOn( owner.in ); // we await a wake-up call
   }
 }
}
```

Figure 4.15 SSF Model of Single-Server Queue : Server.

4.7 SIMULATION SOFTWARE

All the simulation packages described in later subsections run on a PC under Microsoft Windows 2000 or XP. Although in terms of specifics the packages all differ, generally they have many things in common.

Common characteristics include a graphical user interface, animation, and automatically collected outputs to measure system performance. In virtually all packages, simulation results may be displayed in tabular or graphical form in standard reports and interactively while running a simulation. Outputs from different scenarios can be compared graphically or in tabular form. Most provide statistical analyses that include confidence intervals for performance measures and comparisons, plus a variety of other analysis methods. Some of the statistical-analysis modules are described in Section 4.8.

All the packages described here take the process-interaction worldview. A few also allow event-scheduling models and mixed discrete–continuous models. For animation, some emphasize scale drawings in 2-D or 3-D; others emphasize iconic-type animations based on schematic drawings or process-flow diagrams. A few offer both scale drawing and schematic-type animations. Almost all offer dynamic business graphing in the form of time lines, bar charts, and pie charts.

In addition to the information contained in this chapter, the websites given below can be investigated:

Arena
`www.arenasimulation.com/`
AutoMod
`www.automod.com`
Delmia/QUEST
`www.delmia.com` and `www.3ds.com`
Extend
`www.imaginethatinc.com/`
Flexsim
`www.flexsim.com/`
Micro Saint
`www.maad.com`
ProModel
`www.promodel.com/`
SIMUL8
`www.simul8.com/`
WITNESS
`www.witness-for-simulation.com/`

4.7.1 Arena

Arena Basic, Standard, and Professional Editions are offered by Systems Modeling Corporation [Bapat and Sturrock, 2003]. Arena can be used for simulating discrete and continuous systems. A recent addition to the Arena family of products is OptQuest for Arena, an optimization software package (discussed in Section 4.8.2.)

The Arena Basic Edition is targeted at modeling business processes and other systems in support of high-level analysis needs. It represents process dynamics in a hierarchical flowchart and stores system information in data spreadsheets. It has built-in activity-based costing and is closely integrated with the flowcharting software Visio.

The Arena Standard Edition is designed for more detailed models of discrete and continuous systems. First released in 1993, Arena employs an object-based design for entirely graphical model development. Simulation models are built from graphical objects called modules to define system logic and such physical components as machines, operators, and clerks. Modules are represented by icons plus associated data entered in a dialog window. These icons are connected to represent entity flow. Modules are organized into collections called templates. The Arena template is the core collection of modules providing general-purpose features for modeling all types of applications. In addition to standard features, such as resources, queues, process logic, and system data, the Arena template includes modules focused on specific aspects of manufacturing and material-handling systems. Arena SE can also be used to model combined discrete/continuous systems, such as pharmaceutical and chemical production, through its built-in continuous-modeling capabilities.

The Arena Professional Edition enhances Arena SE with the capability to craft custom simulation objects that mirror components of the real system, including terminology, process logic, data, performance metrics, and animation. The Arena family also includes products designed specifically to model call centers and high-speed production lines, namely Arena Contact Center and Arena Packaging.

At the heart of Arena is the SIMAN simulation language. For animating simulation models, Arena's core modeling constructs are accompanied by standard graphics for showing queues, resource status, and entity flow. Arena's 2-D animations are created by using Arena's built-in drawing tools and by incorporating clip art, AutoCAD, Visio, and other graphics.

Arena's Input Analyzer automates the process of selecting the proper distribution and its parameters for representing existing data, such as process and interarrival times. The Output Analyzer and Process Analyzer (discussed in Section 4.8.2) automate comparison of different design alternatives.

4.7.2 AutoMod

The AutoMod Product Suite is offered by Brooks Automation [Rohrer, 2003]. It includes the AutoMod simulation package, AutoStat for experimentation and analysis, and AutoView for making AVI movies of the built-in 3-D animation. The main focus of the AutoMod simulation product is manufacturing and material-handling systems. AutoMod's strength is in detailed, large models used for planning, operational decision support, and control-systems testing.

AutoMod has built-in templates for most common material-handling systems, including vehicle systems, conveyors, automated storage and retrieval systems, bridge cranes, power and free conveyors, and kinematics for robotics. With its Tanks and Pipes module, it also supports continuous modeling of fluid and bulk-material flow.

The pathmover vehicle system can be used to model lift trucks, humans walking or pushing carts, automated guided vehicles, trucks, and cars. All the movement templates are based on a 3-D scale drawing (drawn or imported from CAD as 2-D or 3-D). All the components of a template are highly parameterized. For example, the conveyor template contains conveyor sections, stations for load induction or removal, motors, and photo-eyes. Sections are defined by length, width, speed, acceleration, and type (accumulating or nonaccumulating), plus other specialized parameters. Photo-eyes have blocked and cleared timeouts that facilitate modeling of detailed conveyor logic.

In addition to the material-handling templates, AutoMod contains a full simulation programming language. Its 3-D animation can be viewed from any angle or perspective in real time. The user can freely zoom, pan, or rotate the 3-D world.

An AutoMod model consists of one or more systems. A system can be either a process system, in which flow and control logic are defined, or a movement system based on one of the material-handling templates. A model may contain any number of systems, which can be saved and reused as objects in other models. Processes can contain complex logic to control the flow of either manufacturing materials or control messages, to contend for resources, or to wait for user-specified times. Loads can move between processes with or without using movement systems.

In the AutoMod worldview, loads (products, parts, etc.) move from process to process and compete for resources (equipment, operators, vehicles, and queues). The load is the active entity, executing action statements in each process. To move between processes, loads may use a conveyor or vehicle in a movement system.

AutoStat, described in Section 4.8.2, works with AutoMod models to provide a complete environment for the user to define scenarios, conduct experimentation, and perform analyses. It offers optimization based on an evolutionary strategies algorithm.

4.7.3 Extend

The Extend family of products is offered by Imagine That, Inc. [Krahl, 2003]. Extend OR, Industry, and Suite are used for simulating discrete and mixed discrete–continuous systems; Extend CP is for continuous modeling only. Extend combines a block-diagram approach to model-building with a development environment for creating new blocks.

Each Extend block has an icon and encapsulates code, parameters, user interface, animation, and online help. Extend includes a large set of elemental blocks; libraries of blocks for specific application areas, such as manufacturing, business processes, and high-speed processes, are also available. Third-party developers have created Extend libraries for vertical market applications, including supply-chain dynamics, reliability engineering, and pulp and paper processing.

Models are built by placing and connecting blocks and entering the parameters on the block's dialog window. Elemental blocks in Extend include Generator, Queue, Activity, Resource Pool, and Exit. The active entities, called items in Extend, are created at Generator blocks and move from block to block by way of item connectors. Separate value connectors allow the attachment of a calculation to a block parameter or the retrieval of statistical information for reporting purposes. Input parameters can be changed interactively during a model run and can come from external sources. Outputs are displayed dynamically and in graphical and tabular format. The Industry and Suite products also provide an embedded database for centralized information management.

Extend provides iconic process-flow animation of the block diagram. For scaled 2-D animation, Proof Animation [Henriksen, 2002] from Wolverine Software is included in the Suite product. Collections of blocks representing a submodel, such as a subassembly line or functional process, can be grouped into a hierarchical block on the model worksheet; hierarchical blocks can also be stored in a library for reuse. Parameters from the submodel can be grouped and displayed at the level of the hierarchical block for access to model I/O. Extend supports the Microsoft component object model (COM/ActiveX), open database connectivity (ODBC), and Internet data exchange. Activity-based costing, statistical analysis of output data with confidence intervals, and the Evolutionary Optimizer are included.

For creating new blocks, Extend comes with a compiled C-like programming environment. The message-based language includes simulation-specific functions and supports custom interface development. Extend has an open architecture; in most cases, the source code for blocks is available for custom development. The architecture also supports linking to and using code and routines written in external languages.

4.7.4 Flexsim

Flexsim simulation software is developed and owned by Flexsim Software Products, Inc. of Orem, Utah (Nordgren, 2003). Flexsim is a discrete-event, object-oriented simulator developed in C++, using Open GL technology. Animation can be shown in tree view, 2-D, 3-D, and virtual reality. All views can be shown concurrently during the model development of run phase. It integrates Microsoft's Visual C++ IDE and compiler within a graphical 3-D click-and-drag simulation environment.

Flexsim software is used to build models that behave like the actual physical or conceptual systems they represent. A simulation model of any flow system or process can be created in Flexsim by using drag-and-drop model-building objects.

Flexsim is used to improve production efficiencies and reduce operating costs through simulation, experimentation, and optimization of dynamic flow systems. Engineers and managers use Flexsim to evaluate plant capacity, balance packaging and manufacturing lines, manage bottlenecks, solve work-in-process problems, justify capital expenditures, plan equipment maintenance schedules, establish proper inventory levels, improve order-picking systems, and optimize production rates. Flexsim allows end users to introduce and simulate new conditions for the model and to analyze their effects and results in order to find ways to improve the system being studied. By using Flexsim, efficiencies—increased throughput and decreased costs—can be identified, tested, and proven prior to implementing them in the actual system. The results of each simulation can be analyzed graphically through 3-D animation and through statistical reports and graphs, which are all also useful in communicating a model's purpose and results to both technical and nontechnical audiences.

4.7.5 Micro Saint

Micro Saint is offered by Micro Analysis and Design, Inc. [Bloechle and Schunk, 2003]. Micro Saint is a general-purpose, discrete-event, network simulation-software package for building models that simulate real-life processes. With Micro Saint models, users can gain useful information about processes that might be too expensive or time-consuming to test in the real world.

Micro Saint does not use the terminology or graphic representations of a specific industry. A Micro Saint model can be built for any process that can be represented by a flowchart diagram. The terms that are used are defined by the user. In addition, the icons and background for the ActionView animation and the flowcharting symbols are customizable. Micro Saint provides two views of the simulation model. The network diagram view shows the process flowchart in action, and ActionView provides a realistic 2-D picture of the process.

Micro Saint supports the development of models of various complexity to match the user's needs. Simple, functional models can be built by drawing a network diagram and filling in the task-timing information. More complex models can also be built that include dynamically changing variables, probabilistic and tactical branching logic, sorted queues, conditional task execution, animation, optimization, and extensive data collection.

A separate module (called COM Services) is available that enables Micro Saint to exchange data with other software applications and makes it easy to customize the model. In addition, OptQuest optimization is included with Micro Saint and is designed to automatically search for and find optimal or near-optimal solutions to the model.

4.7.6 ProModel

ProModel is offered by PROMODEL Corporation [Harrell, 2003]. It is a simulation and animation tool designed to model manufacturing systems. The company also offers MedModel for healthcare systems and ServiceModel for service systems. ProModel offers 2-D animation with an optional 3-D like perspective view. ProModel's animation is generated automatically as the model is developed.

ProModel has manufacturing-oriented modeling elements and rule-based decision logic. Some systems can be modeled by selecting from ProModel's set of highly parameterized modeling elements. In addition, its simulation programming language provides for modeling special situations not covered by the built-in choices.

The modeling elements in ProModel are parts (entities), locations, resources, path networks, routing and processing logic, and arrivals. Parts arrive and follow the routing and processing logic from location to location. Resources are used to represent people, tools, or vehicles that transport parts between locations, perform an operation on a part at a location, or perform maintenance on a location or other resource that is down. Resources may travel on path networks with given speeds, accelerations, and pickup and setdown travel times. The routing and processing element allows user-defined procedural logic in ProModel's simulation-programming language.

ProModel includes logic for automatically generating cost data associated with a process. Costs can be added for location usage, resources, and entities.

ProModel comes complete with an output viewer, allowing for straightforward data presentation and useful graphics and charts, such as state diagrams.

ProModel's runtime interface allows a user to define multiple scenarios for experimentation. SimRunner (discussed in Section 4.8.2) adds the capability to perform an optimization. It is based on an evolutionary-strategy algorithm, a variant of the genetic algorithm approach. The OptQuest Optimizer (OptQuest for ProModel) is available as an add-on product.

4.7.7 QUEST

QUEST® is offered by Delmia Corp. QUEST (Queuing Event Simulation Tool) is a manufacturing-oriented simulation package. QUEST combines an object-based, true 3-D simulation environment with a graphical user interface and material-flow modules for modeling labor, conveyors, automated guided vehicles, kinematic devices, cranes, fluids, power and free conveyors, and automated storage and retrieval systems. QUEST models incorporate 2-D and 3-D CAD geometry to create a virtual factory environment.

Delmia also offers a number of workcell simulators, including IGRIP® for robotic simulation and programming and ERGO™ for ergonomic analyses. Robots and human-based workcells that are simulated in IGRIP and ERGO can be imported into QUEST models both visually and numerically.

Delmia provides even further integration with QUEST and other manufacturing technologies through PROCESS ENGINEER™, Delmia's process-planning environment. The Manufacturing Hub infrastructure behind this software consists of an object-oriented database for storing Product, Process, and Resource objects that are configuration-managed and effectivity-controlled. A QUEST model is automatically created from the information stored in the database, and the resulting model can be linked to the database for automatic update purposes. QUEST can be used to introduce and update resource-specific information and model output results into the Manufacturing Hub for use in other products.

A QUEST model consists of elements from a number of element classes. Built-in element classes include AGVs and transporters, subresources, buffers, conveyors, power and free systems, labor, machines, parts, container parts, and processes. Each element has associated geometric data and parameters that define its behavior. Parts may have a route and control rules to govern part flow. Commonly needed behavior logic is selected from comprehensive menus, many parameter-driven.

For unique problems, Delmia's QUEST Simulation Control Language (SCL) can be used. This structured simulation-programming language provides distributed processing with access to all system variables. SCL allows expert users to define custom behaviors and to gain control over the simulation.

Delmia QUEST's open architecture allows the advanced user to perform batch simulation runs to automatically collect and tabulate data by using the Batch Control Language (BCL). Replications and parameter optimization are controlled with batch command files or by the OptQuest optimization software, as described in Section 4.8.2.

Output is available both numerically (with the statistical reporting mechanisms) and visually (with a resulting virtual factory-like animation). Statistical output results are available internally through the graphical user interface or externally through HTML and can be customized by using XML or QUEST's own BCL. Digital movies can be created from the animation, or a read-only encrypted version of the model can be authored for viewing and experimentation in QUEST Express™, a "lite" version of QUEST.

4.7.8 SIMUL8

SIMUL8 is provided by SIMUL8 Corporation and was first introduced in 1995. In SIMUL8, models are created by drawing the flow of work with the computer mouse, using a series of icons and arrows to represent the resources and queues in the system. Default values are provided for all properties of the icons, so that the animation can be viewed very early in the modeling process. Drilling down in property boxes opens up progressively more detailed properties. The main focus of SIMUL8 is service industries where people are processing transactions.

Like some other packages, SIMUL8 has the concepts of "Templates" and "Components." Templates, or prebuilt simulations, focus on particular recurring decision types that can be quickly parameterized to fit a specific company issue. Components are user-defined icons that can be reused and shared across a company's simulations. This reduces the time to build simulations, standardizes how some situation are handled across a corporation, and often removes much of the data-collection phase of a simulation study.

SIMUL8 Corporation's approach to business is different from most of the other packages here in that they claim to be aiming to spread simulation very widely across businesses, rather than concentrate it in the hands of dedicated and highly trained simulation professionals. This means they have very different pricing and support policies, but it also means the software has to contain features that watch how the product is being used and provide assistance if some potentially invalid analysis is conducted.

SIMUL8 saves its simulation model and data in XML format so that it will be easy to transfer it to and from other applications. It provides some nonsimulation features that make it possible for the model-builder to create custom user interfaces in spreadsheet, dialog, or wizard form. SIMUL8 has a VBA interface and supports ActiveX/COM so that external applications can build and control SIMUL8 simulations.

The product is available in two levels, Standard and Professional. The two levels provide the same simulation features, but Professional adds 3-D, "Virtual Reality" views of the simulation, and database links to corporate databases and has certain features that are likely to be useful only to full-time simulation modelers. SIMUL8 Professional comes with a license to distribute simulations with a free SIMUL8 Viewer.

4.7.9 WITNESS

WITNESS is offered by the Lanner Group and has separate versions for manufacturing and service industries. It contains many elements for discrete-part manufacturing and also contains elements for continuous processing, such as the flow of fluids through processors, tanks, and pipes.

WITNESS models are based on template elements. These may be customized and combined into module elements and templates for reuse. The standard machine elements can be single, batch, production, assembly, multistation, or multicycle. Other discrete modeling elements include multiple types of conveyor, tracks, vehicles, labor, and carriers. The behavior of each element is described on a tabbed detail form in the WITNESS user interface.

The models are displayed in a 2-D layout animation with multiple windows and display layers; there are optional process-flow displays and element-routing overlays. Models can be changed at any point in a model run and saved at any run point for future reload.

Optional WITNESS modules include WITNESS VR, an integrated virtual reality 3-D view of the working model, where there is full mouse control of the camera flight and position. Options exist, too, for post-processed VR with multiscreen projection and various headset technologies. Other WITNESS modules include links to CAD systems, a model documentor, and the WITNESS Optimizer outlined in the section below.

WITNESS has object-model and ActiveX control for simulation embedding and includes direct data links to Microsoft Excel, MINITAB, and any OLEDB database source. XML data format saves offer additional linkage functionality.

4.8 EXPERIMENTATION AND STATISTICAL-ANALYSIS TOOLS

4.8.1 Common Features

Virtually all simulation packages offer various degrees of support for statistical analysis of simulation outputs. In recent years, many packages have added optimization as one of the analysis tools. To support analysis, most packages provide scenario definition, run-management capabilities, and data export to spreadsheets and other external applications.

Optimization is used to find a "near-optimal" solution. The user must define an objective or fitness function, usually a cost or cost-like function that incorporates the trade-off between additional throughput and additional resources. Until recently, the methods available for optimizing a system had difficulty coping with the random and nonlinear nature of most simulation outputs. Advances in the field of metaheuristics have offered new approaches to simulation optimization, ones based on artificial intelligence, neural networks, genetic algorithms, evolutionary strategies, tabu search, and scatter search.

4.8.2 Products

This section briefly discusses Arena's Output and Process Analyzer, AutoStat for AutoMod, OptQuest (which is used in a number of simulation products) and SimRunner for ProModel.

Arena's Output and Process Analyzer

Arena comes with the Output Analyzer and Process Analyzer. In addition, Arena uses OptQuest for optimization.

The Output Analyzer provides confidence intervals, comparison of multiple systems, and warm-up determination to reduce initial condition biases. It creates various plots, charts, and histograms, smoothes responses, and does correlation analysis. To compute accurate confidence intervals, it does internal batching (both within and across replications, with no user intervention) and data truncation to provide stationary, independent, and normally distributed data sets.

The Process Analyzer adds sophisticated scenario-management capabilities to Arena for comprehensive design of experiments. It allows a user to define scenarios, make the desired runs, and analyze the results. It allows an arbitrary number of controls and responses. Responses can be added after runs have been completed. It will rank scenarios by any response and provide summaries and statistical measures of the responses. A user can view 2-D and 3-D charts of response values across either replications or scenarios.

AutoStat

AutoStat is the run manager and statistical-analysis product in the AutoMod product family [Rohrer, 2003]. AutoStat provides a number of analyses, including warm-up determination for steady-state analysis, absolute and comparison confidence intervals, design of experiments, sensitivity analysis, and optimization via an evolutionary strategy. The evolutionary-strategies algorithm used by AutoStat is well suited to finding a near-optimal solution without getting trapped at a local optimum.

With AutoStat, an end user can define any number of scenarios by defining factors and their range of values. Factors include single parameters, such as resource capacity or vehicle speed; single cells in a data file; and complete data files. By allowing a data file to be a factor, a user can experiment with, for example, alternate production schedules, customer orders for different days, different labor schedules, or any other numerical inputs typically specified in a data file. Any standard or custom output can be designated as a response. For each defined response, AutoStat computes descriptive statistics (average, standard deviation, minimum, and maximum) and confidence intervals. New responses can be defined after runs are made, because AutoStat archives and compresses the standard and custom outputs from all runs. Various charts and plots are available to provide graphical comparisons.

AutoStat supports correlated sampling (see Chapter 12) using common random numbers. This sampling technique minimizes variation between paired samples, giving a better indication of the true effects of model changes.

AutoStat is capable of distributing simulation runs across a local area network and pulling back all results to the user's machine. Support for multiple machines and CPU's gives users the ability to make many more runs of the simulation than would otherwise be possible, by using idle machines during off hours. This is especially useful in multifactor analysis and optimization, both of which could require large numbers of runs. AutoStat also has a diagnostics capability that automatically detects "unusual" runs, where the definition of "unusual" is user-definable.

AutoStat also works with two other products from AutoSimulations: the AutoMod Simulator, a spreadsheet-based job-shop simulator; and AutoSched AP, a rule-based simulation package for finite-capacity scheduling in the semiconductor industry.

OptQuest

OptQuest® was developed by Dr. Fred Glover of the University of Colorado, cofounder of OptTek Systems, Inc. [April *et al.*, 2003].

OptQuest is based on a combination of methods: scatter search, tabu search, linear/integer programming, and neural networks. Scatter search is a population-based approach where existing solutions are combined to create new solutions. Tabu search is then superimposed to prohibit the search from reinvestigating previous solutions, and neural networks screen out solutions likely to be poor. The combination of methods allows the search process to escape local optimality in the quest for the best solution.

Some of the differences between OptTek's methods and other methods include

- the ability to avoid being trapped in locally optimal solutions to problems that contain nonlinearities (which commonly are present in real-world problems);
- the ability to handle nonlinear and discontinuous relationships that are not specifiable by the kinds of equations and formulas that are used in standard mathematical programming formulations;
- the ability to solve problems that involve uncertainties, such as those arising from uncertain supplies, demands, prices, costs, flow rates, and queuing rates.

SimRunner

SimRunner was developed by PROMODEL Corporation out of the simulation-optimization research of Royce Bowden, Mississippi State University [Harrell *et al.*, 2003]. It is available for ProModel, MedModel, and ServiceModel.

SimRunner uses genetic algorithms and evolution strategies, which are variants of evolutionary algorithms. Evolutionary algorithms are population-based direct-search techniques. A user first specifies input factors (integer or real-valued decision variables) composed of ProModel macros and then specifies an objective function composed of simulation-output responses. SimRunner manipulates the input factors within boundaries specified by the user seeking to minimize, to maximize, or to achieve a user-specified target value for the objective function. The optimization-output report includes a confidence interval on the mean value of the objective function for each solution evaluated over the course of the optimization and displays 3-D plots of the simulation's output-response surface for the solutions evaluated. In addition to the multivariable optimization module, SimRunner has a utility for helping users estimate the end of the warm-up phase (initialization bias) of a steady-state simulation and the number of replications needed to obtain an estimate of the objective function's mean value to within a specified percentage error and confidence level.

REFERENCES

APRIL, J., F. GLOVER, J. P. KELLY, AND M. LAGUNA [2003], "Practical Introduction to Simulation Optimization," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 71–78.

BANKS, J., J. S. CARSON, AND J. N. SY [1995], *Getting Started with GPSS/H*, 2d ed., Wolverine Software Corporation, Annandale, VA.

BANKS, J. [1996], "Interpreting Software Checklists," *OR/MS Today*, June.

BAPAT, V. AND D. STURROCK [2003], "The Arena Product Family: Enterprise Modeling Solutions," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 210– 217.

BLOECHLE, W. K., AND D. SCHUNK [2003], "Micro Saint Sharp Simulation Software," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 182–187.

COWIE, J. [1999], "Scalable Simulation Framework API Reference Manual," `www.ssfnet.org/SSFdocs/ssfapiManual.pdf`.

CRAIN, R. C., AND J. O. HENRIKSEN [1999], "Simulation Using GPSS/H," *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, G. W. Evans, eds., Phoenix, AZ, Dec. 5–8, pp. 182–187.

HARRELL, C. R., B. K. GHOSH, AND R. BOWDEN [2003], *Simulation Using ProModel*, 2d ed., New York: McGraw-Hill.

HARRELL, C. R., AND R. N. PRICE [2003], "Simulation Modeling Using ProModel," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 175–181.

HENRIKSEN, J. O. [1999], "General-Purpose Concurrent and Post-Processed Animation with Proof," *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, G. W. Evans, eds., Phoenix, AZ, Dec. 5–8, pp. 176–181.

KRAHL, D. [2003], "Extend: An Interactive Simulation Environment," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 188–196.

MEHTA, A., AND I. RAWLS [1999], "Business Solutions Using Witness," *Proceedings of the 1999 Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, G. W. Evans, eds., Phoenix, AZ, Dec. 5–8, pp. 230–233.

NANCE, R. E. [1995], "Simulation Programming Languages: An Abridged History," Proceedings of the 1995 Winter Simulation Conference, X. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, eds., Arlington, VA, Dec. 13–16, pp. 1307–1313.

NORDGREN, W. B. [2003], "Flexsim Simulation Environment," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 197–200.

PRITSKER, A. A. B., AND C. D. PEGDEN [1979], *Introduction to Simulation and SLAM*, John Wiley, New York.

ROHRER, M. W. [2003], "Maximizing Simulation ROI with AutoMod," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., New Orleans, LA, Dec. 7–10, pp. 201–209.

SWAIN, J. J. [2003], "Simulation Reloaded: Sixth Biennial Survey of Discrete-Event Software Tools," *OR/MS Today*, August, Vol. 30, No. 4, pp. 46–57.

TOCHER, D. D., AND D. G. OWEN [1960], "The Automatic Programming of Simulations," *Proceedings of the Second International Conference on Operational Research*, J. Banbury and J. Maitland, eds., pp. 50–68.

WILSON, J. R., *et al.* [1992], "The Winter Simulation Conference: Perspectives of the Founding Fathers," *Proceedings of the 1992 Winter Simulation Conference*, J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson, eds., Arlington, VA, Dec. 13–16, pp.37–62.

EXERCISES

For the exercises below, reader should code the model in a general-purpose language (such as C, C++, or Java), a special-purpose simulation language (such as GPSS/H), or any desired simulation package.

Most problems contain activities that are uniformly distributed over an interval $[a,b]$. Assume that all values between a and b are possible; that is, the activity time is a *continuous* random variable.

The uniform distribution is denoted by $U(a, b)$, where a and b are the endpoints of the interval, or by $m \pm h$, where m is the mean and h is the "spread" of the distribution. These four parameters are related by the equations

$$m = \frac{a+b}{2} \qquad h = \frac{b-a}{2}$$
$$a = m-h \qquad b = m+h$$

Some of the uniform-random-variate generators available require specification of a and b; others require m and h.

Some problems have activities that are assumed to be normally distributed, as denoted by $N(\mu, \sigma^2)$, where μ is the mean and σ^2 the variance. (Since activity times are nonnegative, the normal distribution is appropriate only if $\mu \geq k\sigma$, where k is at least 4 and preferably 5 or larger. If a negative value is generated, it

is discarded.) Other problems use the exponential distribution with some rate λ or mean $1/\lambda$. Chapter 5 reviews these distributions; Chapter 8 covers the generation of random variates having these distributions. All of the languages have a facility to easily generate samples from these distributions. For C, C++, or Java simulations, the student may use the functions given in Section 4.4 for generating samples from the normal and exponential distributions.

1. Make the necessary modifications to the Java model of the checkout counter (Example 4.2) so that the simulation will run for exactly 60 hours.
2. In addition to the changes in Exercise 1, assume that an arriving customer does not join the queue if three or more customers are waiting for service. Make necessary changes to the Java code and run the model.
3. Implement the changes in Exercises 1 and 2 in any of the simulation packages.
4. Ambulances are dispatched at a rate of one every 15 ± 10 minutes in a large metropolitan area. Fifteen percent of the calls are false alarms, which require 12 ± 2 minutes to complete. All other calls can be one of two kinds. The first kind are classified as serious. They constitute 15% of the non-false alarm calls and take 25 ± 5 minutes to complete. The remaining calls take 20 ± 10 minutes to complete. Assume that there are a very large number of available ambulances, and that any number can be on call at any time. Simulate the system until 500 calls are completed.
5. In Exercise 4, estimate the number of ambulances required to provide 100% service.
6. **(a)** In Exercise 4, suppose that there is only one ambulance available. Any calls that arrive while the ambulance is out must wait. Can one ambulance handle the work load?
   (b) Simulate with x ambulances, where $x = 1,2,3,$ or 4, and compare the alternatives on the basis of length of time a call must wait, percentage of calls that must wait, and percentage of time the ambulance is out on call.
7. Passengers arrive at the security screening area at Chattahoochee Airport according to a time given by N(20, 3) seconds. At the first point, the boarding pass and ID are checked by one of two people in a time that is distributed N(12, 1) seconds. (Passengers always pick the shortest line when there is an option.) The next step is the X-ray area which takes a time that is N(15, 2) seconds; there are two lanes open at all times. Some 15% of the people have to be rechecked for a time that N(100, 10) seconds. The number of recheckers needed is to be determined. Simulate this system for eight hours with one and two recheckers.
8. A superhighway connects one large metropolitan area to another. A vehicle leaves the first city every 20 ± 15 seconds. Twenty percent of the vehicles have 1 passenger, 30% of the vehicles have 2 passengers, 10% have 3 passengers, and 10% have 4 passengers. The remaining 30% of the vehicles are buses, which carry 40 people. It takes 60 ± 10 minutes for a vehicle to travel between the two metropolitan areas. How long does it take for 5000 people to arrive in the second city?
9. A restaurant has two sections, that is, meals section and tiffin section. Customers arrive at the restaurant at the rate of one every 60 ± 30 seconds. Of the arriving customers, 50% take only tiffin and 50% take only meals. Immaterial of the type of the customer, it takes 75 ± 40 seconds to provide service. Assuming that there are sufficient number of servers available, determine the time taken to serve 100 customers.
10. Re-do Exercise 9, assuming that of the arriving customers, 50% take only tiffin, 30% take only meals, and the remaining 20% take a combination of meals and tiffin.
11. For Exercise 10, what is the maximum number of servers needed during the course of simulation? Reduce the number of servers one by one and determine the total time to complete 100 services.

12. Customers arrive at an Internet center at the rate of one every 15 ± 5 minutes. 80% of the customers check simply their email inbox, while the remaining 20% download and upload files. An email customer spends 5 ± 2 minutes in the center and the download customer spends 15 ± 5 minutes. Simulate the service completion of 500 customers. Of these 500 customers, determine the number of email and download customers and compare with the input percentage.

13. An airport has two concourses. Concourse 1 passengers arrive at a rate of one every 15 ± 2 seconds. Concourse 2 passengers arrive at a rate of one every 10 ± 5 seconds. It takes 30 ± 5 seconds to walk down concourse 1 and 35 ± 10 seconds to walk down concourse 2. Both concourses empty into the main lobby, adjacent to the baggage claim. It takes 10 ± 3 seconds to reach the baggage claim area from the main lobby. Only 60% of the passengers go to the baggage claim area. Simulate the passage of 500 passengers through the airport system. How many of these passengers went through the baggage claim area? In this problem, the expected number through the baggage claim area can be computed by 0.60(500)=300. How close is the simulation estimate to the expected number? Why the difference?

14. In a multiphasic screening clinic, patients arrive at a rate of one every 5 ± 2 minutes to enter the audiology section. The examination takes 3 ± 1 minutes. Eighty percent of the patients were passed on to the next test with no problems. Of the remaining 20%, one-half require simple procedures that take 2 ± 1 minutes and are then sent for reexamination with the same probability of failure. The other half are sent home with medication. Simulate the system to estimate how long it takes to screen and pass 200 patients. (*Note:* Persons sent home with medication are not considered "passed.")

15. Consider a bank with four tellers. Tellers 3 and 4 deal only with business accounts; Tellers 1 and 2 deal only with general accounts. Clients arrive at the bank at a rate of one every 3 ± 1 minutes. Of the clients, 33% are business accounts. Clients randomly choose between the two tellers available for each type of account. (Assume that a customer chooses a line without regard to its length and does not change lines.) Business accounts take 15 ± 10 minutes to complete, and general accounts take 6 ± 5 minutes to complete. Simulate the system for 500 transactions to be completed. What percentage of time is each type of teller busy? What is the average time that each type of customer spends in the bank?

16. Repeat Exercise 15, but assuming that customers join the shortest line for the teller handling their type of account.

17. In Exercises 15 and 16, estimate the mean delay of business customers and of general customers. (Delay is time spent in the waiting line, and is exclusive of service time.) Also estimate the mean length of the waiting line, and the mean proportion of customers who are delayed longer than 1 minute.

18. Three different machines are available for machining a special type of part for 1 hour of each day. The processing-time data is as follows:

Machine	*Time to Machine One Part*
1	20 ± 4 seconds
2	10 ± 3 seconds
3	15 ± 5 seconds

Assume that parts arrive by conveyor at a rate of one every 15 ± 5 seconds for the first 3 hours of the day. Machine 1 is available for the first hour, machine 2 for the second hour, and machine 3 for the third hour of each day. How many parts are produced in a day? How large a storage area is needed for parts waiting for a machine? Do parts "pile up" at any particular time? Why?

19. People arrive at a self-service cafeteria at the rate of one every 30 ± 20 seconds. Forty percent go to the sandwich counter, where one worker makes a sandwich in 60 ± 30 seconds. The rest go to the main counter, where one server spoons the prepared meal onto a plate in 45 ± 30 seconds. All customers must pay a single cashier, which takes 25 ± 10 seconds. For all customers, eating takes 20 ± 10 minutes. After eating, 10% of the people go back for dessert, spending an additional 10 ± 2 minutes altogether in the cafeteria. Simulate until 100 people have left the cafeteria. How many people are left in the cafeteria, and what are they doing, at the time the simulation stops?

20. Customers arrive at a nationalized bank at the rate of one every 60 ± 40 seconds. 60% of the customers perform money transactions and the remaining 40% do other things such as getting the draft, updating passbooks, etc., which require 3 ± 1 and 4 ± 1 minutes, respectively. Currently, there are separate counters for both the activities. Customers feel that if single window concept is introduced, average waiting time could be reduced. Justify by simulating 200 arrivals.

21. In Exercise 20, in single window system, if an arriving customer balks if three or more customers are in the queue, determine the number of customers balked in each category.

22. Loana Tool Company rents chain saws. Customers arrive to rent chain saws at the rate of one every 30 ± 30 minutes. Dave and Betty handle these customers. Dave can rent a chain saw in 14 ± 4 minutes. Betty takes 10 ± 5 minutes. Customers returning chain saws arrive at the same rate as those renting chain saws. Dave and Betty spend 2 minutes with a customer to check in the returned chain saw. Service is first-come-first-served. When no customers are present, or Betty alone is busy, Dave gets these returned saws ready for rerenting. For each saw, this maintenance and cleanup takes him 6 ± 4 minutes and 10 ± 6 minutes, respectively. Whenever Dave is idle, he begins the next maintenance or cleanup. Upon finishing a maintenance or cleanup, Dave begins serving customers if one or more is waiting. Betty is always available for serving customers. Simulate the operation of the system starting with an empty shop at 8:00 A.M., closing the doors at 6:00 P.M., and getting chain saws ready for re-renting until 7:00 P.M. From 6:00 until 7:00 P.M., both Dave and Betty do maintenance and cleanup. Estimate the mean delay of customers who are renting chain saws.

23. The Department of Industrial Engineering of a university has one Xerox machine. Users of this machine arrive at the rate of one every 20 ± 2 minutes and use it for 15 ± 10 minutes. If the machine is busy, 90% of the users wait and finish the job, while the 10% of the users come back after 10 minutes. Assume that they do not balk again. Simulate for 500 customers and find out the probability that a balking customer need not wait during the second attempt.

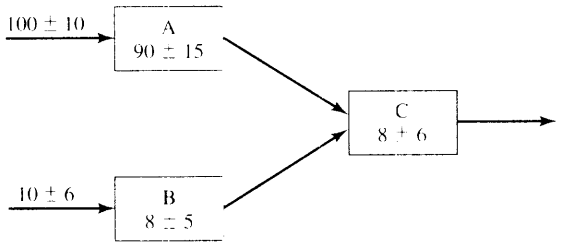
24. Go Ape! buys a Banana II computer to handle all of its web-browsing needs. Web-browsing employees arrive every 10 ± 10 minutes to use the computer. Web-browsing takes 7 ± 7 minutes. The monkeys that run the computer cause a system failure every 60 ± 60 minutes. The failure lasts for 8 ± 4 minutes. When a failure occurs, the web-browsing that was being done resumes processing from where it was left off. Simulate the operation of this system for 24 hours. Estimate the mean system response time. (A system response time is the length of time from arrival until web-browsing is completed.) Also estimate the mean delay for those web-browsing employees that are in service when a computer system failure occurs.

25. Able, Baker, and Charlie are three carhops at the Sonic Drive-In (service at the speed of sound!). Cars arrive every 5 ± 5 minutes. The carhops service customers at the rate of one every 10 ± 6 minutes. However, the customers prefer Able over Baker, and Baker over Charlie. If the carhop of choice is busy, the customers choose the first available carhop. Simulate the system for 1000 service completions. Estimate Able's, Baker's, and Charlie's utilization (percentage of time busy).

26. Jiffy Car Wash is a five-stage operation that takes 2 ± 1 minutes for each stage. There is room for 6 cars to wait to begin the car wash. The car wash facility holds 5 cars, which move through the system in

order, one car not being able to move until the car ahead of it moves. Cars arrive every 2.5 ± 2 minutes for a wash. If the car cannot get into the system, it drives across the street to Speedy Car Wash. Estimate the balking rate per hour. That is, how many cars drive off per hour? Simulate for one 12-hour day.

27. Consider the three machines *A*, *B*, and *C* pictured below. Arrivals of parts and processing times are as indicated (times in minutes).



Machine *A* processes type *I* parts, machine *B* processes type *II* parts, and machine *C* processes both types of parts. All machines are subject to random breakdown: machine *A* every 400 ± 350 minutes with a down time of 15 ± 14 minutes, machine *B* every 200 ± 150 minutes with a downtime of 10 ± 8 minutes, and machine *C* almost never, so its downtime is ignored. Parts from machine *A* are processed at machine *C* as soon as possible, ahead of any type *II* parts from machine *B*. When machine *A* breaks down, any part in it is sent to machine *B* and processed as soon as *B* becomes free, but processing begins over again, taking 100 ± 20 minutes. Again, type *I* parts from machine *A* are processed ahead of any parts waiting at *B*, but after any part currently being processed. When machine *B* breaks down, any part being processed resumes processing as soon as *B* becomes available. All machines handle one part at a time. Make two independent replications of the simulation. Each replication will consist of an 8-hour initialization phase to load the system with parts, followed by a 40-hour steady-state run. (Independent replications means that each run uses a different stream of random numbers.) Management is interested in the long-run throughput [i.e., the number of parts of each type (*I* and *II*) produced per 8-hour day], long-run utilization of each machine, and the existence of bottlenecks (long "lines" of waiting parts, as measured by the queue length at each machine). Report the output data in a table similar to the following:

	Run 1	*Run 2*	*Average of 2 Runs*
Utilization A			
Utilization B			
Etc.			

Include a brief statement summarizing the important results.

28. Students are arriving at the college office at the rate of one every 6 ± 2 minutes to pay the fees. They hand over the forms to one of the two clerks available and it takes 10 ± 2 minutes for the clerk to verify each form. Then the forms are sent to a single cashier who takes 6 ± 1 minute per form. Simulate the system for 100 hours and determine the

(a) utilization of each clerk
(b) utilization of the cashier
(c) average time required to process a form (clerk + cashier)

29. People arrive at a visa office at the rate of one every 15 ±10 minutes. There are three officers (A, B, and C) who scrutinize the applications for a duration of 30 ± 10 minutes. From the past records, it is found

that on an average, 25% of the applications are rejected. Visa applicants form a single line and go to the officer whoever becomes free. If all the three are free, customers always select officer B who is believed to be considerate. Simulate for 500 visa applicants and determine

a) How many of them selected officer B?
b) How many visa applications are rejected?

30. People arrive at a microscope exhibit at a rate of one every 8 ± 2 minutes. Only one person can see the exhibit at a time. It takes 5 ± 2 minutes to see the exhibit. A person can buy a "privilege" ticket for \$1 which gives him or her priority in line over those who are too cheap to spend the buck. Some 50% of the viewers are willing to do this, but they make their decision to do so only if one or more people are in line when they arrive. The exhibit is open continuously from 10:00 A.M. to 4:00 P.M. Simulate the operation of the system for one complete day. How much money is generated from the sale of privilege tickets?

31. Two machines are available for drilling parts (A-type and B-type). A-type parts arrive at a rate of one every 10 ± 3 minutes, B-type parts at a rate of one every 3 ± 2 minutes. For B-type parts, workers choose an idle machine, or if both drills, the Dewey and the Truman, are busy, they choose a machine at random and stay with their choice. A-type parts must be drilled as soon as possible; therefore, if a machine is available, preferably the Dewey, it is used; otherwise the part goes to the head of the line for the Dewey drill. All jobs take 4 ± 3 minutes to complete. Simulate the completion of 100 A-type parts. Estimate the mean number of A-type parts waiting to be drilled.

32. A computer center has two color printers. Students arrive at a rate of one every 8 ± 2 minutes to use the color printer. They can be interrupted by professors, who arrive at a rate of one every 12 ± 2 minutes. There is one systems analyst who can interrupt anyone, but students are interrupted before professors. The systems analyst spends 6 ± 4 minutes on the color printer and then returns in 20 ± 5 minutes. Professors and students spend 4 ± 2 minutes on the color printer. If a person is interrupted, that person joins the head of the queue and resumes service as soon as possible. Simulate for 50 professor-or-analyst jobs. Estimate the interruption rate per hour, and the mean length of the waiting line of students.

33. Parts are machined on a drill press. They arrive at a rate of one every 5 ± 3 minutes, and it takes 3 ± 2 minutes to machine them. Every 60 ± 60 minutes, a rush job arrives, which takes 12 ± 3 minutes to complete. The rush job interrupts any nonrush job. When the regular job returns to the machine, it stays only for its remaining process time. Simulate the machining of 10 rush jobs. Estimate the mean system response time for each type of part. (A response time is the total time that a part spends in the system.)

34. Pull system is used to assemble items in an assembly line. There are two stations. Station I receives items at the rate of one every 12 ± 3 minutes. The operator in station I takes 14 ± 4 minutes, while the station II operator takes 15 ± 2 minutes. The space between the two stations can accommodate only three parts. Hence, if the space is full, the station I operator has to wait till the station II operator removes one part. Simulate the system for 8 hours of operation.

35. For Exercise 34, comment on the output of the model as to whether it will give the true utilization of the station I server.

36. A patient arrives at the Emergency Room at Hello-Hospital about every 40 ± 19 minutes. Each patient will be treated by either Doctor Slipup or Doctor Gutcut. Twenty percent of the patients are classified as NIA (need immediate attention) and the rest as CW (can wait). NIA patients are given the highest priority (3), see a doctor as soon as possible for 40 ± 37 minutes, but then their priority is reduced to 2 and they wait until a doctor is free again, when they receive further treatment for 30 ± 25 minutes and are then discharged. CW patients initially receive the priority 1 and are treated (when their turn comes) for 15 ± 14 minutes; their priority is then increased to 2, they wait again until a doctor is free and receive

10 ± 8 minutes of final treatment, and are then discharged. Simulate for 20 days of continuous operation, 24 hours per day. Precede this by a 2-day initialization period to load the system with patients. Report conditions at times 0 days, 2 days, and 22 days. Does a 2-day initialization appear long enough to load the system to a level reasonably close to steady-state conditions? (a) Measure the average and maximum queue length of NIA patients from arrival to first seeing a doctor. What percent do not have to wait at all? Also tabulate and plot the distribution of this initial waiting time for NIA patients. What percent wait less than 5 minutes before seeing a doctor? (b) Tabulate and plot the distribution of total time in system for all patients. Estimate the 90% quantile—that is, 90% of the patients spend less than x amount of time in the system. Estimate x. (c) Tabulate and plot the distribution of remaining time in system from after the first treatment to discharge, for all patients. Estimate the 90% quantile. (*Note:* Most simulation packages provide the facility to automatically tabulate the distribution of any specified variable.)

37. People arrive at a newspaper stand with an interarrival time that is exponentially distributed with a mean of 0.5 minute. Fifty-five percent of the people buy just the morning paper, 25% buy the morning paper and a *Wall Street Journal*. The remainder buy only the *Wall Street Journal*. One clerk handles the *Wall Street Journal* sales, another clerk morning-paper sales. A person buying both goes to the *Wall Street Journal* clerk. The time it takes to serve a customer is normally distributed with a mean of 40 seconds and a standard deviation of 4 seconds for all transactions. Collect statistics on queues for each type of transaction. Suggest ways for making the system more efficient. Simulate for 4 hours.

38. Bernie remodels houses and makes room additions. The time it takes to finish a job is normally distributed with a mean of 17 elapsed days and a standard deviation of 3 days. Homeowners sign contracts for jobs at exponentially distributed intervals having a mean of 20 days. Bernie has only one crew. Estimate the mean waiting time (from signing the contract until work begins) for those jobs where a wait occurs. Also estimate the percentage of time the crew is idle. Simulate until 100 jobs have been completed.

39. In a certain factory, the tool crib is manned by a single clerk. There are two types of tool request and the time to process a tool request depends on the type of tool request as

Type of Request	*Interarrival Time (Second)*	*Service Time (Second)*
1	Exponential with mean 420	Normal (300,75)
2	Exponential with mean 300	Normal (100,40)

The clerk has been serving the mechanics on FCFS basis. Simulate the system for one day operation (8 hours).

40. In Exercise 39, the management feels that the average number of waiting mechanics can be reduced if Type 2 requests are served ahead of Type 1. Justify.

41. The interarrival time for parts needing processing is given as follows:

Interarrival Time (Seconds)	*Proportion*
10–20	0.20
20–30	0.30
30–40	0.50

There are three types of parts: A, B, and C. The proportion of each part, and the mean and standard deviation of the normally distributed processing times are as follows:

Part Type	*Proportion*	*Mean*	*Standard Deviation*
A	0.5	30 seconds	3 seconds
B	0.3	40 seconds	4 seconds
C	0.2	50 seconds	7 seconds

Each machine processes any type of part, one part at a time. Use simulation to compare one with two with three machines working in parallel. What criteria would be appropriate for such a comparison?

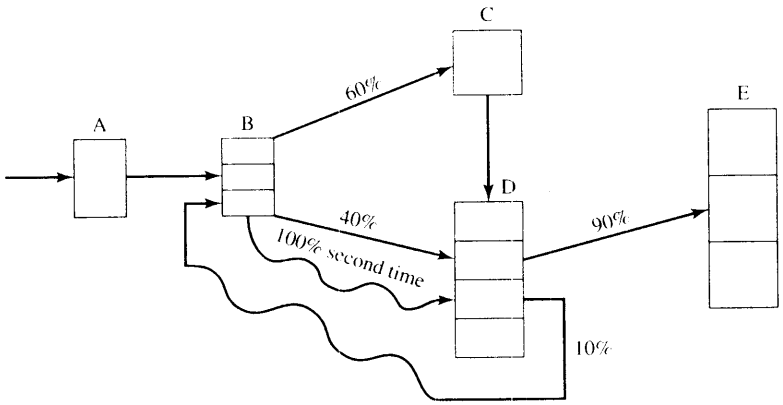
42. Orders are received for one of four types of parts. The interarrival time between orders is exponentially distributed with a mean of 10 minutes. The table that follows shows the proportion of the parts by type and the time to fill each type of order by the single clerk.

Part Type	*Percentage*	*Service Time (Minutes)*
A	40	N(6.1, 1.3)
B	30	N(9.1, 2.9)
C	20	N(11.8, 4.1)
D	10	N(15.1, 4.5)

Orders of types A and B are picked up immediately after they are filled, but orders of types C and D must wait 10 ± 5 minutes to be picked up. Tabulate the distribution of time to complete delivery for all orders combined. What proportion take less than 15 minutes? What proportion take less than 25 minutes? Simulate for an 8-hour initialization period, followed by a 40-hour run. Do not use any data collected in the 8-hour initialization period.

43. Three independent widget-producing machines all require the same type of vital part, which needs frequent maintenance. To increase production it is decided to keep two spare parts on hand (for a total of $2 + 3 = 5$ parts). After 2 hours of use, the part is removed from the machine and taken to a single technician, who can do the required maintenance in 30 ± 20 minutes. After maintenance, the part is placed in the pool of spare parts, to be put into the first machine that requires it. The technician has other duties, namely, repairing other items which have a higher priority and which arrive every 60 ± 20 minutes requiring 15 ± 15 minutes to repair. Also, the technician takes a 15-minute break in each 2-hour time period. That is, the technician works 1 hour 45 minutes, takes off 15 minutes, works 1 hour 45 minutes, takes off 15 minutes, and so on. (a) What are the model's initial conditions—that is, where are the parts at time 0 and what is their condition? Are these conditions typical of "steady state"? (b) Make each replication of this experiment consist of an 8-hour initialization phase followed by a 40-hour data-collection phase. Make four statistically independent replications of the experiment all in one computer run (i.e., make four runs with each using a different set of random numbers). (c) Estimate the mean number of busy machines and the proportion of time the technician is busy. (d) Parts are estimated to cost the company $50 per part per 8-hour day (regardless of how much they are in use). The cost of the technician is $20 per hour. A working machine produces widgets worth $100 for each hour of production. Develop an expression to represent total cost per hour which can be attributed to widget production (i.e., not all of the technician's time is due to widget production). Evaluate this expression, given the results of the simulation.

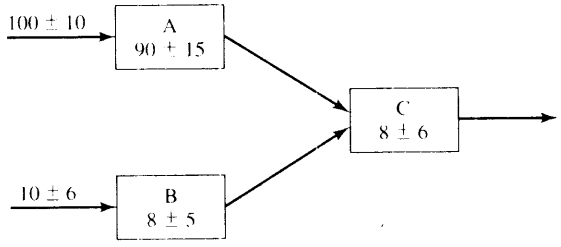
44. The Wee Willy Widget Shop overhauls and repairs all types of widgets. The shop consists of five work stations, and the flow of jobs through the shop is as depicted here:



Regular jobs arrive at station A at the rate of one every 15 ± 13 minutes. Rush jobs arrive every 4 ± 3 hours and are given a higher priority except at station C, where they are put on a conveyor and sent through a cleaning and degreasing operation along with all other jobs. For jobs the first time through a station, processing and repair times are as follows:

Station	*Number Machines or Workers*	*Processing and/or Repair Times (Minutes)*	*Description*
A	1	12 ± 21	Receiving clerk
B	3	40 ± 20	Disassembly and parts replacement
C	1	20	Degreaser
D	4	50 ± 40	Reassembly and adjustments
E	3	40 ± 5	Packing and shipping

The times listed above hold for all jobs that follow one of the two sequences $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ or $A \rightarrow B \rightarrow D \rightarrow E$. However, about 10% of the jobs coming out of station D are sent back to B for further work (which takes 30 ± 10 minutes) and then are sent to D and finally to E. The path of these jobs is as follows:



Every 2 hours, beginning 1 hour after opening, the degreasing station C shuts down for routine maintenance, which takes 10 ± 1 minute. However, this routine maintenance does not begin until the current widget, if any, has completed its processing.

(a) Make three independent replications of the simulation model, where one replication equals an 8-hour simulation run, preceded by a 2-hour initialization run. The three sets of output represent three typical days. The main performance measure of interest is mean response time per job, where a response time is the total time a job spends in the shop. The shop is never empty in the morning, but the model will be empty without the initialization phase. So run the model for a 2-hour initialization period and collect statistics from time 2 hours to time 10 hours. This "warm-up" period will reduce the downward bias in the estimate of mean response time. Note that the 2-hour warm-up is a device to load a simulation model to some more realistic level than empty. From each of the three independent replications, obtain an estimate of mean response time. Also obtain an overall estimate, the sample average of the three estimates

(b) Management is considering putting one additional worker at the busiest station (A, B, D, or E). Would this significantly improve mean response time?

(c) As an alternative to part (b), management is considering replacing machine C with a faster one that processes a widget in only 14 minutes. Would this significantly improve mean response time?

45. A building-materials firm loads trucks with two payloader tractors. The distribution of truck-loading times has been found to be exponential with a mean loading time of 6 minutes. The truck interarrival time is exponentially distributed with an arrival rate of 16 per hour. The waiting time of a truck and driver is estimated to cost \$50 per hour. How much (if any) could the firm save (per 10 hour day) if an overhead hopper system that would fill any truck in a constant time of 2 minutes is installed? (Assume that the present tractors could and would adequately service the conveyors loading the hoppers.)

46. A milling-machine department has 10 machines. The runtime until failure occurs on a machine is exponentially distributed with a mean of 20 hours. Repair times are uniformly distributed between 3 and 7 hours. Select an appropriate run length and appropriate initial conditions.

(a) How many repair persons are needed to ensure that the mean number of machines running is greater than eight?

(b) If there are two repair persons, estimate the number of machines that are either running or being served.

47. Jobs arrive every 300 ± 30 seconds to be processed through a process that consists of four operations: OP10 requires 50 ± 20 seconds, OP20 requires 70 ± 25 seconds, OP30 requires 60 ± 15 seconds, OP40 requires 90 ± 30 seconds. Simulate this process until 250 jobs are completed; then combine the four operations of the job into one with the distribution 240 ± 100 seconds and simulate the process with this distribution. Does the average time in the system change for the two alternatives?

48. Ships arrive at a harbor at the rate of one every 60 ± 30 minutes. There are six berths to accommodate them. They also need the service of a crane for unloading and only one crane is available. After unloading, 10% of the ships stay for refuel before leaving, while the others leave immediately. Ships do not require the use of crane for refueling. It takes 7 ± 3 hours for unloading and 60 ± 20 minutes for refueling. Assume that the crane is subjected to routine maintenance once in every 100 hours, and it takes 5 ± 2 hours to complete the maintenance. The crane's unloading operation is not interrupted for maintenance. The crane is taken for maintenance as early as possible after completing the current unloading activity. Simulate the system for unloading 500 ships that require refueling.

49. Two types of jobs arrive to be processed on the same machine. Type 1 jobs arrive every 80 ± 30 seconds and require 35 ± 20 seconds for processing. Type 2 jobs arrive every 100 ± 40 seconds and require

20 ± 15 seconds for processing. Engineering has judged that there is excess capacity on the machine. For a simulation of 8 hours of operation of the system, find X for Type 3 jobs that arrive every $X \pm 0.4X$ seconds and require a time of 30 seconds on the machine so that the average number of jobs waiting to be processed is two or less.

50. Using spreadsheet software, generate 1000 uniformly distributed random values with mean 10 and spread 2. Plot these values with intervals of width 0.5 between 8 and 12. How close did the simulated set of values come to the expected number in each interval?

51. Using a spreadsheet, generate 1000 exponentially distributed random values with a mean of 10. What is the maximum of the simulated values? What fraction of the generated values is less than the mean of 10? Plot a histogram of the generated values. (Hint: If you cannot find an exponential generator in the spreadsheet you use, use the formula -10*LOG(1−R), where R is a uniformly distributed random number from 0 to 1 and LOG is the natural logarithm. The rationale for this formula is explained in Chapter 8 on random-variate generators.)

Part II

Mathematical and Statistical Models

5

Statistical Models in Simulation

In modeling real-world phenomena, there are few situations where the actions of the entities within the system under study can be predicted completely. The world the model-builder sees is probabilistic rather than deterministic. There are many causes of variation. The time it takes a repairperson to fix a broken machine is a function of the complexity of the breakdown, whether the repairperson brought the proper replacement parts and tools to the site, whether another repairperson asks for assistance during the course of the repair, whether the machine operator receives a lesson in preventive maintenance, and so on. To the model-builder, these variations appear to occur by chance and cannot be predicted. However, some statistical model might well describe the time to make a repair.

An appropriate model can be developed by sampling the phenomenon of interest. Then, through educated guesses (or using software for the purpose), the model-builder would select a known distribution form, make an estimate of the parameter(s) of this distribution, and then test to see how good a fit has been obtained. Through continued efforts in the selection of an appropriate distribution form, a postulated model could be accepted. This multistep process is described in Chapter 9.

Section 5.1 contains a review of probability terminology and concepts. Some typical applications of statistical models, or distribution forms, are given in Section 5.2. Then, a number of selected discrete and continuous distributions are discussed in Sections 5.3 and 5.4. The selected distributions are those that describe a wide variety of probabilistic events and, further, appear in different contexts in other chapters of this text. Additional discussion about the distribution forms appearing in this chapter, and about distribution forms mentioned but not described, is available from a number of sources [Hines and Montgomery, 1990; Ross, 2002; Papoulis, 1990; Devore, 1999; Walpole and Myers, 2002; Law and Kelton, 2000]. Section 5.5 describes the Poisson process and its relationship to the exponential distribution. Section 5.6 discusses empirical distributions.

5.1 REVIEW OF TERMINOLOGY AND CONCEPTS

1. *Discrete random variables.* Let X be a random variable. If the number of possible values of X is finite, or countably infinite, X is called a discrete random variable. The possible values of X may be listed as $x_1, x_2, \ldots$. In the finite case, the list terminates; in the countably infinite case, the list continues indefinitely.

Example 5.1

The number of jobs arriving each week at a job shop is observed. The random variable of interest is X, where

$$X = \text{number of jobs arriving each week}$$

The possible values of X are given by the range space of X, which is denoted by R_X. Here $R_X = \{0, 1, 2, \ldots\}$.

Let X be a discrete random variable. With each possible outcome x_i in R_X, a number $p(x_i) = P(X = x_i)$ gives the probability that the random variable equals the value of x_i. The numbers $p(x_i)$, $i = 1, 2, \ldots$, must satisfy the following two conditions:

1. $p(x_i) \geq 0$, for all i
2. $\sum_i p(x_i) = 1$

The collection of pairs $(x_i, p(x_i))$, $i = 1, 2, \ldots$ is called the probability distribution of X, and $p(x_i)$ is called the probability mass function (pmf) of X.

Example 5.2

Consider the experiment of tossing a single die. Define X as the number of spots on the up face of the die after a toss. Then $R_X = \{1, 2, 3, 4, 5, 6\}$. Assume the die is loaded so that the probability that a given face lands up is proportional to the number of spots showing. The discrete probability distribution for this random experiment is given by

x_i	1	2	3	4	5	6
$p(x_i)$	1/21	2/21	3/21	4/21	5/21	6/21

The conditions stated earlier are satisfied—that is, $p(x_i) \geq 0$ for $i = 1, 2, \ldots, 6$ and $\sum_i p(x_i) = 1/21 + \cdots + 6/21 = 1$. The distribution is shown graphically in Figure 5.1.

2. *Continuous random variables.* If the range space R_X of the random variable X is an interval or a collection of intervals, X is called a continuous random variable. For a continuous random variable X, the probability that X lies in the interval $[a, b]$ is given by

$$P(a \leq X \leq b) = \int_a^b f(x)\,dx \tag{5.1}$$

The function $f(x)$ is called the probability density function (pdf) of the random variable X. The pdf satisfies the following conditions:

a. $f(x) \geq 0$ for all x in R_X

b. $\int_{R_X} f(x)dx = 1$

c. $f(x) = 0$ if x is not in R_X

As a result of Equation (5.1), for any specified value x_0, $P(X = x_0) = 0$, because

$$\int_{x_0}^{x_0} f(x)\,dx = 0$$

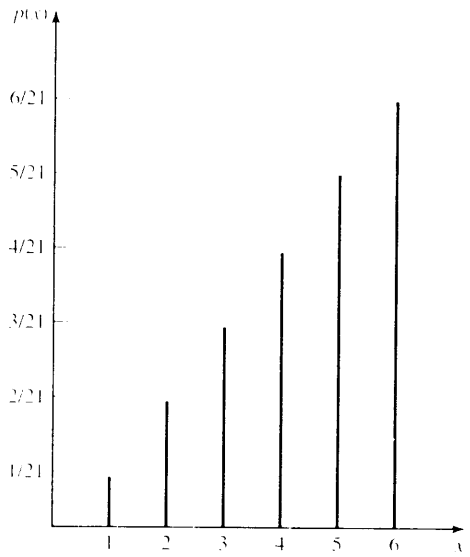


Figure 5.1 Probability mass function for loaded-die example.

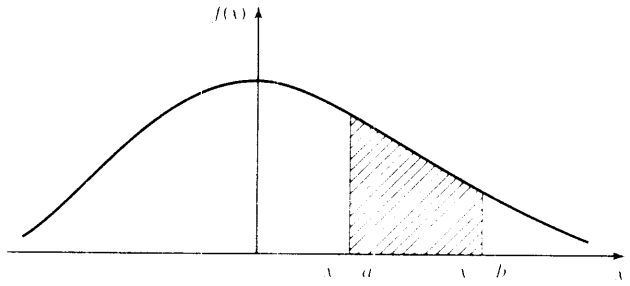


Figure 5.2 Graphical interpretation of $P(a < X < b)$.

$P(X = x_0) = 0$ also means that the following equations hold:

$$P(a \leq X \leq b) = P(a < X \leq b) = P(a \leq X < b) = P(a < X < b) \tag{5.2}$$

The graphical interpretation of Equation (5.1) is shown in Figure 5.2. The shaded area represents the probability that X lies in the interval $[a, b]$.

Example 5.3

The life of a device used to inspect cracks in aircraft wings is given by X, a continuous random variable assuming all values in the range $x \geq 0$. The pdf of the lifetime, in years, is as follows:

$$f(x) = \begin{cases} \frac{1}{2}e^{-\frac{x}{2}}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

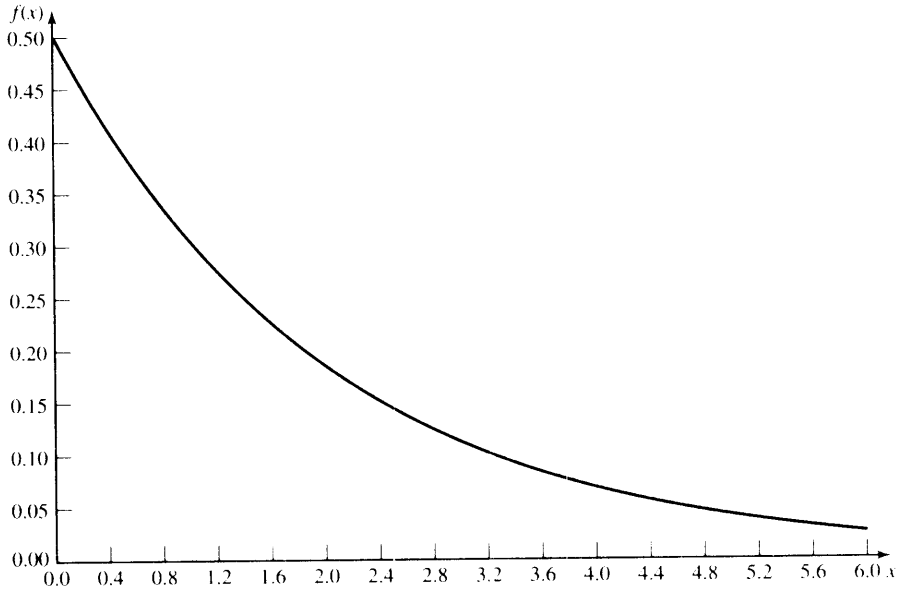


Figure 5.3 pdf for inspection-device life.

This pdf is shown graphically in Figure 5.3. The random variable X is said to have an exponential distribution with mean 2 years.

The probability that the life of the device is between 2 and 3 years is calculated as

$$P(2 \le X \le 3) = \frac{1}{2}\int_2^3 e^{-x/2}dx$$
$$= -e^{-3/2} + e^{-1} = -0.223 + 0.368 = 0.145$$

3. *Cumulative distribution function.* The cumulative distribution function (cdf), denoted by $F(x)$, measures the probability that the random variable X assumes a value less than or equal to x, that is, $F(x) = P(X \le x)$.

If X is discrete, then

$$F(x) = \sum_{\substack{\text{all } x_i \\ x_i \le x}} p(x_i) \tag{5.3}$$

If X is continuous, then

$$F(x) = \int_{-\infty}^{x} f(t)dt \tag{5.4}$$

Some properties of the cdf are listed here:

a. F is a nondecreasing function. If $a < b$, then $F(a) \le F(b)$.
b. $\lim_{x \to \infty} F(x) = 1$
c. $\lim_{x \to -\infty} F(x) = 0$

All probability questions about X can be answered in terms of the cdf. For example,

$$P(a < X \le b) = F(b) - F(a) \quad \text{for all } a < b \tag{5.5}$$

For continuous distributions, not only does Equation (5.5) hold, but also the probabilities in Equation (5.2) are equal to $F(b) - F(a)$.

Example 5.4

The die-tossing experiment described in Example 5.2 has a cdf given as follows:

x	$(-\infty, 1)$	$[1, 2)$	$[2, 3)$	$[3, 4)$	$[4, 5)$	$[5, 6)$	$[6, \infty)$
$F(x)$	0	1/21	3/21	6/21	10/21	15/21	21/21

where $[a, b) = \{a \le x < b\}$. The cdf for this example is shown graphically in Figure 5.4.

If X is a discrete random variable with possible values $x_1, x_2, \ldots$, where $x_1 < x_2 < \ldots$, the cdf is a step function. The value of the cdf is constant in the interval $[x_{i-1}, x_i)$ and then takes a step, or jump, of size $p(x_i)$ at x_i. Thus, in Example 5.4, $p(3) = 3/21$, which is the size of the step when $x = 3$.

Example 5.5

The cdf for the device described in Example 5.3 is given by

$$F(x) = \frac{1}{2}\int_0^x e^{-t/2}dt = 1 - e^{-x/2}$$

The probability that the device will last for less than 2 years is given by

$$P(0 \le X \le 2) = F(2) - F(0) = F(2) = 1 - e^{-1} = 0.632$$

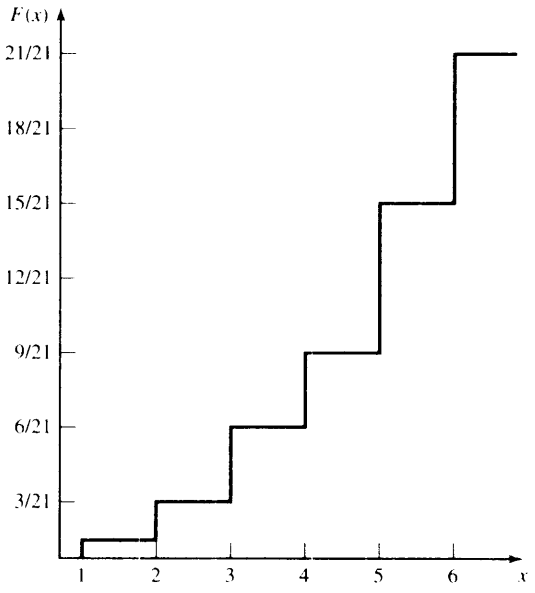


Figure 5.4 cdf for loaded-die example.

The probability that the life of the device is between 2 and 3 years is calculated as

$$P(2 \le X \le 3) = F(3) - F(2) = (1 - e^{-3/2}) - (1 - e^{-1})$$
$$= -e^{-3/2} + e^{-1} = -0.223 + 0.368 = 0.145$$

as found in Example 5.3.

4. *Expectation.* An important concept in probability theory is that of the expectation of a random variable. If X is a random variable, the expected value of X, denoted by $E(X)$, for discrete and continuous variables is defined as follows:

$$E(X) = \sum_{\text{all } x} x_i p(x_i) \qquad \text{if } X \text{ is discrete} \tag{5.6}$$

and

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx \qquad \text{if } X \text{ is continuous} \tag{5.7}$$

The expected value $E(X)$ of a random variable X is also referred to as the mean, μ, or the first moment of X. The quantity $E(X^n)$, $n \ge 1$, is called the nth moment of X, and is computed as follows:

$$E(X^n) = \sum_{\text{all } x} x_i^n p(x_i) \qquad \text{if } X \text{ is discrete} \tag{5.8}$$

and

$$E(X^n) = \int_{-\infty}^{\infty} x^n f(x) dx \qquad \text{if } X \text{ is continuous} \tag{5.9}$$

The variance of a random variable, X, denoted by $V(X)$ or $\text{var}(X)$ or σ^2, is defined by

$$V(X) = E[(X - E[X])^2]$$

A useful identity in computing $V(X)$ is given by

$$V(X) = E(X^2) - [E(X)]^2 \tag{5.10}$$

The mean $E(X)$ is a measure of the central tendency of a random variable. The variance of X measures the expected value of the squared difference between the random variable and its mean. Thus, the variance, $V(X)$, is a measure of the spread or variation of the possible values of X around the mean $E(X)$. The standard deviation, σ, is defined to be the square root of the variance, σ^2. The mean, $E(X)$, and the standard deviation, $\sigma = \sqrt{V(X)}$, are expressed in the same units.

Example 5.6

The mean and variance of the die-tossing experiment described in Example 5.2 are computed as follows:

$$E(X) = 1\left(\frac{1}{21}\right) + 2\left(\frac{2}{21}\right) + \cdots + 6\left(\frac{6}{21}\right) = \frac{91}{21} = 4.33$$

To compute $V(X)$ from Equation (5.10), first compute $E(X^2)$ from Equation (5.8) as follows:

$$E(X^2) = 1^2\left(\frac{1}{21}\right) + 2^2\left(\frac{2}{21}\right) + \cdots + 6^2\left(\frac{6}{21}\right) = 21$$

Thus,

$$V(X) = 21 - \left(\frac{91}{21}\right)^2 = 21 - 18.78 = 2.22$$

and

$$\sigma = \sqrt{V(X)} = 1.49$$

Example 5.7

The mean and variance of the life of the device described in Example 5.3 are computed as follows:

$$E(X) = \frac{1}{2}\int_0^\infty xe^{-x/2}dx = -xe^{-x/2}\Big|_0^\infty + \int_0^\infty e^{-x/2}dx$$

$$= 0 + \frac{1}{1/2}e^{-x/2}\Big|_0^\infty = 2 \text{ years}$$

To compute $V(X)$ from Equation (5.10), first compute $E(X^2)$ from Equation (5.9) as follows:

$$E(X^2) = \frac{1}{2}\int_0^\infty x^2 e^{-x/2}dx$$

Thus,

$$E(X^2) = -x^2 e^{-x/2}\Big|_0^\infty + 2\int_0^\infty xe^{-x/2}dx = 8$$

giving

$$V(X) = 8 - 2^2 = 4 \text{ years}^2$$

and

$$\sigma = \sqrt{V(X)} = 2 \text{ years}$$

With a mean life of 2 years and a standard deviation of 2 years, most analysts would conclude that actual lifetimes, X, have a fairly large variability.

5. *The mode.* The mode is used in describing several statistical models that appear in this chapter. In the discrete case, the mode is the value of the random variable that occurs most frequently. In the continuous case, the mode is the value at which the pdf is maximized. The mode might not be unique; if the modal value occurs at two values of the random variable, the distribution is said to be bimodal.

5.2 USEFUL STATISTICAL MODELS

Numerous situations arise in the conduct of a simulation where an investigator may choose to introduce probabilistic events. In Chapter 2, queueing, inventory, and reliability examples were given. In a queueing system, interarrival and service times are often probabilistic. In an inventory model, the time between demands and the lead times (time between placing and receiving an order) can be probabilistic. In a reliability model, the time to failure could be probabilistic. In each of these instances, the simulation analyst desires to generate random events and to use a known statistical model if the underlying distribution can be found. In the following

paragraphs, statistical models appropriate to these application areas will be discussed. Additionally, statistical models useful in the case of limited data are mentioned.

1. *Queueing systems.* In Chapter 2, examples of waiting-line problems were given. In Chapters 2, 3, and 4, these problems were solved via simulation. In the queueing examples, interarrival- and service-time patterns were given. In these examples, the times between arrivals and the service times were always probabilistic, as is usually the case. However, it is possible to have a constant interarrival time (as in the case of a line moving at a constant speed in the assembly of an automobile), or a constant service time (as in the case of robotized spot welding on the same assembly line). The following example illustrates how probabilistic interarrival times might occur.

Example 5.8

Mechanics arrive at a centralized tool crib as shown in Table 5.1. Attendants check in and check out the requested tools to the mechanics. The collection of data begins at 10:00 A.M. and continues until 20 different interarrival times are recorded. Rather than record the actual time of day, the absolute time from a given origin could have been computed. Thus, the first mechanic could have arrived at time zero, the second mechanic at time 7:13 (7 minutes, 13 seconds), and so on.

Example 5.9

Another way of presenting interarrival data is to find the number of arrivals per time period. Here, such arrivals occur over approximately 1 1/2 hours; it is convenient to look at 10-minute time intervals for the first 20 mechanics. That is, in the first 10-minute time period, one arrival occurred at 10:05::03. In the second time period, two mechanics arrived, and so on. The results are summarized in Table 5.2. This data could then be plotted in a histogram, as shown in Figure 5.5.

Table 5.1 Arrival Data

Arrival Number	*Arrival (Hour:Minutes::Seconds)*	*Interarrival Time (Minutes::Seconds)*
1	10:05::03	—
2	10:12::16	7::13
3	10:15::48	3::32
4	10:24::27	8::39
5	10:32::19	7::52
6	10:35::43	3::24
7	10:39::51	4::08
8	10:40::30	0::39
9	10:41::17	0::47
10	10:44::12	2::55
11	10:45::47	1::35
12	10:50::47	5::00
13	11:00::05	9::18
14	11:04::58	4::53
15	11:06::12	1::14
16	11:11::23	5::11
17	11:16::31	5::08
18	11:17::18	0::47
19	11:21::26	4::08
20	11:24::43	3::17
21	11:31::19	6::36

Table 5.2 Arrivals in Successive Time Periods

Time Period	*Number of Arrivals*	*Time Period*	*Number of Arrivals*
1	1	6	1
2	2	7	3
3	1	8	3
4	3	9	2
5	4	—	—

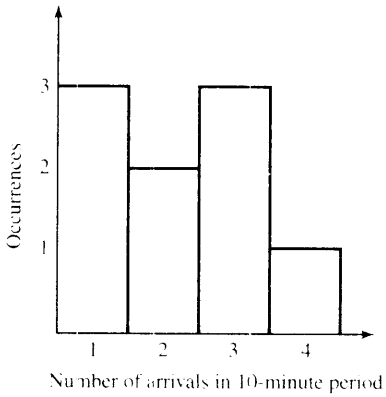


Figure 5.5 Histogram of arrivals per time period.

The distribution of time between arrivals and the distribution of the number of arrivals per time period are important in the simulation of waiting-line systems. "Arrivals" occur in numerous ways; as machine breakdowns, as jobs coming into a jobshop, as units being assembled on a line, as orders to a warehouse, as data packets to a computer system, as calls to a call center, and so on.

Service times could be constant or probabilistic. If service times are completely random, the exponential distribution is often used for simulation purposes; however, there are several other possibilities. It could happen that the service times are constant, but some random variability causes fluctuations in either a positive or a negative way. For example, the time it takes for a lathe to traverse a 10-centimeter shaft should always be the same. However, the material could have slight differences in hardness or the tool might wear; either event could cause different processing times. In these cases, the normal distribution might describe the service time.

A special case occurs when the phenomenon of interest seems to follow the normal probability distribution, but the random variable is restricted to be greater than or less than a certain value. In this case, the truncated normal distribution can be utilized.

The gamma and Weibull distributions are also used to model interarrival and service times. (Actually, the exponential distribution is a special case of both the gamma and the Weibull distributions.) The differences between the exponential, gamma, and Weibull distributions involve the location of the modes of the pdf's and the shapes of their tails for large and small times. The exponential distribution has its mode at the origin, but the gamma and Weibull distributions have their modes at some point (≥ 0) that is a function of the parameter values selected. The tail of the gamma distribution is long, like an exponential distribution; the tail of the Weibull distribution can decline more rapidly or less rapidly than that of an exponential distribution.

In practice, this means that, if there are more large service times than an exponential distribution can account for, a Weibull distribution might provide a better model of these service times.

2. *Inventory and supply-chain systems.* In realistic inventory and supply-chain systems, there are at least three random variables: (1) the number of units demanded per order or per time period, (2) the time between demands, and (3) the lead time. (The lead time is defined as the time between the placing of an order for stocking the inventory system and the receipt of that order.) In very simple mathematical models of inventory systems, demand is a constant over time, and lead time is zero, or a constant. However, in most real-world cases, and, hence, in simulation models, demand occurs randomly in time, and the number of units demanded each time a demand occurs is also random, as illustrated by Figure 5.6.

Distributional assumptions for demand and lead time in inventory theory texts are usually based on mathematical tractability, but those assumptions could be invalid in a realistic context. In practice, the lead-time distribution can often be fitted fairly well by a gamma distribution [Hadley and Whitin, 1963]. Unlike analytic models, simulation models can accommodate whatever assumptions appear most reasonable.

The geometric, Poisson, and negative binomial distributions provide a range of distribution shapes that satisfy a variety of demand patterns. The geometric distribution, which is a special case of the negative binomial, has its mode at unity, given that at least one demand has occurred. If demand data are characterized by a long tail, the negative binomial distribution might be appropriate. The Poisson distribution is often used to model demand because it is simple, it is extensively tabulated, and it is well known. The tail of the Poisson distribution is generally shorter than that of the negative binomial, which means that fewer large demands will occur if a Poisson model is used than if a negative binomial distribution is used (assuming that both models have the same mean demand).

3. *Reliability and maintainability.* Time to failure has been modeled with numerous distributions, including the exponential, gamma, and Weibull. If only random failures occur, the time-to-failure distribution may be modeled as exponential. The gamma distribution arises from modeling standby redundancy, where each component has an exponential time to failure. The Weibull distribution has been extensively used to represent time to failure, and its nature is such that it can be made to approximate many observed phenomena [Hines and Montgomery, 1990]. When there are a number of components in a system and failure is due to

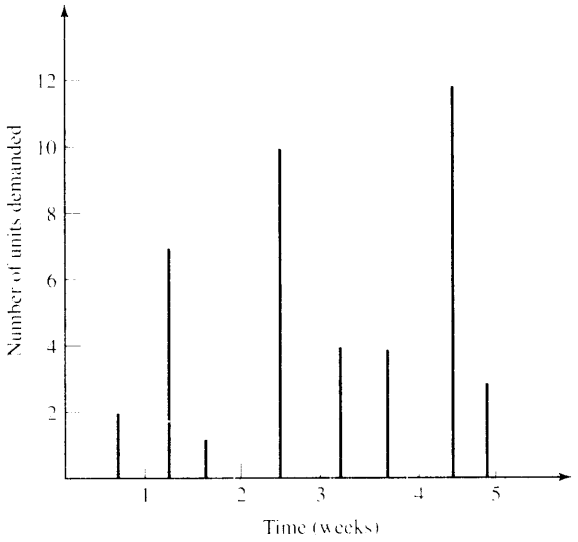


Figure 5.6 Random demands in time.

the most serious of a large number of defects, or possible defects, the Weibull distribution seems to do particularly well as a model. In situations where most failures are due to wear, the normal distribution might very well be appropriate [Hines and Montgomery, 1990]. The lognormal distribution has been found to be applicable in describing time to failure for some types of components.

4. *Limited data.* In many instances, simulations begin before data collection has been completed. There are three distributions that have application to incomplete or limited data. These are the uniform, triangular, and beta distributions. The uniform distribution can be used when an interarrival or service time is known to be random, but no information is immediately available about the distribution [Gordon, 1975]. However, there are those who do not favor using the uniform distribution, calling it the "distribution of maximum ignorance" because it is not necessary to specify more than the continuous interval in which the random variable may occur. The triangular distribution can be used when assumptions are made about the minimum, maximum, and modal values of the random variable. Finally, the beta distribution provides a variety of distributional forms on the unit interval, ones that, with appropriate modification, can be shifted to any desired interval. The uniform distribution is a special case of the beta distribution. Pegden, Shannon, and Sadowski [1995] discuss the subject of limited data in some detail, and we include further discussion in Chapter 9.

5. *Other distributions.* Several other distributions may be useful in discrete-system simulation. The Bernoulli and binomial distributions are two discrete distributions which might describe phenomena of interest. The hyperexponential distribution is similar to the exponential distribution, but its greater variability might make it useful in certain instances.

5.3 DISCRETE DISTRIBUTIONS

Discrete random variables are used to describe random phenomena in which only integer values can occur. Numerous examples were given in Section 5.2—for example, demands for inventory items. Four distributions are described in the following subsections.

1. *Bernoulli trials and the Bernoulli distribution.* Consider an experiment consisting of n trials, each of which can be a success or a failure. Let $X_j = 1$ if the jth experiment resulted in a success, and let $X_j = 0$ if the jth experiment resulted in a failure. The n Bernoulli trials are called a Bernoulli process if the trials are independent, each trial has only two possible outcomes (success or failure), and the probability of a success remains constant from trial to trial. Thus,

$$p(x_1, x_2, \ldots, x_n) = p_1(x_1) \cdot p_2(x_2) \cdots p_n(x_n)$$

and

$$p_j(x_j) = p(x_j) = \begin{cases} p, & x_j = 1,\ j = 1, 2, \ldots, n \\ 1 - p = q, & x_j = 0,\ j = 1, 2, \ldots, n \\ 0, & \text{otherwise} \end{cases} \tag{5.11}$$

For one trial, the distribution given in Equation (5.11) is called the Bernoulli distribution. The mean and variance of X_j are calculated as follows:

$$E(X_j) = 0 \cdot q + 1 \cdot p = p$$

and

$$V(X_j) = [(0^2 \cdot q) + (1^2 \cdot p)] - p^2 = p(1 - p)$$

2. *Binomial distribution.* The random variable X that denotes the number of successes in n Bernoulli trials has a binomial distribution given by $p(x)$, where

$$p(x) = \begin{cases} \binom{n}{x} p^x q^{n-x}, & x = 0, 1, 2, \ldots, n \\ 0, & \text{otherwise} \end{cases} \tag{5.12}$$

Equation (5.12) is motivated by computing the probability of a particular outcome with all the successes, each denoted by S, occurring in the first x trials, followed by the $n - x$ failures, each denoted by an F—that is,

$$P(\overbrace{SSS\ldots\ldots SS}^{x \text{ of these}}\overbrace{FF\ldots\ldots FF}^{n-x \text{ of these}}) = p^x q^{n-x}$$

where $q = 1 - p$. There are

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

outcomes having the required number of S's and F's. Therefore, Equation (5.12) results. An easy approach to calculating the mean and variance of the binomial distribution is to consider X as a sum of n independent Bernoulli random variables, each with mean p and variance $p(1 - p) = pq$. Then,

$$X = X_1 + X_2 + \cdots + X_n$$

and the mean, $E(X)$, is given by

$$E(X) = p + p + \cdots + p = np \tag{5.13}$$

and the variance $V(X)$ is given by

$$V(X) = pq + pq + \cdots + pq = npq \tag{5.14}$$

Example 5.10

A production process manufactures computer chips on the average at 2% nonconforming. Every day, a random sample of size 50 is taken from the process. If the sample contains more than two nonconforming chips, the process will be stopped. Compute the probability that the process is stopped by the sampling scheme.

Consider the sampling process as $n = 50$ Bernoulli trials, each with $p = 0.02$; then the total number of nonconforming chips in the sample, X, would have a binomial distribution given by

$$p(x) = \begin{cases} \binom{50}{x} (0.02)^x (0.98)^{50-x}, & x = 0, 1, 2, \ldots, 50 \\ 0, & \text{otherwise} \end{cases}$$

It is much easier to compute the right-hand side of the following identity to compute the probability that more than two nonconforming chips are found in a sample:

$$P(X > 2) = 1 - P(X \leq 2)$$

The probability $P(X \leq 2)$ is calculated from

$$P(X \le 2) = \sum_{x=0}^{2} \binom{50}{x} (0.02)^x (0.98)^{50-x}$$
$$= (0.98)^{50} + 50(0.02)(0.98)^{49} + 1225(0.02)^2(0.98)^{48}$$
$$= 0.92$$

Thus, the probability that the production process is stopped on any day, based on the sampling process, is approximately 0.08. The mean number of nonconforming chips in a random sample of size 50 is given by

$$E(X) = np = 50(0.02) = 1$$

and the variance is given by

$$V(X) = npq = 50(0.02)(0.98) = 0.98$$

The cdf for the binomial distribution has been tabulated by Banks and Heikes [1984] and others. The tables decrease the effort considerably for computing probabilities such as $P(a < X \le b)$. Under certain conditions on n and p, both the Poisson distribution and the normal distribution may be used to approximate the binomial distribution [Hines and Montgomery, 1990].

3. *Geometric and Negative Binomial distributions.* The geometric distribution is related to a sequence of Bernoulli trials; the random variable of interest, X, is defined to be the number of trials to achieve the first success. The distribution of X is given by

$$p(x) = \begin{cases} q^{x-1}p, & x = 1, 2, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{5.15}$$

The event $\{X = x\}$ occurs when there are $x - 1$ failures followed by a success. Each of the failures has an associated probability of $q = 1 - p$, and each success has probability p. Thus,

$$P(FFF \cdots FS) = q^{x-1}p$$

The mean and variance are given by

$$E(X) = \frac{1}{p} \tag{5.16}$$

and

$$V(X) = \frac{q}{p^2} \tag{5.17}$$

More generally, the negative binomial distribution is the distribution of the number of trials until the kth success, for $k = 1, 2, \ldots$. If Y has a negative binomial distribution with parameters p and k, then the distribution of Y is given by

$$p(y) = \begin{cases} \binom{y-1}{k-1} q^{y-k} p^k, & y = k, k+1, k+2, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{5.18}$$

Because we can think of the negative binomial random variable Y as the sum of k independent geometric random variables, it is easy to see that $E(Y) = k/p$ and $V(X) = kq/p^2$.

Example 5.11

Forty percent of the assembled ink-jet printers are rejected at the inspection station. Find the probability that the first acceptable ink-jet printer is the third one inspected. Considering each inspection as a Bernoulli trial with $q = 0.4$ and $p = 0.6$ yields

$$p(3) = 0.4^2(0.6) = 0.096$$

Thus, in only about 10% of the cases is the first acceptable printer the third one from any arbitrary starting point. To determine the probability that the third printer inspected is the second acceptable printer, we use the negative binomial distribution (5.18).

$$p(3) = \binom{3-1}{2-1} 0.4^{3-2}(0.6)^2 = \binom{2}{1} 0.4(0.6)^2 = 0.288$$

4. *Poisson distribution.* The Poisson distribution describes many random processes quite well and is mathematically quite simple. The Poisson distribution was introduced in 1837 by S. D. Poisson in a book concerning criminal and civil justice matters. (The title of this rather old text is "Recherches sur la probabilite des jugements en matiere criminelle et en matiere civile." Evidently, the rumor handed down through generations of probability theory professors concerning the origin of the Poisson distribution is just not true. Rumor has it that the Poisson distribution was first used to model deaths from the kicks of horses in the Prussian Army.)

The Poisson probability mass function is given by

$$p(x) = \begin{cases} \dfrac{e^{-\alpha}\alpha^x}{x!}, & x = 0, 1, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{5.19}$$

where $\alpha > 0$. One of the important properties of the Poisson distribution is that the mean and variance are both equal to α, that is,

$$E(X) = \alpha = V(X)$$

The cumulative distribution function is given by

$$F(x) = \sum_{i=0}^{x} \frac{e^{-\alpha}\alpha^i}{i!} \tag{5.20}$$

The pmf and cdf for a Poisson distribution with $\alpha = 2$ are shown in Figure 5.7. A tabulation of the cdf is given in Table A.4.

Example 5.12

A computer repair person is "beeped" each time there is a call for service. The number of beeps per hour is known to occur in accordance with a Poisson distribution with a mean of $\alpha = 2$ per hour. The probability of three beeps in the next hour is given by Equation (5.19) with $x = 3$, as follows:

$$p(3) = \frac{e^{-2}2^3}{3!} = \frac{(0.135)(8)}{6} = 0.18$$

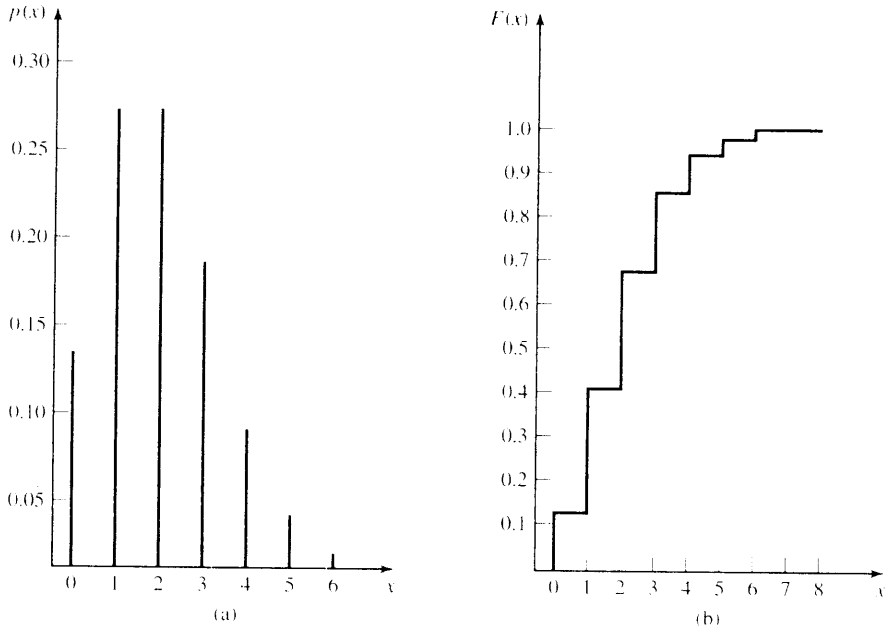


Figure 5.7 Poisson pmf and cdf.

This same result can be read from the left side of Figure 5.7 or from Table A.4 by computing

$$F(3) - F(2) = 0.857 - 0.677 = 0.18$$

Example 5.13

In Example 5.12, find the probability of two or more beeps in a 1-hour period.

$$\begin{aligned} P(2 \text{ or more}) &= 1 - p(0) - p(1) = 1 - F(1) \\ &= 1 - 0.406 = 0.594 \end{aligned}$$

The cumulative probability, $F(1)$, can be read from the right side of Figure 5.7 or from Table A.4.

Example 5.14

The lead-time demand in an inventory system is the accumulation of demand for an item from the point at which an order is placed until the order is received—that is,

$$L = \sum_{i=1}^{T} D_i \tag{5.21}$$

where L is the lead-time demand, D_i is the demand during the ith time period, and T is the number of time periods during the lead time. Both D_i and T may be random variables.

An inventory manager desires that the probability of a stockout not exceed a certain fraction during the lead time. For example, it may be stated that the probability of a shortage during the lead time not exceed 5%.

If the lead-time demand is Poisson distributed, the determination of the reorder point is greatly facilitated. The reorder point is the level of inventory at which a new order is placed.

Assume that the lead-time demand is Poisson distributed with a mean of $\alpha = 10$ units and that 95% protection from a stockout is desired. Thus, it is desired to find the smallest value of x such that the probability that the lead-time demand does not exceed x is greater than or equal to 0.95. Using Equation (5.20) requires finding the smallest x such that

$$F(x) = \sum_{i=0}^{x} \frac{e^{-10}10^i}{i!} \geq 0.95$$

The desired result occurs at $x = 15$, which can be found by using Table A.4 or by computation of $p(0), p(1), \ldots$.

5.4 CONTINUOUS DISTRIBUTIONS

Continuous random variables can be used to describe random phenomena in which the variable of interest can take on any value in some interval—for example, the time to failure or the length of a rod. Eight distributions are described in the following subsections.

1. *Uniform distribution.* A random variable X is uniformly distributed on the interval (a, b) if its pdf is given by

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases} \tag{5.22}$$

The cdf is given by

$$F(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \leq x < b \\ 1, & x \geq b \end{cases} \tag{5.23}$$

Note that

$$P(x_1 < X < x_2) = F(x_2) - F(x_1) = \frac{x_2 - x_1}{b-a}$$

is proportional to the length of the interval, for all x_1 and x_2 satisfying $a \leq x_1 < x_2 \leq b$. The mean and variance of the distribution are given by

$$E(X) = \frac{a+b}{2} \tag{5.24}$$

and

$$V(X) = \frac{(b-a)^2}{12} \tag{5.25}$$

The pdf and cdf when $a = 1$ and $b = 6$ are shown in Figure 5.8.

The uniform distribution plays a vital role in simulation. Random numbers, uniformly distributed between zero and 1, provide the means to generate random events. Numerous methods for generating uniformly distributed random numbers have been devised; some will be discussed in Chapter 7. Uniformly distributed

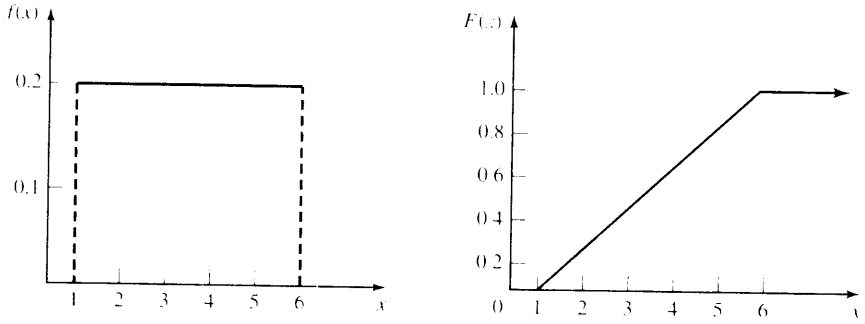


Figure 5.8 pdf and cdf for uniform distribution.

random numbers are then used to generate samples of random variates from all other distributions, as will be discussed in Chapter 8.

Example 5.15

A simulation of a warehouse operation is being developed. About every 3 minutes, a call comes for a fork-lift truck operator to proceed to a certain location. An initial assumption is made that the time between calls (arrivals) is uniformly distributed with a mean of 3 minutes. By Equation (5.25), the uniform distribution with a mean of 3 and the greatest possible variability would have parameter values of $a = 0$ and $b = 6$ minutes. With very limited data (such as a mean of approximately 3 minutes) plus the knowledge that the quantity of interest is variable in a random fashion, the uniform distribution with greatest variance can be assumed, at least until more data are available.

Example 5.16

A bus arrives every 20 minutes at a specified stop beginning at 6:40 A.M. and continuing until 8:40 A.M. A certain passenger does not know the schedule, but arrives randomly (uniformly distributed) between 7:00 A.M. and 7:30 A.M. every morning. What is the probability that the passenger waits more than 5 minutes for a bus?

The passenger has to wait more than 5 minutes only if the arrival time is between 7:00 A.M. and 7:15 A.M. or between 7:20 A.M. and 7:30 A.M. If X is a random variable that denotes the number of minutes past 7:00 A.M. that the passenger arrives, the desired probability is

$$P(0 < X < 15) + P(20 < X < 30)$$

Now, X is a uniform random variable on (0,30). Therefore, the desired probability is given by

$$F(15) + F(30) - F(20) = \frac{15}{30} + 1 - \frac{20}{30} = \frac{5}{6}$$

2. *Exponential distribution.* A random variable X is said to be exponentially distributed with parameter $\lambda > 0$ if its pdf is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & \text{elsewhere} \end{cases} \tag{5.26}$$

The density function is shown in Figures 5.9 and 5.3. Figure 5.9 also shows the cdf.

The exponential distribution has been used to model interarrival times when arrivals are completely random and to model service times that are highly variable. In these instances, λ is a rate: arrivals per hour

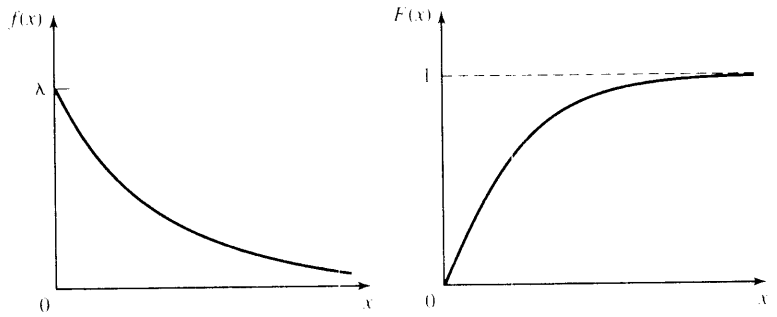


Figure 5.9 Exponential density function and cumulative distribution function.

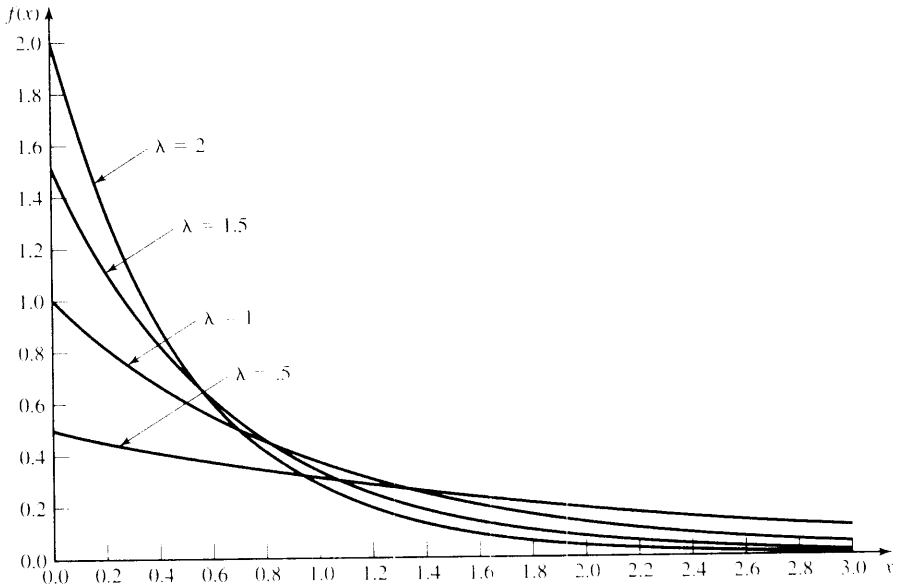


Figure 5.10 pdfs for several exponential distributions.

or services per minute. The exponential distribution has also been used to model the lifetime of a component that fails catastrophically (instantaneously), such as a light bulb; then λ is the failure rate.

Several different exponential pdf's are shown in Figure 5.10. The value of the intercept on the vertical axis is always equal to the value of λ. Note also that all pdf's eventually intersect. (Why?)

The exponential distribution has mean and variance given by

$$E(X) = \frac{1}{\lambda} \quad \text{and} \quad V(X) = \frac{1}{\lambda^2} \tag{5.27}$$

Thus, the mean and standard deviation are equal. The cdf can be exhibited by integrating Equation (5.26) to obtain

$$F(x) = \begin{cases} 0, & x < 0 \\ \int_0^x \lambda e^{-\lambda t} dt = 1 - e^{-\lambda x}, & x \geq 0 \end{cases} \tag{5.28}$$

Example 5.17

Suppose that the life of an industrial lamp, in thousands of hours, is exponentially distributed with failure rate $\lambda = 1/3$ (one failure every 3000 hours, on the average). The probability that the lamp will last longer than its mean life, 3000 hours, is given by $P(X > 3) = 1 - P(X \leq 3) = 1 - F(3)$. Equation (5.28) is used to compute $F(3)$, obtaining

$$P(X > 3) = 1 - (1 - e^{-3/3}) = e^{-1} = 0.368$$

Regardless of the value of λ, this result will always be the same! That is, the probability that an exponential random variable is greater than its mean is 0.368, for any value of λ.

The probability that the industrial lamp will last between 2000 and 3000 hours is computed as

$$P(2 \leq X \leq 3) = F(3) - F(2)$$

Again, from the cdf given by Equation (5.28),

$$\begin{aligned} F(3) - F(2) &= (1 - e^{-3/3}) - (1 - e^{-2/3}) \\ &= -0.368 + 0.513 = 0.145 \end{aligned}$$

One of the most important properties of the exponential distribution is that it is "memoryless," which means that, for all $s \geq 0$ and $t \geq 0$,

$$P(X > s + t | X > s) = P(X > t) \tag{5.29}$$

Let X represent the life of a component (a battery, light bulb, computer chip, laser, etc.) and assume that X is exponentially distributed. Equation (5.29) states that the probability that the component lives for at least $s + t$ hours, given that it has survived s hours, is the same as the initial probability that it lives for at least t hours. If the component is alive at time s (if $X > s$), then the distribution of the remaining amount of time that it survives, namely $X - s$, is the same as the original distribution of a new component. That is, the component does not "remember" that it has already been in use for a time s. A used component is as good as new.

That Equation (5.29) holds is shown by examining the conditional probability

$$P(X > s + t \mid X > s) = \frac{P(X > s + t)}{P(X > s)} \tag{5.30}$$

Equation (5.28) can be used to determine the numerator and denominator of Equation (5.30), yielding

$$\begin{aligned} P(X > s + t \mid X > s) &= \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = e^{-\lambda t} \\ &= P(X > t) \end{aligned}$$

Example 5.18

Find the probability that the industrial lamp in Example 5.17 will last for another 1000 hours, given that it is operating after 2500 hours. This determination can be found using Equations (5.29) and (5.28), as follows:

$$P(X > 3.5 \mid X > 2.5) = P(X > 1) = e^{-1/3} = 0.717$$

Example 5.18 illustrates the *memoryless* property—namely, that a used component that follows an exponential distribution is as good as a new component. The probability that a new component will have

a life greater than 1000 hours is also equal to 0.717. Stated in general, suppose that a component which has a lifetime that follows the exponential distribution with parameter λ is observed and found to be operating at an arbitrary time. Then, the distribution of the remaining lifetime is also exponential with parameter λ. The exponential distribution is the only continuous distribution that has the memoryless property. (The geometric distribution is the only discrete distribution that possesses the memoryless property.)

3. *Gamma distribution.* A function used in defining the gamma distribution is the gamma function, which is defined for all $\beta > 0$ as

$$\Gamma(\beta) = \int_0^\infty x^{\beta-1} e^{-x} dx \tag{5.31}$$

By integrating Equation (5.31) by parts, it can be shown that

$$\Gamma(\beta) = (\beta - 1)\Gamma(\beta - 1) \tag{5.32}$$

If β is an integer, then, by using $\Gamma(1) = 1$ and applying Equation (5.32), it can be seen that

$$\Gamma(\beta) = (\beta - 1)! \tag{5.33}$$

The gamma function can be thought of as a generalization of the factorial notion to all positive numbers, not just integers.

A random variable X is gamma distributed with parameters β and θ if its pdf is given by

$$f(x) = \begin{cases} \dfrac{\beta\theta}{\Gamma(\beta)}(\beta\theta x)^{\beta-1} e^{-\beta\theta x}, & x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.34}$$

β is called the shape parameter, and θ is called the scale parameter. Several gamma distributions for $\theta = 1$ and various values of β are shown in Figure 5.10a.

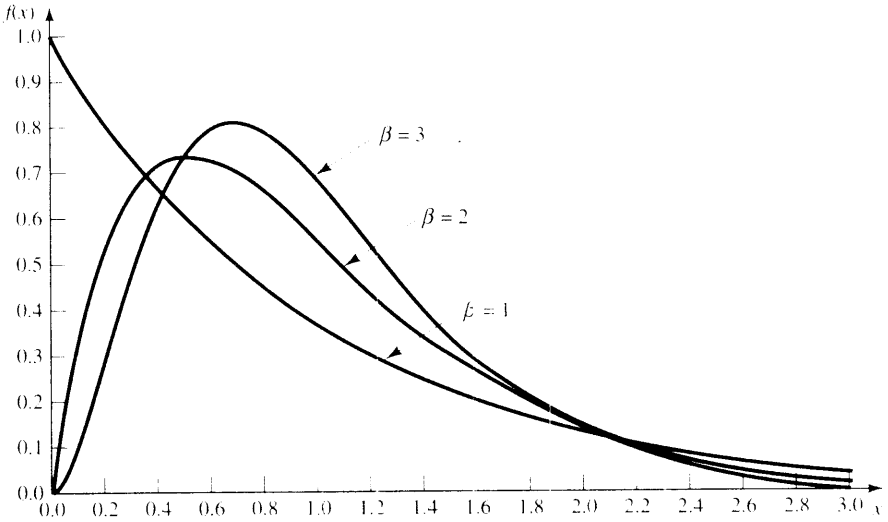


Figure 5.10a

The mean and variance of the gamma distribution are given by

$$E(X) = \frac{1}{\theta} \tag{5.35}$$

and

$$V(X) = \frac{1}{\beta\theta^2} \tag{5.36}$$

The cdf of X is given by

$$F(x) = \begin{cases} 1 - \int_x^\infty \frac{\beta\theta}{\Gamma(\beta)}(\beta\theta t)^{\beta-1} e^{-\beta\theta t} dt, & x > 0 \\ 0, & x \le 0 \end{cases} \tag{5.37}$$

When β is an integer, the gamma distribution is related to the exponential distribution in the following manner: If the random variable, X, is the sum of β independent, exponentially distributed random variables, each with parameter $\beta\theta$, then X has a gamma distribution with parameters β and θ. Thus, if

$$X = X_1 + X_2 + \cdots + X_\beta \tag{5.38}$$

where the pdf of X_j is given by

$$g(x_j) = \begin{cases} (\beta\theta)e^{-\beta\theta x_j}, & x \ge 0 \\ 0, & \text{otherwise} \end{cases}$$

and the X_j are mutually independent, then X has the pdf given in Equation (5.34). Note that, when $\beta = 1$, an exponential distribution results. This result follows from Equation (5.38) or from letting $\beta = 1$ in Equation (5.34).

4. *Erlang distribution.* The pdf given by Equation (5.34) is often referred to as the Erlang distribution of order (or number of phases) k when $\beta = k$, an integer. Erlang was a Danish telephone engineer who was an early developer of queueing theory. The Erlang distribution could arise in the following context: Consider a series of k stations that must be passed through in order to complete the servicing of a customer. An additional customer cannot enter the first station until the customer in process has negotiated all the stations. Each station has an exponential distribution of service time with parameter $k\theta$. Equations (5.35) and (5.36), which state the mean and variance of a gamma distribution, are valid regardless of the value of β. However, when $\beta = k$, an integer, Equation (5.38) may be used to derive the mean of the distribution in a fairly straightforward manner. The expected value of the sum of random variables is the sum of the expected value of each random variable. Thus,

$$E(X) = E(X_1) + E(X_2) + \cdots + E(X_k)$$

The expected value of each of the exponentially distributed X_j is given by $1/k\theta$. Thus,

$$E(X) = \frac{1}{k\theta} + \frac{1}{k\theta} + \cdots + \frac{1}{k\theta} = \frac{1}{\theta}$$

If the random variables X_j are independent, the variance of their sum is the sum of the variances, or

$$V(X) = \frac{1}{(k\theta)^2} + \frac{1}{(k\theta)^2} + \cdots + \frac{1}{(k\theta)^2} = \frac{1}{k\theta^2}$$

When $\beta = k$, a positive integer, the cdf given by Equation (5.37) may be integrated by parts, giving

$$F(x) = \begin{cases} 1 - \sum_{i=0}^{k-1} \frac{e^{-k\theta x}(k\theta x)^i}{i!}, & x > 0 \\ 0, & x \le 0 \end{cases} \tag{5.39}$$

which is the sum of Poisson terms with mean $\alpha = k\theta x$. Tables of the cumulative Poisson distribution may be used to evaluate the cdf when the shape parameter is an integer.

Example 5.19

A college professor of electrical engineering is leaving home for the summer, but would like to have a light burning at all times to discourage burglars. The professor rigs up a device that will hold two light bulbs. The device will switch the current to the second bulb if the first bulb fails. The box in which the light bulbs are packaged says, "Average life 1000 hours, exponentially distributed." The professor will be gone 90 days (2160 hours). What is the probability that a light will be burning when the summer is over and the professor returns?

The probability that the system will operate at least x hours is called the reliability function $R(x)$:

$$R(x) = 1 - F(x)$$

In this case, the total system lifetime is given by Equation (5.38) with $\beta = k = 2$ bulbs and $k\theta = 1/1000$ per hour, so $\theta = 1/2000$ per hour. Thus, $F(2160)$ can be determined from Equation (5.39) as follows:

$$F(2160) = 1 - \sum_{i=0}^{1} \frac{e^{-(2)(1/2000)(2160)}[(2)(1/2000)(2160)]^i}{i!}$$

$$= 1 - e^{-2.16} \sum_{i=1}^{1} \frac{(2.16)^i}{i!} = 0.636$$

Therefore, the chances are about 36% that a light will be burning when the professor returns.

Example 5.20

A medical examination is given in three stages by a physician. Each stage is exponentially distributed with a mean service time of 20 minutes. Find the probability that the exam will take 50 minutes or less. Also, compute the expected length of the exam. In this case, $k = 3$ stages and $k\theta = 1/20$, so that $\theta = 1/60$ per minute. Thus, $F(50)$ can be calculated from Equation (5.39) as follows:

$$F(50) = 1 - \sum_{i=0}^{2} \frac{e^{-(3)(1/60)(50)}[(3)(1/60)(50)]^i}{i!}$$

$$= 1 - \sum_{i=0}^{2} \frac{e^{-5/2}(5/2)^i}{i!}$$

The cumulative Poisson distribution, shown in Table A.4, can be used to calculate that

$$F(50) = 1 - 0.543 = 0.457$$

The probability is 0.457 that the exam will take 50 minutes or less. The expected length of the exam is found from Equation (5.35):

$$E(X) = \frac{1}{\theta} = \frac{1}{1/60} = 60 \text{ minutes}$$

In addition, the variance of X is $V(X) = 1/\beta\theta^2 = 1200$ minutes^2—incidentally, the mode of the Erlang distribution is given by

$$\text{Mode} = \frac{k-1}{k\theta} \tag{5.40}$$

Thus, the modal value in this example is

$$\text{Mode} = \frac{3-1}{3(1/60)} = 40 \text{ minutes}$$

5. *Normal distribution.* A random variable X with mean $-\infty < \mu < \infty$ and variance $\sigma^2 > 0$ has a normal distribution if it has the pdf

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], \quad -\infty < x < \infty \tag{5.41}$$

The normal distribution is used so often that the notation $X \sim N(\mu, \sigma^2)$ has been adopted by many authors to indicate that the random variable X is normally distributed with mean μ and variance σ^2. The normal pdf is shown in Figure 5.11.

Some of the special properties of the normal distribution are listed here:

1. $\lim_{x \to -\infty} f(x) = 0$ and $\lim_{x \to \infty} f(x) = 0$; the value of $f(x)$ approaches zero as x approaches negative infinity and, similarly, as x approaches positive infinity.
2. $f(\mu - x) = f(\mu + x)$; the pdf is symmetric about μ.
3. The maximum value of the pdf occurs at $x = \mu$; the mean and mode are equal.

The cdf for the normal distribution is given by

$$F(x) = P(X \le x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right] dt \tag{5.42}$$

It is not possible to evaluate Equation (5.42) in closed form. Numerical methods could be used, but it appears that it would be necessary to evaluate the integral for each pair (μ, σ^2). However, a transformation of

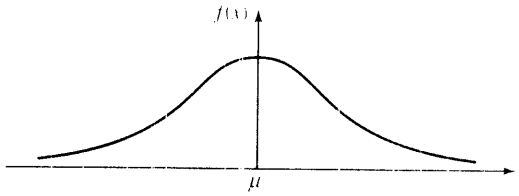


Figure 5.11 pdf of the normal distribution.

variables, $z = (t - \mu)/\sigma$, allows the evaluation to be independent of μ and σ. If $X \sim N(\mu, \sigma^2)$, let $Z = (X - \mu)/\sigma$ to obtain

$$F(x) = P(X \le x) = P\left(Z \le \frac{x - \mu}{\sigma}\right)$$
$$= \int_{-\infty}^{(x-\mu)/\sigma} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$
$$= \int_{-\infty}^{(x-\mu)/\sigma} \phi(z) dz = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

The pdf

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}, \quad -\infty < z < \infty \tag{5.44}$$

is the pdf of a normal distribution with mean zero and variance 1. Thus, $Z \sim N(0, 1)$ and it is said that Z has a standard normal distribution. The standard normal distribution is shown in Figure 5.12. The cdf for the standard normal distribution is given by

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \tag{5.45}$$

Equation (5.45) has been widely tabulated. The probabilities $\Phi(z)$ for $Z \ge 0$ are given in Table A.3. Several examples are now given that indicate how Equation (5.43) and Table A.3 are used.

Example 5.21

Suppose that it is known that $X \sim N(50, 9)$. Compute $F(56) = P(X \le 56)$. Using Equation (5.43) get

$$F(56) = \Phi\left(\frac{56 - 50}{3}\right) = \Phi(2) = 0.9772$$

from Table A.3. The intuitive interpretation is shown in Figure 5.13. Figure 5.13(a) shows the pdf of $X \sim N(50, 9)$ with the specific value, $x_0 = 56$, marked. The shaded portion is the desired probability. Figure 5.13(b) shows the standard normal distribution or $Z \sim N(0, 1)$ with the value 2 marked; $x_0 = 56$ is 2σ (where $\sigma = 3$) greater than the mean. It is helpful to make both sketches such as those in Figure 5.13 to avoid confusion in figuring out required probabilities.

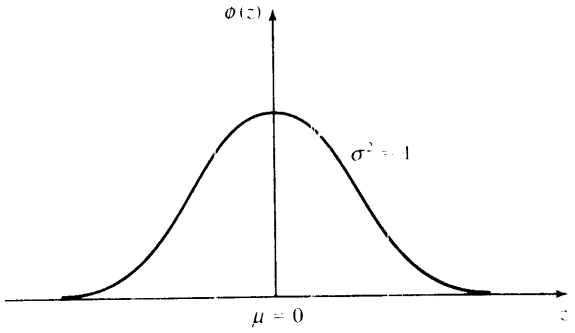


Figure 5.12 pdf of the standard normal distribution.

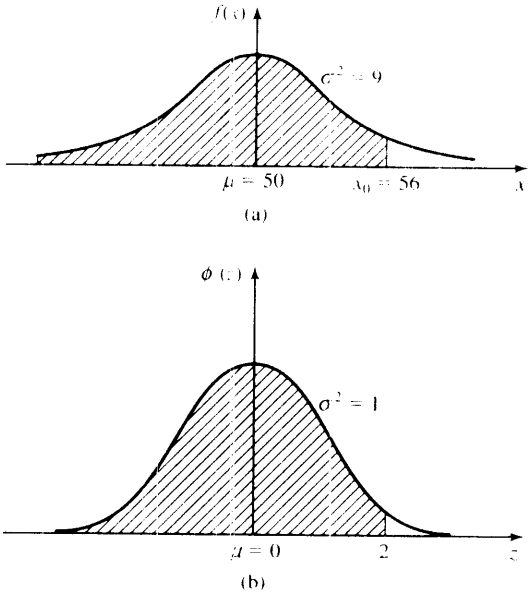


Figure 5.13 Transforming to the standard normal distribution.

Example 5.22

The time in hours required to load an oceangoing vessel, X, is distributed as $N(12,4)$. The probability that the vessel will be loaded in less than 10 hours is given by $F(10)$, where

$$F(10) = \Phi\left(\frac{10-12}{2}\right) = \Phi(-1) = 0.1587$$

The value of $\Phi(-1) = 0.1587$ is looked up in Table A.3 by using the symmetry property of the normal distribution. Note that $\Phi(1) = 0.8413$. The complement of 0.8413, or 0.1587, is contained in the tail, the shaded portion of the standard normal distribution shown in Figure 5.14(a). In Figure 5.14(b), the symmetry property is used to work out the shaded region to be $\Phi(-1) = 1 - \Phi(1) = 0.1587$. [From this logic, it can be seen that $\Phi(2) = 0.9772$ and $\Phi(-2) = 1 - \Phi(2) = 0.0228$. In general, $\Phi(-x) = 1 - \Phi(x)$.]

The probability that 12 or more hours will be required to load the ship can also be discovered by inspection, by using the symmetry property of the normal pdf and the mean as shown by Figure 5.15. The shaded portion of Figure 5.15(a) shows the problem as originally stated [i.e., evaluate $P(X < 12)$]. Now, $P(X > 12) = 1 - F(12)$. The standardized normal in Figure 5.15(b) is used to evaluate $F(12) = \Phi(0) = 0.50$. Thus, $P(X > 12) = 1 - 0.50 = 0.50$. [The shaded portions in both Figure 5.15(a) and (b) contain 0.50 of the area under the normal pdf.]

The probability that between 10 and 12 hours will be required to load a ship is given by

$$P(10 \le X \le 12) = F(12) - F(10) = 0.5000 - 0.1587 = 0.3413$$

using earlier results presented in this example. The desired area is shown in the shaded portion of Figure 5.16(a). The equivalent problem shown in terms of the standardized normal distribution is shown in Figure 5.16(b). The probability statement is $F(12) - F(10) = \Phi(0) - \Phi(-1) = 0.5000 - 0.1587 = 0.3413$, from Table A.3.

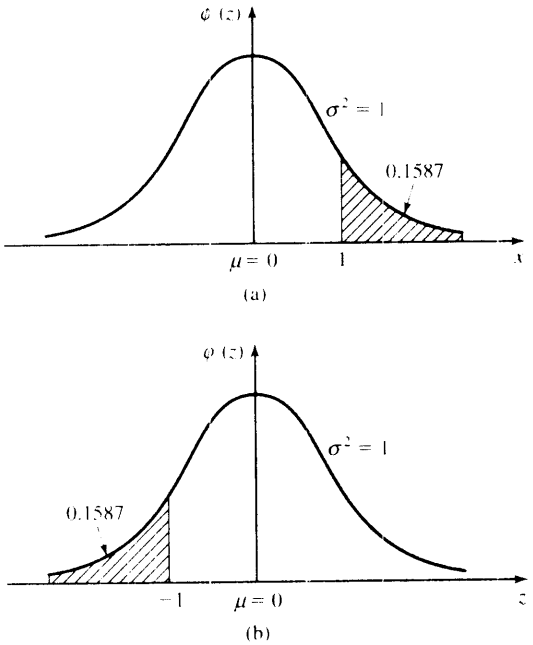


Figure 5.14 Using the symmetry property of the normal distribution.

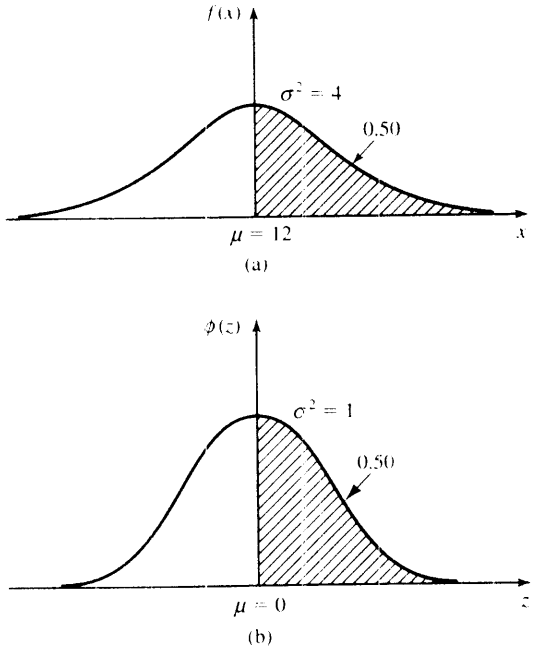


Figure 5.15 Evaluation of probability by inspection.

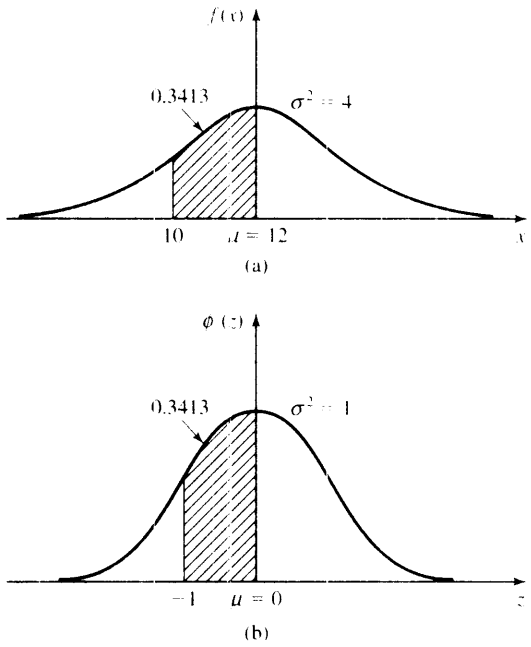


Figure 5.16 Transformation to standard normal for vessel-loading problem.

Example 5.23

The time to pass through a queue to begin self-service at a cafeteria has been found to be $N(15, 9)$. The probability that an arriving customer waits between 14 and 17 minutes is computed as follows:

$$P(14 \le X \le 17) = F(17) - F(14) = \Phi\left(\frac{17-15}{3}\right) - \Phi\left(\frac{14-15}{3}\right)$$
$$= \Phi(0.667) - \Phi(-0.333)$$

The shaded area shown in Figure 5.17(a) represents the probability $F(17) - F(14)$. The shaded area shown in Figure 5.17(b) represents the equivalent probability, $\Phi(0.667) - \Phi(-0.333)$, for the standardized normal distribution. From Table A.3, $\Phi(0.667) = 0.7476$. Now, $\Phi(-0.333) = 1 - \Phi(0.333) = 1 - 0.6304 = 0.3696$. Thus, $\Phi(0.667) - \Phi(-0.333) = 0.3780$. The probability is 0.3780 that the customer will pass through the queue in a time between 14 and 17 minutes.

Example 5.24

Lead-time demand, X, for an item is approximated by a normal distribution having mean 25 and variance 9. It is desired to compute the value for lead time that will be exceeded only 5% of the time. Thus, the problem is to find x_0 such that $P(X > x_0) = 0.05$, as shown by the shaded area in Figure 5.18(a). The equivalent problem is shown as the shaded area in Figure 5.18(b). Now,

$$P(X > x_0) = P\left(Z > \frac{x_0 - 25}{3}\right) = 1 - \Phi\left(\frac{x_0 - 25}{3}\right) = 0.05$$

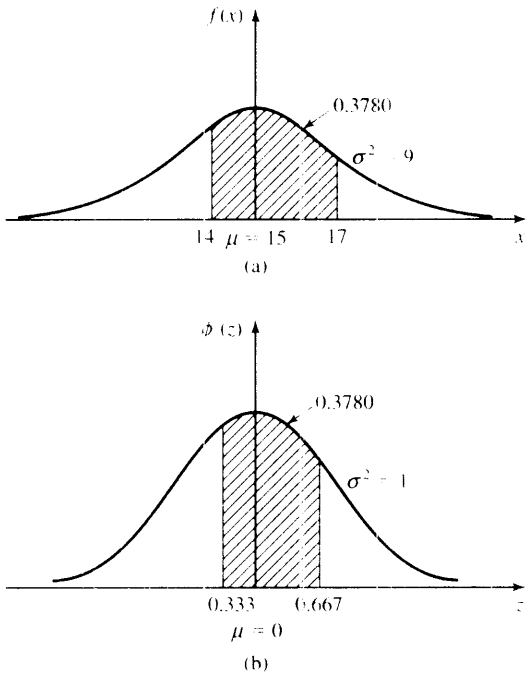


Figure 5.17 Transformation to standard normal for cafeteria problem.

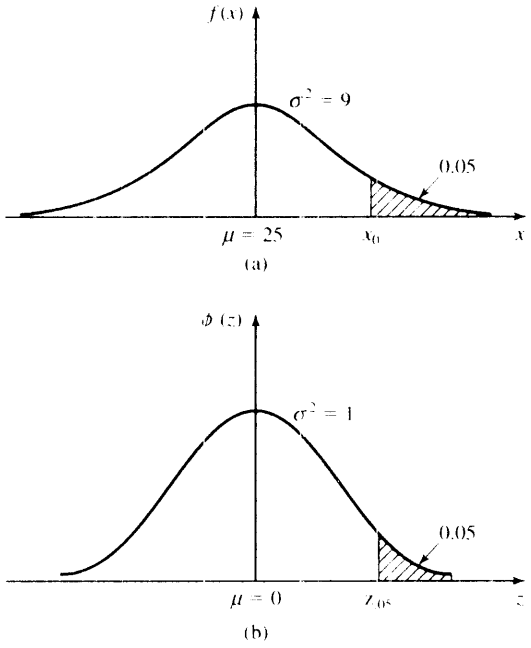


Figure 5.18 Finding x_0 for lead-time–demand problem.

or, equivalently,

$$\Phi\left(\frac{x_0 - 25}{3}\right) = 0.95$$

From Table A.3, it can be seen that $\Phi(1.645) = 0.95$. Thus, x_0 can be found by solving

$$\frac{x_0 - 25}{3} = 1.645$$

or

$$x_0 = 29.935$$

Therefore, in only 5% of the cases will demand during lead time exceed available inventory if an order to purchase is made when the stock level reaches 30.

6. *Weibull distribution.* The random variable X has a Weibull distribution if its pdf has the form

$$f(x) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{x-\nu}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{x-\nu}{\alpha}\right)^{\beta}\right], & x \geq \nu \\ 0, & \text{otherwise} \end{cases} \tag{5.46}$$

The three parameters of the Weibull distribution are ν $(-\infty < \nu < \infty)$, which is the location parameter; α $(\alpha > 0)$, which is the scale parameter; and β $(\beta > 0)$, which is the shape parameter. When $\nu = 0$, the Weibull pdf becomes

$$f(x) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^{\beta}\right], & x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.47}$$

Figure 5.19 shows several Weibull densities when $\nu = 0$ and $\alpha = 1$. When $\beta = 1$, the Weibull distribution is reduced to

$$f(x) = \begin{cases} \frac{1}{\alpha} e^{-x/\alpha}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

which is an exponential distribution with parameter $\lambda = 1/\alpha$.

The mean and variance of the Weibull distribution are given by the following expressions:

$$E(X) = \nu + \alpha\Gamma\left(\frac{1}{\beta} + 1\right) \tag{5.48}$$

$$V(X) = \alpha^2\left[\Gamma\left(\frac{2}{\beta} + 1\right) - \left[\Gamma\left(\frac{1}{\beta} + 1\right)\right]^2\right] \tag{5.49}$$

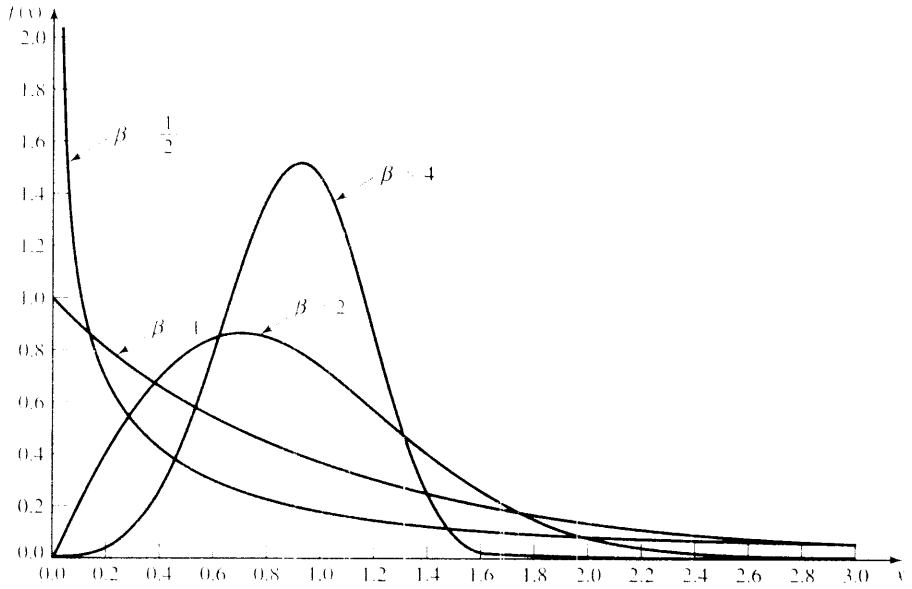


Figure 5.19 Weibull pdfs for $\nu = 0; \alpha = \frac{1}{2}; \beta = \frac{1}{2}, 1, 2, 4.$

where $\Gamma(\cdot)$ is defined by Equation (5.31). Thus, the location parameter, ν, has no effect on the variance; however, the mean is increased or decreased by ν. The cdf of the Weibull distribution is given by

$$F(x) \begin{cases} 0, & x < \nu \\ 1 - \exp\left[-\left(\frac{x-\nu}{\alpha}\right)^{\beta}\right], & x \geq \nu \end{cases} \tag{5.50}$$

Example 5.25

The time to failure for a component screen is known to have a Weibull distribution with $\nu = 0$, $\beta = 1/3$, and $\alpha = 200$ hours. The mean time to failure is given by Equation (5.48) as

$$E(X) = 200\Gamma(3 + 1) = 200(3!) = 1200 \text{ hours}$$

The probability that a unit fails before 2000 hours is computed from Equation (5.50) as

$$F(2000) = 1 - \exp\left[-\left(\frac{2000}{200}\right)^{1/3}\right]$$
$$= 1 - e^{-\sqrt[3]{10}} = 1 - e^{-2.15} = 0.884$$

Example 5.26

The time it takes for an aircraft to land and clear the runway at a major international airport has a Weibull distribution with $\nu = 1.34$ minutes, $\beta = 0.5$, and $\alpha = 0.04$ minute. Find the probability that an incoming

airplane will take more than 1.5 minutes to land and clear the runway. In this case $P(X > 1.5)$ is computed as follows:

$$\begin{aligned} P(X \le 1.5) &= F(1.5) \\ &= 1 - \exp\left[-\left(\frac{1.5-1.34}{0.04}\right)^{0.5}\right] \\ &= 1 - e^{-2} = 1 - 0.135 = 0.865 \end{aligned}$$

Therefore, the probability that an aircraft will require more than 1.5 minutes to land and clear the runway is 0.135.

7. *Triangular distribution.* A random variable X has a triangular distribution if its pdf is given by

$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)}, & a \le x \le b \\ \dfrac{2(c-x)}{(c-b)(c-a)}, & b < x \le c \\ 0, & \text{elsewhere} \end{cases} \tag{5.51}$$

where $a \le b \le c$. The mode occurs at $x = b$. A triangular pdf is shown in Figure 5.20. The parameters (a, b, c) can be related to other measures, such as the mean and the mode, as follows:

$$E(X) = \frac{a+b+c}{3} \tag{5.52}$$

From Equation (5.52) the mode can be determined as

$$\text{Mode} = b = 3E(X) - (a + c) \tag{5.53}$$

Because $a \le b \le c$,

$$\frac{2a+c}{3} \le E(X) \le \frac{a+2c}{3}$$

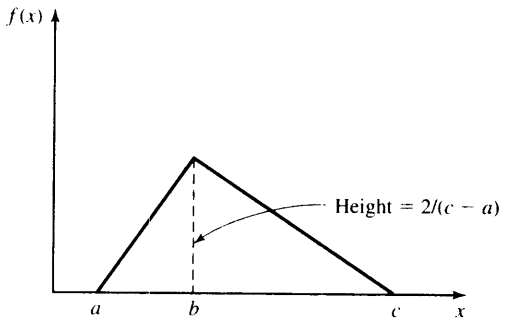


Figure 5.20 pdf of the triangular distribution.

The mode is used more often than the mean to characterize the triangular distribution. As is shown in Figure 5.20, its height is $2/(c-a)$ above the x axis. The variance, $V(X)$, of the triangular distribution is left as an exercise for the student. The cdf for the triangular distribution is given by

$$F(x) = \begin{cases} 0, & x \le a \\ \dfrac{(x-a)^2}{(b-a)(c-a)}, & a < x \le b \\ 1 - \dfrac{(c-x)^2}{(c-b)(c-a)}, & b < x \le c \\ 1, & x > c \end{cases} \tag{5.54}$$

Example 5.27

The central processing unit requirements, for programs that will execute, have a triangular distribution with $a = 0.05$ millisecond, $b = 1.1$ milliseconds, and $c = 6.5$ milliseconds. Find the probability that the CPU requirement for a random program is 2.5 milliseconds or less. The value of $F(2.5)$ is from the portion of the cdf in the interval (0.05, 1.1) plus that portion in the interval (1.1, 2.5). By using Equation (5.54), both portions can be addressed at one time, to yield

$$F(2.5) = 1 - \frac{(6.5-2.5)^2}{(6.5-0.05)(6.5-1.1)} = 0.541$$

Thus, the probability is 0.541 that the CPU requirement is 2.5 milliseconds or less.

Example 5.28

An electronic sensor evaluates the quality of memory chips, rejecting those that fail. Upon demand, the sensor will give the minimum and maximum number of rejects during each hour of production over the past 24 hours. The mean is also given. Without further information, the quality control department has assumed that the number of rejected chips can be approximated by a triangular distribution. The current dump of data indicates that the minimum number of rejected chips during any hour was zero, the maximum was 10, and the mean was 4. Given that $a = 0$, $c = 10$, and $E(X) = 4$, the value of b can be found from Equation (5.53):

$$b = 3(4) - (0 + 10) = 2$$

The height of the mode is $2/(10 - 0) = 0.2$. Thus, Figure 5.21 can be drawn.

The median is the point at which 0.5 of the area is to the left and 0.5 is to the right. The median in this example is 3.7, also shown on Figure 5.21. Finding the median of the triangular distribution requires an initial location of the value to the left or to the right of the mode. The area to the left of the mode is computed from Equation (5.54) as

$$F(2) = \frac{2^2}{20} = 0.2$$

Thus, the median is between b and c. Setting $F(x) = 0.5$ in Equation (5.54) and solving for x = median yields

$$0.5 = 1 - \frac{(10-x)^2}{(10)(8)}$$

with

$$x = 3.7$$

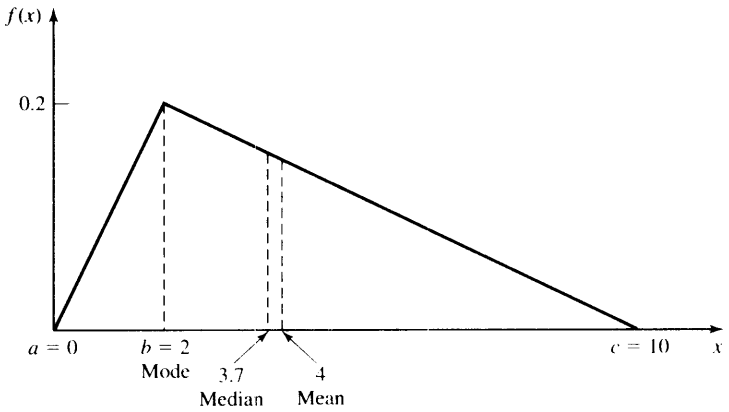


Figure 5.21 Mode, median, and mean for triangular distribution.

This example clearly shows that the mean, mode, and median are not necessarily equal.

8. *Lognormal distribution.* A random variable X has a lognormal distribution if its pdf is given by

$$f(x) = \begin{cases} \dfrac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\dfrac{(\ln x - \mu)^2}{2\sigma^2}\right], & x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.55}$$

where $\sigma^2 > 0$ The mean and variance of a lognormal random variable are

$$E(X) = e^{\mu + \sigma^2/2} \tag{5.56}$$

$$V(X) = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1) \tag{5.57}$$

Three lognormal pdf's, all having mean 1, but variances 1/2, 1, and 2, are shown in Figure 5.22.

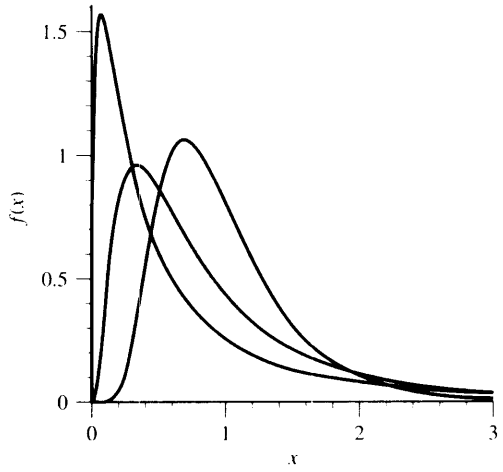


Figure 5.22 pdf of the lognormal distribution.

Notice that the parameters μ and σ^2 are not the mean and variance of the lognormal. These parameters come from the fact that when Y has a $N(\mu, \sigma^2)$ distribution then $X = e^Y$ has a lognormal distribution with parameters μ and σ^2. If the mean and variance of the lognormal are known to be μ_L and σ_L^2, respectively, then the parameters μ and σ^2 are given by

$$\mu = \ln\left(\frac{\mu_L^2}{\sqrt{\mu_L^2 + \sigma_L^2}}\right) \tag{5.58}$$

$$\sigma^2 = \ln\left(\frac{\mu_L^2 + \sigma_L^2}{\mu_L^2}\right) \tag{5.59}$$

Example 5.29

The rate of return on a volatile investment is modeled as having a lognormal distribution with mean 20% and standard deviation 5%. Compute the parameters for the lognormal distribution. From the information given, we have $\mu_L = 20$ and $\sigma_L^2 = 5^2$. Thus, from Equations (5.58) and (5.59),

$$\mu = \ln\left(\frac{20^2}{\sqrt{20^2 + 5^2}}\right) \doteq 2.9654$$

$$\sigma^2 = \ln\left(\frac{20^2 + 5^2}{20^2}\right) \doteq 0.06$$

9. *Beta distribution.* A random variable X is beta-distributed with parameters $\beta_1 > 0$ and $\beta_2 > 0$ if its pdf is given by

$$f(x) = \begin{cases} \dfrac{x^{\beta_1 - 1}(1-x)^{\beta_2 - 1}}{B(\beta_1, \beta_2)}, & 0 < x < 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.60}$$

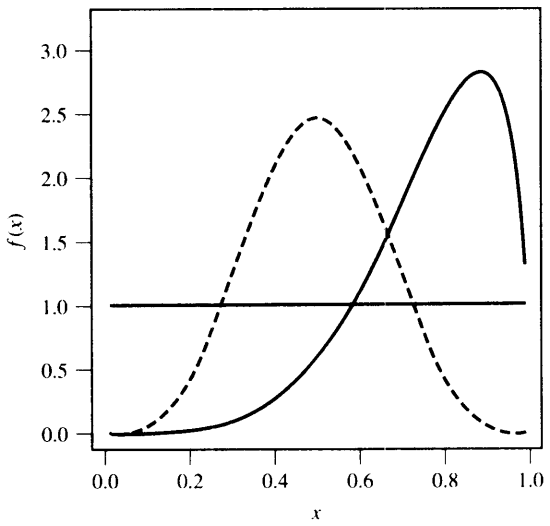


Figure 5.23 The pdf's for several beta distributions.

where $B(\beta_1, \beta_2) = \Gamma(\beta_1)\Gamma(\beta_2)/\Gamma(\beta_1 + \beta_2)$. The cdf of the beta does not have a closed form in general.

The beta distribution is very flexible and has a finite range from 0 to 1, as shown in Figure 5.23. In practice, we often need a beta distribution defined on a different range, say (a, b), with $a < b$, rather than (0, 1). This is easily accomplished by defining a new random variable

$$Y = a + (b - a)X$$

The mean and variance of Y are given by

$$a + (b-a)\left(\frac{\beta_1}{\beta_1 + \beta_2}\right) \tag{5.61}$$

and

$$(b-a)^2\left(\frac{\beta_1\beta_2}{(\beta_1 + \beta_2)^2(\beta_1 + \beta_2 + 1)}\right) \tag{5.62}$$

5.5 POISSON PROCESS

Consider random events such as the arrival of jobs at a job shop, the arrival of e-mail to a mail server, the arrival of boats to a dock, the arrival of calls to a call center, the breakdown of machines in a large factory, and so on. These events may be described by a counting function $N(t)$ defined for all $t \geq 0$. This counting function will represent the number of events that occurred in $[0, t]$. Time zero is the point at which the observation began, regardless of whether an arrival occurred at that instant. For each interval $[0, t]$, the value $N(t)$ is an observation of a random variable where the only possible values that can be assumed by $N(t)$ are the integers 0, 1, 2,....

The counting process, $\{N(t), t \geq 0\}$, is said to be a Poisson process with mean rate λ if the following assumptions are fulfilled:

1. Arrivals occur one at a time.
2. $\{N(t), t \geq 0\}$ has stationary increments: The distribution of the number of arrivals between t and $t + s$ depends only on the length of the interval s, not on the starting point t. Thus, arrivals are completely at random without rush or slack periods.
3. $\{N(t), t \geq 0\}$ has independent increments: The number of arrivals during nonoverlapping time intervals are independent random variables. Thus, a large or small number of arrivals in one time interval has no effect on the number of arrivals in subsequent time intervals. Future arrivals occur completely at random, independent of the number of arrivals in past time intervals.

If arrivals occur according to a Poisson process, meeting the three preceding assumptions, it can be shown that the probability that $N(t)$ is equal to n is given by

$$P[N(t) = n] = \frac{e^{-\lambda t}(\lambda t)^n}{n!} \quad \text{for } t \geq 0 \text{ and } n = 0, 1, 2, \ldots \tag{5.63}$$

Comparing Equation (5.63) to Equation (5.19), it can be seen that $N(t)$ has the Poisson distribution with parameter $\alpha = \lambda t$. Thus, its mean and variance are given by

$$E[N(t)] = \alpha = \lambda t = V[N(t)]$$

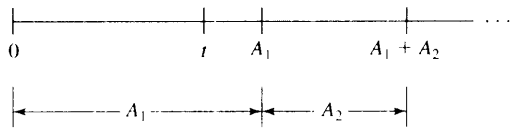


Figure 5.24 Arrival process.

For any times s and t such that $s < t$, the assumption of stationary increments implies that the random variable $N(t) - N(s)$, representing the number of arrivals in the interval from s to t, is also Poisson-distributed with mean $\lambda(t - s)$. Thus,

$$P[N(t) - N(s) = n] = \frac{e^{-\lambda(t-s)}[\lambda(t-s)]^n}{n!} \quad \text{for } n = 0, 1, 2, \ldots$$

and

$$E[N(t) - N(s)] = \lambda(t - s) = V[N(t) - N(s)]$$

Now, consider the time at which arrivals occur in a Poisson process. Let the first arrival occur at time A_1, the second occur at time $A_1 + A_2$, and so on, as shown in Figure 5.24. Thus, $A_1, A_2, \ldots$ are successive interarrival times. The first arrival occurs after time t if and only if there are no arrivals in the interval $[0, t]$, so it is seen that

$$\{A_1 > t\} = \{N(t) = 0\}$$

and, therefore,

$$P(A_1 > t) = P[N(t) = 0] = e^{-\lambda t}$$

the last equality following from Equation (5.63). Thus, the probability that the first arrival will occur in $[0, t]$ is given by

$$P(A_1 \le t) = 1 - e^{-\lambda t}$$

which is the cdf for an exponential distribution with parameter λ. Hence, A_1 is distributed exponentially with mean $E(A_1) = 1/\lambda$. It can also be shown that all interarrival times, $A_1, A_2, \ldots$, are exponentially distributed and independent with mean $1/\lambda$. As an alternative definition of a Poisson process, it can be shown that, if interarrival times are distributed exponentially and independently, then the number of arrivals by time t, say $N(t)$, meets the three previously mentioned assumptions and, therefore, is a Poisson process.

Recall that the exponential distribution is memoryless—that is, the probability of a future arrival in a time interval of length s is independent of the time of the last arrival. The probability of the arrival depends only on the length of the time interval, s. Thus, the memoryless property is related to the properties of independent and stationary increments of the Poisson process.

Additional readings concerning the Poisson process may be obtained from many sources, including Parzen [1999], Feller [1968], and Ross [2002].

Example 5.30

The jobs at a machine shop arrive according to a Poisson process with a mean of $\lambda = 2$ jobs per hour. Therefore, the interarrival times are distributed exponentially, with the expected time between arrivals being $E(A) = 1/\lambda = \frac{1}{2}$ hour.

5.5.1 Properties of a Poisson Process

Several properties of the Poisson process, discussed by Ross [2002] and others, are useful in discrete-system simulation. The first of these properties concerns random splitting. Consider a Poisson process $\{N(t), t \geq 0\}$ having rate λ, as represented by the left side of Figure 5.25.

Suppose that, each time an event occurs, it is classified as either a type I or a type II event. Suppose further that each event is classified as a type I event with probability p and type II event with probability $1-p$, independently of all other events.

Let $N_1(t)$ and $N_2(t)$ be random variables that denote, respectively, the number of type I and type II events occurring in $[0, t]$. Note that $N(t) = N_1(t) + N_2(t)$. It can be shown that $N_1(t)$ and $N_2(t)$ are both Poisson processes having rates λp and $\lambda(1-p)$, as shown in Figure 5.25. Furthermore, it can be shown that the two processes are independent.

Example 5.31: (Random Splitting)

Suppose that jobs arrive at a shop in accordance with a Poisson process having rate λ. Suppose further that each arrival is marked "high priority" with probability 1/3 and "low priority" with probability 2/3. Then a type I event would correspond to a high-priority arrival and a type II event would correspond to a low-priority arrival. If $N_1(t)$ and $N_2(t)$ are as just defined, both variables follow the Poisson process, with rates $\lambda/3$ and $2\lambda/3$, respectively.

Example 5.32

The rate in Example 5.31 is $\lambda = 3$ per hour. The probability that no high-priority jobs will arrive in a 2-hour period is given by the Poisson distribution with parameter $\alpha = \lambda p t = 2$. Thus,

$$P(0) = \frac{e^{-2} 2^0}{0!} = 0.135$$

Now, consider the opposite situation from random splitting, namely the pooling of two arrival streams. The process of interest is illustrated in Figure 5.26. It can be shown that, if $N_i(t)$ are random variables representing independent Poisson processes with rates λ_i, for $i = 1$ and 2, then $N(t) = N_1(t) + N_2(t)$ is a Poisson process with rate $\lambda_1 + \lambda_2$.

Example 5.33: (Pooled Process)

A Poisson arrival stream with $\lambda_1 = 10$ arrivals per hour is combined (or pooled) with a Poisson arrival stream with $\lambda_2 = 17$ arrivals per hour. The combined process is a Poisson process with $\lambda = 27$ arrivals per hour.

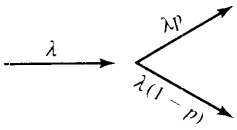


Figure 5.25 Random splitting.

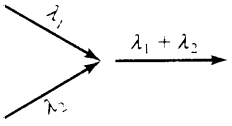


Figure 5.26 Pooled process.

5.5.2 Nonstationary Poisson Process

If we keep the Poisson Assumptions 1 and 3, but drop Assumption 2 (stationary increments) then we have a *nonstationary Poisson process* (NSPP), which is characterized by $\lambda(t)$, the arrival rate at time t. The NSPP is useful for situations in which the arrival rate varies during the period of interest, including meal times for restaurants, phone calls during business hours, and orders for pizza delivery around 6 P.M.

The key to working with a NSPP is the expected number of arrivals by time t, denoted by

$$\Lambda(t) = \int_0^t \lambda(s)\,ds$$

To be useful as an arrival-rate function, $\lambda(t)$ must be nonnegative and integrable. For a stationary Poisson process with rate λ we have $\Lambda(t) = \lambda t$, as expected.

Let $T_1, T_2, \ldots$ be the arrival times of stationary Poisson process $N(t)$ with $\lambda = 1$, and let $\mathcal{T}_1, \mathcal{T}_2, \ldots$ be the arrival times for a NSPP $\mathcal{N}(t)$ with arrival rate $\lambda(t)$. The fundamental relationship for working with NSPPs is the following:

$$T_i = \Lambda(\mathcal{T}_i)$$
$$\mathcal{T}_i = \Lambda^{-1}(T_i)$$

In words, an NSPP can be transformed into a stationary Poisson process with arrival rate 1, and a stationary Poisson process with arrival rate 1 can be transformed into an NSPP with rate $\lambda(t)$, and the transformation in both cases is related to $\Lambda(t)$.

Example 5.34

Suppose that arrivals to a Post Office occur at a rate of 2 per minute from 8 A.M. until 12 P.M., then drop to 1 every 2 minutes until the day ends at 4 P.M. What is the probability distribution of the number of arrivals between 11 A.M. and 2 P.M?

Let time $t = 0$ correspond to 8 A.M. Then this situation could be modeled as a NSPP $\mathcal{N}(t)$ with rate function

$$\lambda(t) = \begin{cases} 2, & 0 \le t < 4 \\ \dfrac{1}{2}, & 4 \le t \le 8 \end{cases}$$

The expected number of arrivals by time t is therefore

$$\Lambda(t) = \begin{cases} 2t, & 0 \le t < 4 \\ \dfrac{t}{2} + 6, & 4 \le t \le 8 \end{cases}$$

Notice that computing the expected number of arrivals for $4 \le t \le 8$ requires that the integration be done in two parts:

$$\Lambda(t) = \int_0^t \lambda(s)\,ds = \int_0^4 2\,ds + \int_4^t \tfrac{1}{2}\,ds = \frac{t}{2} + 6$$

Since 2 P.M. and 11 A.M. correspond to times 6 and 3, respectively, we have

$$
\begin{aligned}
P[\mathcal{N}(6)-\mathcal{N}(3)=k] &= P[N(\Lambda(6))-N(\Lambda(3))=k] \\
&= P[N(9)-N(6)=k] \\
&= \frac{e^{9-6}(9-6)^k}{k!} \\
&= \frac{e^3(3)^k}{k!}
\end{aligned}
$$

where $N(t)$ is a stationary Poisson process with arrival rate 1.

5.6 EMPIRICAL DISTRIBUTIONS

An empirical distribution, which may be either discrete or continuous in form, is a distribution whose parameters are the observed values in a sample of data. This is in contrast to parametric distribution families (such as the exponential, normal, or Poisson), which are characterized by specifying a small number of parameters such as the mean and variance. An empirical distribution may be used when it is impossible or unnecessary to establish that a random variable has any particular parametric distribution. One advantage of an empirical distribution is that nothing is assumed beyond the observed values in the sample; however, this is also a disadvantage because the sample might not cover the entire range of possible values.

Example 5.35: (Discrete)

Customers at a local restaurant arrive at lunchtime in groups of from one to eight persons. The number of persons per party in the last 300 groups has been observed; the results are summarized in Table 5.3. The relative frequencies appear in Table 5.3 and again in Figure 5.27, which provides a histogram of the data that were gathered. Figure 5.28 provides a cdf of the data. The cdf in Figure 5.28 is called the empirical distribution of the given data.

Example 5.36: (Continuous)

The time required to repair a conveyor system that has suffered a failure has been collected for the last 100 instances; the results are shown in Table 5.4. There were 21 instances in which the repair took between 0 and 0.5 hour, and so on. The empirical cdf is shown in Figure 5.29. A piecewise linear curve is formed by the connection of the points of the form $[x, F(x)]$. The points are connected by a straight line. The first connected pair is (0, 0) and (0.5, 0.21); then the points (0.5, 0.21) and (1.0, 0.33) are connected; and so on. More detail on this method is provided in Chapter 8.

Table 5.3 Arrivals per Party Distribution

Arrivals per Party	*Frequency*	*Relative Frequency*	*Cumulative Relative Frequency*
1	30	0.10	0.10
2	110	0.37	0.47
3	45	0.15	0.62
4	71	0.24	0.86
5	12	0.04	0.90
6	13	0.04	0.94
7	7	0.02	0.96
8	12	0.04	1.00

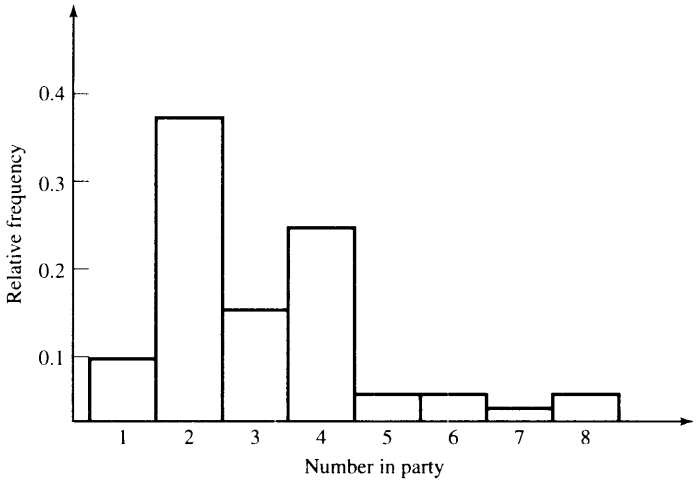


Figure 5.27 Histogram of party size.

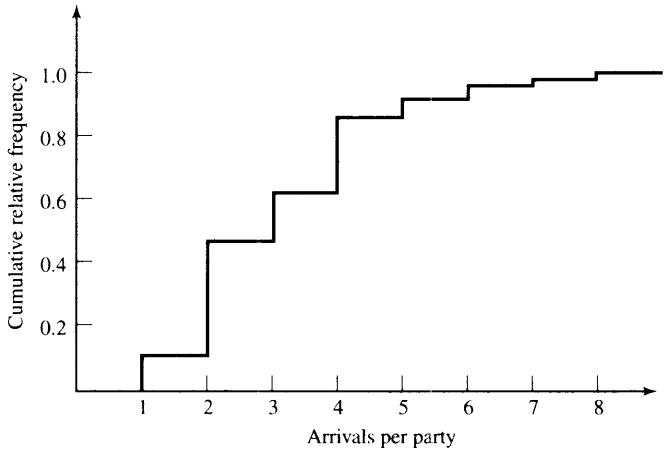


Figure 5.28 Empirical cdf of party size.

Table 5.4 Repair Times for Conveyor

Interval (Hours)	*Relative Frequency*	*Cumulative Frequency*	*Frequency*
$0 < x \leq 0.5$	21	0.21	0.21
$0.5 < x \leq 1.0$	12	0.12	0.33
$1.0 < x \leq 1.5$	29	0.29	0.62
$1.5 < x \leq 2.0$	19	0.19	0.81
$2.0 < x \leq 2.5$	8	0.08	0.89
$2.5 < x \leq 3.0$	11	0.11	1.00

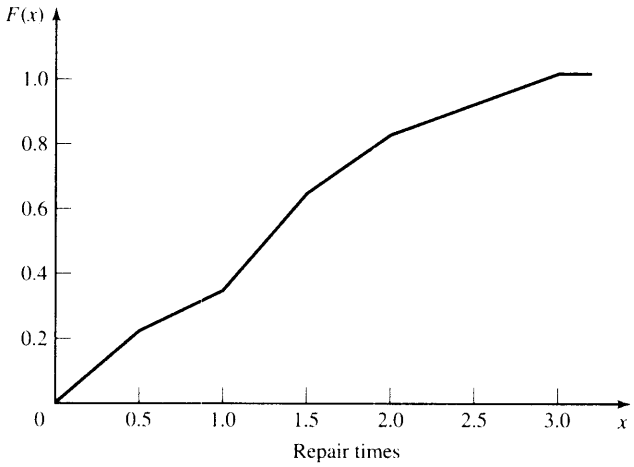


Figure 5.29 Empirical cdf for repair times.

5.7 SUMMARY

In many instances, the world the simulation analyst sees is probabilistic rather than deterministic. The purposes of this chapter were to review several important probability distributions, to familiarize the reader with the notation used in the remainder of the text, and to show applications of the probability distributions in a simulation context.

A major task in simulation is the collection and analysis of input data. One of the first steps in this task is hypothesizing a distributional form for the input data. This is accomplished by comparing the shape of the probability density function or mass function to a histogram of the data and by an understanding that certain physical processes give rise to specific distributions. (Computer software is available to assist in this effort, as will be discussed in Chapter 9.) This chapter was intended to reinforce the properties of various distributions and to give insight into how these distributions arise in practice. In addition, probabilistic models of input data are used in generating random events in a simulation.

Several features that should have made a strong impression on the reader include the differences between discrete, continuous, and empirical distributions; the Poisson process and its properties; and the versatility of the gamma and the Weibull distributions.

REFERENCES

BANKS, J., AND R. G. HEIKES [1984], *Handbook of Tables and Graphs for the Industrial Engineer and Manager*, Reston Publishing, Reston, VA.

DEVORE, J. L. [1999], *Probability and Statistics for Engineers and the Sciences*, 5th ed., Brooks/Cole, Pacific Grove, CA.

FELLER, W. [1968], *An Introduction to Probability Theory and Its Applications*, Vol. I, 3d ed., Wiley, New York.

GORDON, G. [1975], *The Application of GPSS V to Discrete System Simulation*, Prentice-Hall, Englewood Cliffs, NJ.

HADLEY, G., AND T. M. WHITIN [1963], *Analysis of Inventory Systems*, Prentice-Hall, Englewood Cliffs, NJ.

HINES, W. W., AND D. C. MONTGOMERY [1990], *Probability and Statistics in Engineering and Management Science*, 3d ed., Wiley, New York.

LAW, A. M., AND W. D. KELTON [2000], *Simulation Modeling & Analysis*, 3d ed., McGraw-Hill, New York.

PAPOULIS, A. [1990], *Probability and Statistics*, Prentice Hall, Englewood Cliffs, NJ.

PARZEN, E. [1999], *Stochastic Process*, Classics in Applied Mathematics, 24, Society for Industrial & Applied Mathematics, Philadelphia, PA.

PEGDEN, C. D., R. E. SHANNON, AND R. P. SADOWSKI [1995], *Introduction to Simulation Using SIMAN*, 2d ed., McGraw–Hill, New York.

ROSS, S. M. [2002], *Introduction to Probability Models*, 8th ed., Academic Press, New York.

WALPOLE, R. E., AND R. H. MYERS [2002], *Probability and Statistics for Engineers and Scientists*, 7th ed., Prentice Hall, Upper Saddle River, NJ.

EXERCISES

1. Of the orders a job shop receives, 25% are welding jobs and 75% are machining jobs. What is the probability that

   (a) half of the next five jobs will be machining jobs?
   (b) the next four jobs will be welding jobs?

2. Three different items are moving together in a conveyor. These items are inspected visually and defective items are removed. The previous production data are given as

	Item A	*Item B*	*Item C*
Accepted	25	280	190
Rejected	975	720	810

   What is the probability that

   (a) one item is removed at a time?
   (b) two items are removed at a time?
   (c) three items are removed simultaneously?

3. A recent survey indicated that 82% of single women aged 25 years old will be married in their lifetime. Using the binomial distribution, find the probability that two or three women in a sample of twenty will never be married.

4. The Hawks are currently winning 0.55 of their games. There are 5 games in the next two weeks. What is the probability that they will win more games than they lose?

5. Joe Coledge is the third-string quarterback for the University of Lower Alatoona. The probability that Joe gets into any game is 0.40.

   (a) What is the probability that the first game Joe enters is the fourth game of the season?
   (b) What is the probability that Joe plays in no more than two of the first five games?

6. For the random variables X_1 and X_2, which are exponentially distributed with parameter $\lambda = 1$, compute $P(X_1 + X_2 > 2)$.

7. Show that the geometric distribution is memoryless.

8. Hurricane hitting the eastern coast of India follows Poisson with a mean of 0.5 per year. Determine

   (a) the probability of more than three hurricanes hitting the Indian eastern coast in a year.
   (b) the probability of not hitting the Indian eastern coast in a year.

9. Students' arrival at a university library follows Poisson with a mean of 20 per hour. Determine

(a) the probability that there are 50 arrivals in the next 1 hour.
(b) the probability that no student arrives in the next 1 hour.
(c) the probability that there are 75 arrivals in the next 2 hours.

10. Records indicate that 1.8% of the entering students at a large state university drop out of school by midterm. What is the probability that three or fewer students will drop out of a random group of 200 entering students?

11. Lane Braintwain is quite a popular student. Lane receives, on the average, four phone calls a night (Poisson distributed). What is the probability that, tomorrow night, the number of calls received will exceed the average by more than one standard deviation?

12. A car service station receives cars at the rate of 5 every hour in accordance with Poisson. What is the probability that a car will arrive 2 hours after its predecessor?

13. A random variable X that has pmf given by $p(x) = 1/(n+1)$ over the range $R_X = \{0, 1, 2, \ldots, n\}$ is said to have a discrete uniform distribution.

(a) Find the mean and variance of this distribution. *Hint:*

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \text{ and } \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

(b) If $R_X = \{a, a+1, a+2, \ldots, b\}$, compute the mean and variance of X.

14. The lifetime, in years, of a satellite placed in orbit is given by the following pdf:

$$f(x) = \begin{cases} 0.4e^{-0.4x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

(a) What is the probability that this satellite is still "alive" after 5 years?
(b) What is the probability that the satellite dies between 3 and 6 years from the time it is placed in orbit?

15. The cars arriving at a gas station is Poisson distributed with a mean of 10 per minute. Determine the number of pumps to be installed if the firm wants to have 50% of arriving cars as zero entries (i.e., cars serviced without waiting).

16. (The Poisson distribution can be used to approximate the binomial distribution when n is large and p is small—say, p less than 0.1. In utilizing the Poisson approximation, let $\lambda = np$.) In the production of ball bearings, bubbles or depressions occur, rendering the ball bearing unfit for sale. It has been noted that, on the average, one in every 800 of the ball bearings has one or more of these defects. What is the probability that a random sample of 4000 will yield fewer than three ball bearings with bubbles or depressions?

17. For an exponentially distributed random variable X, find the value of λ that satisfies the following relationship:

$$P(X \leq 3) = 0.9P(X \leq 4)$$

18. The time between calls to a fire service station in Chennai follows exponential with a mean of 20 hours. What is the probability that there will be no calls during the next 24 hours?

19. The time to failure of a chip follows exponential with a mean of 5000 hours.

(a) The chip is in operation for the past 1000 hours. What is the probability that the chip will be in operation for another 6000 hours?
(b) After 7000 hours of operation, what is the probability that the chip will not fail for another 2000 hours?

20. The headlight bulb of a car owned by a professor has an exponential time to failure with a mean of 100 weeks. The professor has fitted a new bulb 50 weeks ago. What is the probability that the bulb will not fuse within the next 60 weeks?

21. The service time at the college cafeteria follows exponential with a mean of 2 minutes.

(a) What is the probability that two customers in front of an arriving customer will each take less than 90 seconds to complete their transactions?
(b) What is the probability that two customers in front will finish their transactions so that an arriving customer can reach the service window within 4 minutes?

22. Determine the variance, $V(X)$, of the triangular distribution.

23. The daily demand for rice at a departmental store in thousands of kilogram is found to follow gamma distribution with shape parameter 3 and scale parameter ½. Determine the probability of demand exceeding 5000 kg on a given day.

24. When Admiral Byrd went to the North Pole, he wore battery-powered thermal underwear. The batteries failed instantaneously rather than gradually. The batteries had a life that was exponentially distributed, with a mean of 12 days. The trip took 30 days. Admiral Byrd packed three batteries. What is the probability that three batteries would be a number sufficient to keep the Admiral warm?

25. In an organization's service-complaints mail box, interarrival time of mails are exponentially distributed with a mean of 10 minutes. What is the probability that five mails will arrive in 20 minutes duration?

26. The rail shuttle cars at Atlanta airport have a dual electrical braking system. A rail car switches to the standby system automatically if the first system fails. If both systems fail, there will be a crash! Assume that the life of a single electrical braking system is exponentially distributed, with a mean of 4,000 operating hours. If the systems are inspected every 5,000 operating hours, what is the probability that a rail car will not crash before that time?

27. Suppose that cars arriving at a toll booth follow a Poisson process with a mean interarrival time of 15 seconds. What is the probability that up to one minute will elapse until three cars have arrived?

28. Suppose that an average of 30 customers per hour arrive at the Sticky Donut Shop in accordance with a Poisson process. What is the probability that more than 5 minutes will elapse before both of the next two customers walk through the door?

29. Professor Dipsy Doodle gives six problems on each exam. Each problem requires an average of 30 minutes grading time for the entire class of 15 students. The grading time for each problem is exponentially distributed, and the problems are independent of each other.

(a) What is the probability that the Professor will finish the grading in $2\frac{1}{2}$ hours or less?
(b) What is the most likely grading time?
(c) What is the expected grading time?

30. An aircraft has dual hydraulic systems. The aircraft switches to the standby system automatically if the first system fails. If both systems have failed, the plane will crash. Assume that the life of a hydraulic system is exponentially distributed, with a mean of 2000 air hours.

(a) If the hydraulic systems are inspected every 2500 hours, what is the probability that an aircraft will crash before that time?
(b) What danger would there be in moving the inspection point to 3000 hours?

31. Show that the beta distribution becomes the uniform distribution over the unit interval when $\beta_1 = \beta_2 = 1$.

32. Lead time of a product in weeks is gamma-distributed with shape parameter 2 and scale parameter 1. What is the probability that the lead time exceeds 3 weeks?

33. Lifetime of an inexpensive video card for a PC, in months, denoted by the random variable X, is gamma-distributed with $\beta = 4$ and $\theta = 1/16$. What is the probability that the card will last for at least 2 years?

34. In a statewide competitive examination for engineering admission, the register number allotted to the candidates is of the form CCNNNN, where C is a character like A, B, and C, etc., and N is a number from 0 to 9. Assume that you are scanning through the rank list (based on marks secured in the competitive examination), what is the probability that

(a) the next five entries in the list will have numbers 7000 or higher?
(b) the next three entries will have numbers greater than 3000?

35. Let X be a random variable that is normally distributed, with mean 10 and variance 4. Find the values a and b such that $P(a < X < b) = 0.90$ and $|\mu - a| = |\mu - b|$.

36. Given the following distributions,

Normal (10, 4)
Triangular (4, 10, 16)
Uniform (4, 16)
find the probability that $6 < X < 8$ for each of the distributions.

37. Demand for an item follows normal distribution with a mean of 50 units and a standard deviation of 7 units. Determine the probabilities of demand exceeding 45, 55, and 65 units.

38. The annual rainfall in Chennai is normally distributed with mean 129 cm and standard deviation 32 cm.

(a) What is the probability of getting excess rain (i.e., 140 cm and above) in a given year?
(b) What is the probability of deficient rain (i.e., 80 cm and below) in a given year?

39. Three shafts are made and assembled into a linkage. The length of each shaft, in centimeters, is distributed as follows:

Shaft 1: $N(60, 0.09)$
Shaft 2: $N(40, 0.05)$
Shaft 3: $N(50, 0.11)$

(a) What is the distribution of the length of the linkage?
(b) What is the probability that the linkage will be longer than 150.2 centimeters?

(c) The tolerance limits for the assembly are (149.83, 150.21). What proportion of assemblies are within the tolerance limits? (*Hint:* If $\{X_i\}$ are n independent normal random variables, and if X_i has mean μ_i and variance σ_i^2, then the sum

$$Y = X_1 + X_2 + \cdots + X_n$$

is normal with mean $\sum_{i=1}^{n} \mu_i$ and variance $\sum_{i=1}^{n} \sigma_i^2$.)

40. The circumferences of battery posts in a nickel–cadmium battery are Weibull-distributed with $\nu = 3.25$ centimeters, $\beta = 1/3$, and $\alpha = 0.005$ centimeters.

(a) Find the probability that a battery post chosen at random will have a circumference larger than 3.40 centimeters.

(b) If battery posts are larger than 3.50 centimeters, they will not go through the hole provided; if they are smaller than 3.30 centimeters, the clamp will not tighten sufficiently. What proportion of posts will have to be scrapped for one of these reasons?

41. The time to failure of a nickel– cadmium battery is Weibull distributed with parameters $\nu = 0$, $\beta = 1/4$, and $\alpha = 1/2$ years.

(a) Find the fraction of batteries that are expected to fail prior to 1.5 years.

(b) What fraction of batteries are expected to last longer than the mean life?

(c) What fraction of batteries are expected to fail between 1.5 and 2.5 years?

42. The time required to assemble a component follows triangular distribution with $a = 10$ seconds and $c = 25$ seconds. The median is 15 seconds. Compute the modal value of assembly time.

43. The time to failure (in months) of a computer follows Weibull distribution with location parameter = 0, scale parameter = 2, and shape parameter = 0.35.

(a) What is the mean time to failure?

(b) What is the probability that the computer will fail by 3 months?

44. The consumption of raw material for a fabrication firm follows triangular distribution with minimum of 200 units, maximum of 275 units, and mean of 220 units. What is the median value of raw material consumption?

45. A postal letter carrier has a route consisting of five segments with the time in minutes to complete each segment being normally distributed, with means and variances as shown:

Tennyson Place	$N(38, 16)$
Windsor Parkway	$N(99, 29)$
Knob Hill Appartments	$N(85, 25)$
Evergreen Drive	$N(73, 20)$
Chastain Shopping Center	$N(52, 12)$

In addition to the times just mentioned, the letter carrier must organize the mail at the central office, which activity requires a time that is distributed by $N(90, 25)$. The drive to the starting point of the route requires a time that is distributed $N(10, 4)$. The return from the route requires a time that is distributed $N(15, 4)$. The letter carrier then performs administrative tasks with a time that is distributed $N(30, 9)$.

(a) What is the expected length of the letter carrier's work day?

(b) Overtime occurs after eight hours of work on a given day. What is the probability that the letter carrier works overtime on any given day?

(c) What is the probability that the letter carrier works overtime on two or more days in a six-day week?
(d) What is the probability that the route will be completed within ±24 minutes of eight hours on any given day? (*Hint:* See Exercise 39.)

46. The light used in the operation theater of a hospital has two bulbs. One bulb is sufficient to get the necessary lighting. The bulbs are connected in such a way that when one fails, automatically the other gets switched on. The life of each bulb is exponentially distributed with a mean of 5000 hours and the lives of the bulbs are independent of one another. What is the probability that the combined life of the light is greater than 7000 hours?

47. High temperature in Biloxi, Mississippi on July 21, denoted by the random variable X, has the following probability density function, where X is in degrees F.

$$f(x) = \begin{cases} \dfrac{2(x-85)}{119}, & 85 \le x \le 92 \\ \dfrac{2(102-x)}{170} & 92 < x \le 102 \\ 0, & \text{otherwise} \end{cases}$$

(a) What is the variance of the temperature, $V(X)$? (If you worked Exercise 22, this is quite easy.)
(b) What is the median temperature?
(c) What is the modal temperature?

48. The time to failure of Eastinghome light bulbs is Weibull distributed with $\nu = 1.8 \times 10^3$ hours, $\beta = 1/2$, and $\alpha = 1/3 \times 10^3$ hours.

(a) What fraction of bulbs are expected to last longer than the mean lifetime?
(b) What is the median lifetime of a light bulb?

49. Let time $t = 0$ correspond to 6 A.M., and suppose that the arrival rate (in arrivals per hour) of customers to a breakfast restaurant that is open from 6 to 9 A.M. is

$$\lambda(t) = \begin{cases} 30, & 0 \le t < 1 \\ 45, & 1 \le t < 2 \\ 20, & 2 \le t \le 4 \end{cases}$$

Assuming a NSPP model is appropriate, do the following: (a) Derive $\Lambda(t)$. (b) Compute the expected number of arrivals between 6:30 and 8:30 A.M. (c) Compute the probability that there are fewer than 60 arrivals between 6:30 and 8:30 A.M.

6

Queueing Models

Simulation is often used in the analysis of queueing models. In a simple but typical queueing model, shown in Figure 6.1, customers arrive from time to time and join a *queue* (waiting line), are eventually served, and finally leave the system. The term "customer" refers to any type of entity that can be viewed as requesting "service" from a system. Therefore, many service facilities, production systems, repair and maintenance facilities, communications and computer systems, and transport and material-handling systems can be viewed as queueing systems.

Queueing models, whether solved mathematically or analyzed through simulation, provide the analyst with a powerful tool for designing and evaluating the performance of queueing systems. Typical measures of system performance include server utilization (percentage of time a server is busy), length of waiting lines, and delays of customers. Quite often, when designing or attempting to improve a queueing system, the analyst (or decision maker) is involved in tradeoffs between server utilization and customer satisfaction in terms of line lengths and delays. Queueing theory and simulation analysis are used to predict these measures of system performance as a function of the input parameters. The input parameters include the arrival rate of customers, the service demands of customers, the rate at which a server works, and the number and

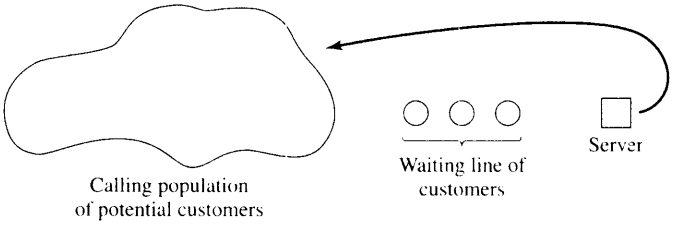


Figure 6.1 Simple queueing model.

arrangement of servers. To a certain degree, some of the input parameters are under management's direct control. Consequently, the performance measures could be under their indirect control, provided that the relationship between the performance measures and the input parameters is adequately understood for the given system.

For relatively simple systems, these performance measures can be computed mathematically—at great savings in time and expense as compared with the use of a simulation model—but, for realistic models of complex systems, simulation is usually required. Nevertheless, analytically tractable models, although usually requiring many simplifying assumptions, are valuable for rough-cut estimates of system performance. These rough-cut estimates may then be refined by use of a detailed and more realistic simulation model. Simple models are also useful for developing an understanding of the dynamic behavior of queueing systems and the relationships between various performance measures. This chapter will not develop the mathematical theory of queues but instead will discuss some of the well-known models. For an elementary treatment of queueing theory, the reader is referred to the survey chapters in Hillier and Lieberman [2005], Wagner [1975] or Winston [1997]. More extensive treatments with a view toward applications are given by Cooper [1990], Gross and Harris [1997], Hall [1991] and Nelson [1995]. The latter two texts especially emphasize engineering and management applications.

This chapter discusses the general characteristics of queues, the meanings and relationships of the important performance measures, estimation of the mean measures of performance from a simulation, the effect of varying the input parameters, and the mathematical solution of a small number of important and basic queueing models.

6.1 CHARACTERISTICS OF QUEUEING SYSTEMS

The key elements of a queueing system are the customers and servers. The term "customer" can refer to people, machines, trucks, mechanics, patients, pallets, airplanes, e-mail, cases, orders, or dirty clothes—anything that arrives at a facility and requires service. The term "server" might refer to receptionists, repairpersons, mechanics, tool-crib clerks, medical personnel, automatic storage and retrieval machines (e.g., cranes), runways at an airport, automatic packers, order pickers, CPUs in a computer, or washing machines—any resource (person, machine, etc.) that provides the requested service. Although the terminology employed will be that of a customer arriving to a service facility, sometimes the server moves to the customer; for example, a repairperson moving to a broken machine. This in no way invalidates the models but is merely a matter of terminology. Table 6.1 lists a number of different systems together with a subsystem consisting of "arriving customers" and one or more "servers." The remainder of this section describes the elements of a queueing system in more detail.

6.1.1 The Calling Population

The population of potential customers, referred to as the *calling population*, may be assumed to be finite or infinite. For example, consider a bank of five machines that are curing tires. After an interval of time, a machine automatically opens and must be attended by a worker who removes the tire and puts an uncured tire into the machine. The machines are the "customers," who "arrive" at the instant they automatically open. The worker is the "server," who "serves" an open machine as soon as possible. The calling population is finite and consists of the five machines.

In systems with a large population of potential customers, the calling population is usually assumed to be infinite. For such systems, this assumption is usually innocuous and, furthermore, it might simplify the model. Examples of infinite populations include the potential customers of a restaurant, bank, or other similar service facility and also very large groups of machines serviced by a technician. Even though the actual

Table 6.1 Examples of Queueing Systems

System	*Customers*	*Server(s)*
Reception desk	People	Receptionist
Repair facility	Machines	Repairperson
Garage	Trucks	Mechanic
Tool crib	Mechanics	Tool-crib clerk
Hospital	Patients	Nurses
Warehouse	Pallets	Fork-lift Truck
Airport	Airplanes	Runway
Production line	Cases	Case-packer
Warehouse	Orders	Order-picker
Road network	Cars	Traffic light
Grocery	Shoppers	Checkout station
Laundry	Dirty linen	Washing machines/dryers
Job shop	Jobs	Machines/workers
Lumberyard	Trucks	Overhead crane
Sawmill	Logs	Saws
Computer	Jobs	CPU, disk, CDs
Telephone	Calls	Exchange
Ticket office	Football fans	Clerk
Mass transit	Riders	Buses, trains

population could be finite but large, it is generally safe to use infinite population models—provided that the number of customers being served or waiting for service at any given time is a small proportion of the population of potential customers.

The main difference between finite and infinite population models is how the arrival rate is defined. In an infinite population model, the arrival rate (i.e., the average number of arrivals per unit of time) is not affected by the number of customers who have left the calling population and joined the queueing system. When the arrival process is homogeneous over time (e.g., there are no "rush hours"), the arrival rate is usually assumed to be constant. On the other hand, for finite calling-population models, the arrival rate to the queueing system does depend on the number of customers being served and waiting. To take an extreme case, suppose that the calling population has one member, for example, a corporate jet. When the corporate jet is being serviced by the team of mechanics who are on duty 24 hours per day, the arrival rate is zero, because there are no other potential customers (jets) who can arrive at the service facility (team of mechanics). A more typical example is that of the five tire-curing machines serviced by a single worker. When all five are closed and curing a tire, the worker is idle and the arrival rate is at a maximum, but the instant a machine opens and requires service, the arrival rate decreases. At those times when all five are open (so four machines are waiting for service while the worker is attending the other one), the arrival rate is zero; that is, no arrival is possible until the worker finishes with a machine, in which case it returns to the calling population and becomes a potential arrival. It may seem odd that the arrival rate is at its maximum when all five machines are closed. But the arrival rate is defined as the expected number of arrivals in the next unit of time, so it becomes clear that this expectation is largest when all machines could potentially open in the next unit of time.

6.1.2 System Capacity

In many queueing systems, there is a limit to the number of customers that may be in the waiting line or system. For example, an automatic car wash might have room for only 10 cars to wait in line to enter the mechanism. It might be too dangerous (or illegal) for cars to wait in the street. An arriving customer who

finds the system full does not enter but returns immediately to the calling population. Some systems, such as concert ticket sales for students, may be considered as having unlimited capacity, since there are no limits on the number of students allowed to wait to purchase tickets. As will be seen later, when a system has limited capacity, a distinction is made between the arrival rate (i.e., the number of arrivals per time unit) and the effective arrival rate (i.e., the number who arrive and enter the system per time unit).

6.1.3 The Arrival Process

The arrival process for infinite-population models is usually characterized in terms of interarrival times of successive customers. Arrivals may occur at scheduled times or at random times. When at random times, the interarrival times are usually characterized by a probability distribution. In addition, customers may arrive one at a time or in batches. The batch may be of constant size or of random size.

The most important model for random arrivals is the Poisson arrival process. If A_n represents the interarrival time between customer $n-1$ and customer n (A_1 is the actual arrival time of the first customer), then, for a Poisson arrival process, A_n is exponentially distributed with mean $1/\lambda$ time units. The arrival rate is λ customers per time unit. The number of arrivals in a time interval of length t, say $N(t)$, has the Poisson distribution with mean λt customers. For further discussion of the relationship between the Poisson distribution and the exponential distribution, the reader is referred to Section 5.5.

The Poisson arrival process has been employed successfully as a model of the arrival of people to restaurants, drive-in banks, and other service facilities; the arrival of telephone calls to a call center; the arrival of demands, or orders for a service or product; and the arrival of failed components or machines to a repair facility.

A second important class of arrivals is scheduled arrivals, such as patients to a physician's office or scheduled airline flight arrivals to an airport. In this case, the interarrival times $\{A_n, n = 1, 2, \ldots\}$ could be either *constant* or *constant plus or minus a small random amount* to represent early or late arrivals.

A third situation occurs when at least one customer is assumed to always be present in the queue, so that the server is never idle because of a lack of customers. For example, the "customers" may represent raw material for a product, and sufficient raw material is assumed to be always available.

For finite-population models, the arrival process is characterized in a completely different fashion. Define a customer as *pending* when that customer is outside the queueing system and a member of the potential calling population. For example, a tire-curing machine is "pending" when it is closed and curing a tire, and it becomes "not pending" the instant it opens and demands service from the worker. Define a *runtime* of a given customer as the length of time from departure from the queueing system until that customer's next arrival to the queue. Let $A_1^{(i)}, A_2^{(i)}, \ldots$ be the successive runtimes of customer i, and let $S_1^{(i)}, S_2^{(i)}, \ldots$ be the corresponding successive system times; that is, $S_n^{(i)}$ is the total time spent in system by customer i during the nth visit. Figure 6.2 illustrates these concepts for machine 3 in the tire-curing example. The total arrival process is the superposition of the arrival times of all customers. Figure 6.2 shows the first and second arrival of machine 3, but these two times are not necessarily two successive arrivals to the system. For instance,

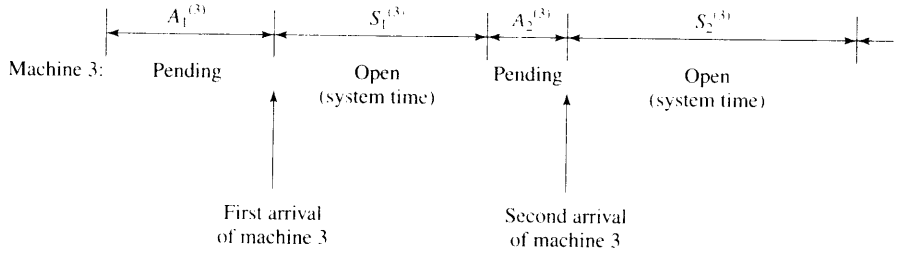


Figure 6.2 Arrival process for a finite-population model.

if it is assumed that all machines are pending at time 0, the first arrival to the system occurs at time $A_1 = \min\{A_1^{(1)}, A_1^{(2)}, A_1^{(3)}, A_1^{(4)}, A_1^{(5)}\}$. If $A_1 = A_1^{(2)}$, then machine 2 is the first arrival (i.e., the first to open) after time 0. As discussed earlier, the arrival rate is not constant but is a function of the number of pending customers.

One important application of finite-population models is the machine-repair problem. The machines are the customers, and a runtime is also called time to failure. When a machine fails, it "arrives" at the queueing system (the repair facility) and remains there until it is "served" (repaired). Times to failure for a given class of machine have been characterized by the exponential, the Weibull, and the gamma distributions. Models with an exponential runtime are sometimes analytically tractable; an example is given in Section 6.5. Successive times to failure are usually assumed to be statistically independent, but they could depend on other factors, such as the age of a machine since its last major overhaul.

6.1.4 Queue Behavior and Queue Discipline

Queue behavior refers to the actions of customers while in a queue waiting for service to begin. In some situations, there is a possibility that incoming customers will balk (leave when they see that the line is too long), renege (leave after being in the line when they see that the line is moving too slowly), or jockey (move from one line to another if they think they have chosen a slow line).

Queue discipline refers to the logical ordering of customers in a queue and determines which customer will be chosen for service when a server becomes free. Common queue disciplines include first-in–first-out (FIFO); last-in–first-out (LIFO); service in random order (SIRO); shortest processing time first (SPT); and service according to priority (PR). In a job shop, queue disciplines are sometimes based on due dates and on expected processing time for a given type of job. Notice that a FIFO queue discipline implies that services begin in the same order as arrivals, but that customers could leave the system in a different order because of different-length service times.

6.1.5 Service Times and the Service Mechanism

The service times of successive arrivals are denoted by S_1, S_2, S_3,... They may be constant or of random duration. In the latter case, $\{S_1, S_2, S_3, \ldots\}$ is usually characterized as a sequence of independent and identically distributed random variables. The exponential, Weibull, gamma, lognormal and truncated normal distributions have all been used successfully as models of service times in different situations. Sometimes services are identically distributed for all customers of a given type or class or priority, whereas customers of different types might have completely different service-time distributions. In addition, in some systems, service times depend upon the time of day or upon the length of the waiting line. For example, servers might work faster than usual when the waiting line is long, thus effectively reducing the service times.

A queueing system consists of a number of service centers and interconnecting queues. Each service center consists of some number of servers, c, working in parallel; that is, upon getting to the head of the line, a customer takes the first available server. Parallel service mechanisms are either single server ($c = 1$), multiple server ($1 < c < \infty$), or unlimited servers ($c = \infty$). A self-service facility is usually characterized as having an unlimited number of servers.

Example 6.1

Consider a discount warehouse where customers may either serve themselves or wait for one of three clerks, then finally leave after paying a single cashier. The system is represented by the flow diagram in Figure 6.3. The subsystem, consisting of queue 2 and service center 2, is shown in more detail in Figure 6.4. Other variations of service mechanisms include batch service (a server serving several customers simultaneously), and a customer's requiring several servers simultaneously. In the discount warehouse, a clerk might pick several small orders at the same time, but it may take two of the clerks to handle one heavy item.

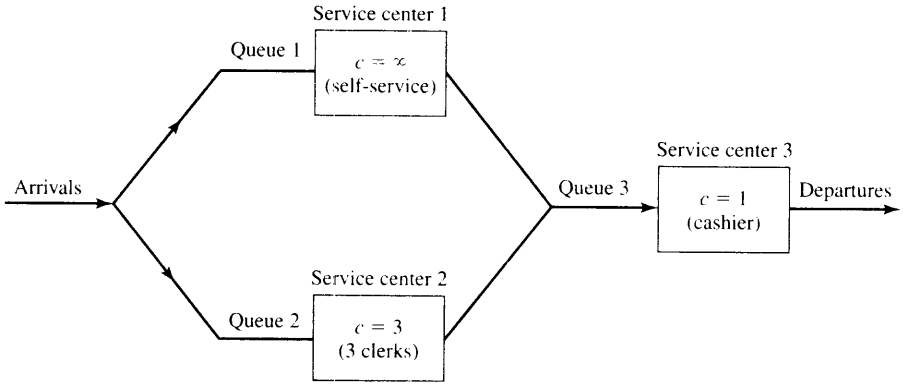


Figure 6.3 Discount warehouse with three service centers.

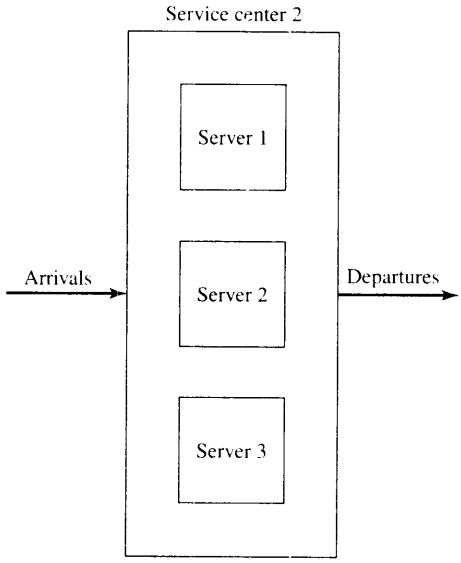


Figure 6.4 Service center 2, with $c = 3$ parallel servers.

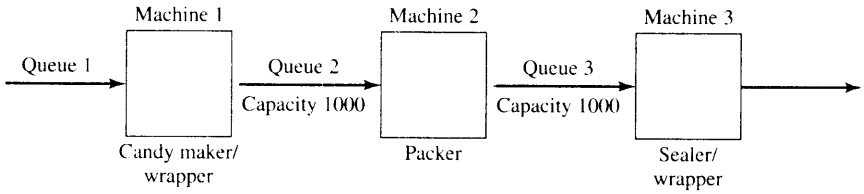


Figure 6.5 Candy-production line.

Example 6.2

A candy manufacturer has a production line that consists of three machines separated by inventory-in-process buffers. The first machine makes and wraps the individual pieces of candy, the second packs 50 pieces in a box, and the third machine seals and wraps the box. The two inventory buffers have capacities of 1000 boxes

each. As illustrated by Figure 6.5, the system is modeled as having three service centers, each center having $c = 1$ server (a machine), with queue capacity constraints between machines. It is assumed that a sufficient supply of raw material is always available at the first queue. Because of the queue capacity constraints, machine 1 shuts down whenever its inventory buffer (queue 2) fills to capacity, and machine 2 shuts down whenever its buffer empties. In brief, the system consists of three single-server queues in series with queue capacity constraints and a contisnuous arrival stream at the first queue.

6.2 QUEUEING NOTATION

Recognizing the diversity of queueing systems, Kendall [1953] proposed a notational system for parallel server systems which has been widely adopted. An abridged version of this convention is based on the format $A/B/c/N/K$. These letters represent the following system characteristics:

A represents the interarrival-time distribution.

B represents the service-time distribution.

c represents the number of parallel servers.

N represents the system capacity.

K represents the size of the calling population.

Common symbols for A and B include M (exponential or Markov), D (constant or deterministic), E_k (Erlang of order k), PH (phase-type), H (hyperexponential), G (arbitrary or general), and GI (general independent).

For example, $M/M/1/\infty/\infty$ indicates a single-server system that has unlimited queue capacity and an infinite population of potential arrivals. The interarrival times and service times are exponentially distributed. When N and K are infinite, they may be dropped from the notation. For example, $M/M/1/\infty/\infty$ is often shortened to $M/M/1$. The tire-curing system can be initially represented by $G/G/1/5/5$.

Additional notation used throughout the remainder of this chapter for parallel server systems is listed in Table 6.2. The meanings may vary slightly from system to system. All systems will be assumed to have a FIFO queue discipline.

Table 6.2 Queueing Notation for Parallel Server Systems

P_n	Steady-state probability of having n customers in system
$P_n(t)$	Probability of n customers in system at time t
λ	Arrival rate
λ_e	Effective arrival rate
μ	Service rate of one server
ρ	Server utilization
A_n	Interarrival time between customers $n-1$ and n
S_n	Service time of the nth arriving customer
W_n	Total time spent in system by the nth arriving customer
W_n^Q	Total time spent in the waiting line by customer n
$L(t)$	The number of customers in system at time t
$L_Q(t)$	The number of customers in queue at time t
L	Long-run time-average number of customers in system
L_Q	Long-run time-average number of customers in queue
w	Long-run average time spent in system per customer
w_Q	Long-run average time spent in queue per customer

6.3 LONG-RUN MEASURES OF PERFORMANCE OF QUEUEING SYSTEMS

The primary long-run measures of performance of queueing systems are the long-run time-average number of customers in the system (L) and in the queue (L_Q), the long-run average time spent in system (w) and in the queue (w_Q) per customer, and the server utilization, or proportion of time that a server is busy (ρ). The term "system" usually refers to the waiting line plus the service mechanism, but, in general, can refer to any subsystem of the queueing system; whereas the term "queue" refers to the waiting line alone. Other measures of performance of interest include the long-run proportion of customers who are delayed in queue longer than t_0 time units, the long-run proportion of customers turned away because of capacity constraints, and the long-run proportion of time the waiting line contains more than k_0 customers.

This section defines the major measures of performance for a general $G/G/c/N/K$ queueing system, discusses their relationships, and shows how they can be estimated from a simulation run. There are two types of estimators: an ordinary sample average, and a time-integrated (or time-weighted) sample average.

6.3.1 Time-Average Number in System L

Consider a queueing system over a period of time T, and let $L(t)$ denote the number of customers in the system at time t. A simulation of such a system is shown in Figure 6.6.

Let T_i denote the total time during $[0, T]$ in which the system contained exactly i customers. In Figure 6.6, it is seen that $T_0 = 3$, $T_1 = 12$, $T_2 = 4$, and $T_3 = 1$. (The line segments whose lengths total $T_1 = 12$ are labelled "T_1" in Figure 6.6, etc.) In general, $\sum_{i=0}^{\infty} T_i = T$. The time-weighted-average number in a system is defined by

$$\hat{L} = \frac{1}{T}\sum_{i=1}^{\infty} iT_i = \sum_{i=0}^{\infty} i\left(\frac{T_i}{T}\right) \tag{6.1}$$

For Figure 6.6, $\hat{L} = [0(3) + 1(12) + 2(4) + 3(1)]/20 = 23/20 = 1.15$ customers. Notice that T_i/T is the proportion of time the system contains exactly i customers. The estimator $\hat{L}$ is an example of a time-weighted average.

By considering Figure 6.6, it can be seen that the total area under the function $L(t)$ can be decomposed into rectangles of height i and length T_i. For example, the rectangle of area $3 \times T_3$ has base running from

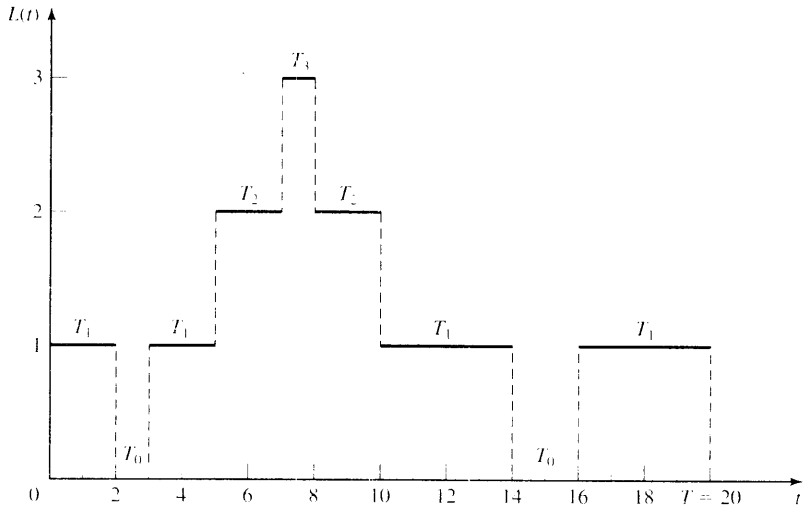


Figure 6.6 Number in system, $L(t)$, at time t.

$t = 7$ to $t = 8$ (thus $T_3 = 1$); however, most of the rectangles are broken into parts, such as the rectangle of area $2 \times T_2$ which has part of its base between $t = 5$ and $t = 7$ and the remainder from $t = 8$ to $t = 10$ (thus $T_2 = 2 + 2 = 4$). It follows that the total area is given by $\sum_{i=0}^{\infty} iT_i = \int_0^T L(t)dt$, and, therefore, that

$$\hat{L} = \frac{1}{T}\sum_{i=1}^{\infty} iT_i = \frac{1}{T}\int_0^T L(t)dt \tag{6.2}$$

The expressions in Equations (6.1) and (6.2) are always equal for any queueing system, regardless of the number of servers, the queue discipline, or any other special circumstances. Equation (6.2) justifies the terminology *time-integrated average*.

Many queueing systems exhibit a certain kind of long-run stability in terms of their average performance. For such systems, as time T gets large, the observed time-average number in the system $\hat{L}$ approaches a limiting value, say L, which is called the long-run time-average number in system—that is, with probability 1,

$$\hat{L} = \frac{1}{T}\int_0^T L(t)dt \to L \text{ as } T \to \infty \tag{6.3}$$

The estimator $\hat{L}$ is said to be strongly consistent for L. If simulation run length T is sufficiently long, the estimator $\hat{L}$ becomes arbitrarily close to L. Unfortunately, for $T < \infty$, $\hat{L}$ depends on the initial conditions at time 0.

Equations (6.2) and (6.3) can be applied to any subsystem of a queueing system as well as they can to the whole system. If $L_Q(t)$ denotes the number of customers waiting in line, and T_i^Q denotes the total time during $[0, T]$ in which exactly i customers are waiting in line, then

$$\hat{L}_Q = \frac{1}{T}\sum_{i=0}^{\infty} iT_i^Q = \frac{1}{T}\int_0^T L_Q(t)dt \to L_Q \text{ as } T \to \infty \tag{6.4}$$

where $\hat{L}_Q$ is the observed time-average number of customers waiting in line from time 0 to time T and L_Q is the long-run time-average number waiting in line.

Example 6.3

Suppose that Figure 6.6 represents a single-server queue—that is, a $G/G/1/N/K$ queueing system ($N \geq 3$, $K \geq 3$). Then the number of customers waiting in line is given by $L_Q(t)$ defined by

$$L_Q(t) = \begin{cases} 0 & \text{if } L(t) = 0 \\ L(t) - 1 & \text{if } L(t) \geq 1 \end{cases}$$

and shown in Figure 6.7. Thus, $T_0^Q = 5 + 10 = 15$, $T_1^Q = 2 + 2 = 4$, and $T_2^Q = 1$. Therefore,

$$\hat{L}_Q = \frac{0(15) + 1(4) + 2(1)}{20} = 0.3 \text{ customers}$$

6.3.2 Average Time Spent in System Per Customer w

If we simulate a queueing system for some period of time, say T, then we can record the time each customer spends in the system during $[0, T]$, say $W_1, W_2, \ldots, W_N$, where N is the number of arrivals during $[0, T]$. The average

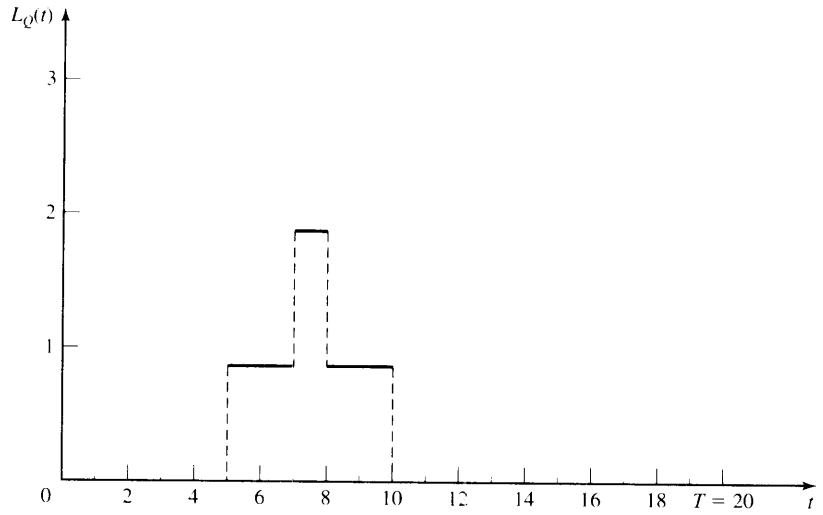


Figure 6.7 Number waiting in line, $L_Q(t)$, at time t.

time spent in system per customer, called the average system time, is given by the ordinary sample average

$$\hat{w} = \frac{1}{N}\sum_{i=1}^{N} W_i \tag{6.5}$$

For stable systems, as $N \to \infty$,

$$\hat{w} \to w \tag{6.6}$$

with probability 1, where w is called the long-run average system time.

If the system under consideration is the queue alone, Equations (6.5) and (6.6) are written as

$$\hat{w}_Q = \frac{1}{N}\sum_{i=1}^{N} W_i^Q \to w_Q \quad \text{as } N \to \infty \tag{6.7}$$

where W_i^Q is the total time customer i spends waiting in queue, $\hat{w}_Q$ is the observed average time spent in queue (called delay), and w_Q is the long-run average delay per customer. The estimators $\hat{w}$ and $\hat{w}_Q$ are influenced by initial conditions at time 0 and the run length T, analogously to $\hat{L}$.

Example 6.4

For the system history shown in Figure 6.6, $N = 5$ customers arrive, $W_1 = 2$, and $W_5 = 20 - 16 = 4$, but W_2, W_3, and W_4 cannot be computed unless more is known about the system. Assume that the system has a single server and a FIFO queue discipline. This implies that customers will depart from the system in the same order in which they arrived. Each jump upward of $L(t)$ in Figure 6.6 represents an arrival. Arrivals occur at times 0, 3, 5, 7, and 16. Similarly, departures occur at times 2, 8, 10, and 14. (A departure may or may not have occurred at time 20.) Under these assumptions, it is apparent that $W_2 = 8 - 3 = 5$, $W_3 = 10 - 5 = 5$, $W_4 = 14 - 7 = 7$, and therefore

$$\hat{w} = \frac{2+5+5+7+4}{5} = \frac{23}{5} = 4.6 \text{ time units}$$

Thus, on the average, an arbitrary customer spends 4.6 time units in the system. As for time spent in the waiting line, it can be computed that $W_1^Q = 0$, $W_2^Q = 0$, $W_3^Q = 8 - 5 = 3$, $W_4^Q = 10 - 7 = 3$, and $W_5^Q = 0$; thus,

$$\hat{w}_Q = \frac{0+0+3+3+0}{5} = 1.2 \text{ time units}$$

6.3.3 The Conservation Equation: $L = \lambda w$

For the system exhibited in Figure 6.6, there were $N = 5$ arrivals in $T = 20$ time units, and thus the observed arrival rate was $\hat{\lambda} = N/T = 1/4$ customer per time unit. Recall that $\hat{L} = 1.15$ and $\hat{w} = 4.6$; hence, it follows that

$$\hat{L} = \hat{\lambda}\hat{w} \tag{6.8}$$

This relationship between L, λ, and w is not coincidental; it holds for almost all queueing systems or subsystems regardless of the number of servers, the queue discipline, or any other special circumstances. Allowing $T \to \infty$ and $N \to \infty$, Equation (6.8) becomes

$$L = \lambda w \tag{6.9}$$

where $\hat{\lambda} \to \lambda$, and λ is the long-run average arrival rate. Equation (6.9) is called a conservation equation and is usually attributed to Little [1961]. It says that the average number of customers in the system at an arbitrary point in time is equal to the average number of arrivals per time unit, times the average time spent in the system. For Figure 6.6, there is one arrival every 4 time units (on the average) and each arrival spends 4.6 time units in the system (on the average), so at an arbitrary point in time there will be (1/4) (4.6) = 1.15 customers present (on the average).

Equation (6.8) can also be derived by reconsidering Figure 6.6 in the following manner: Figure 6.8 shows system history, $L(t)$, exactly as in Figure 6.6, with each customer's time in the system, W_i, represented by a rectangle. This representation again assumes a single-server system with a FIFO queue discipline. The

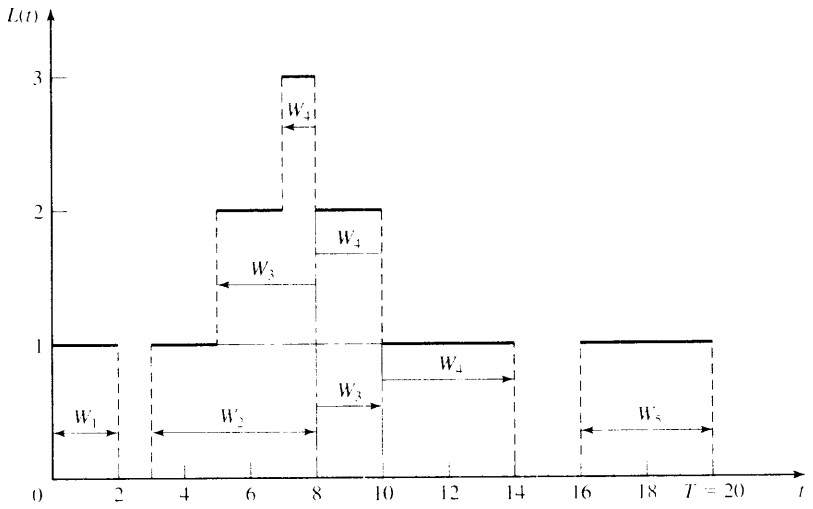


Figure 6.8 System times, W_i, for single-server FIFO system.

rectangles for the third and fourth customers are in two and three separate pieces, respectively. The ith rectangle has height 1 and length W_i for each $i = 1, 2, \ldots, N$. It follows that the total system time of all customers is given by the total area under the number-in-system function, $L(t)$; that is,

$$\sum_{i=1}^{N} W_i = \int_0^T L(t)\,dt \tag{6.10}$$

Therefore, by combining Equations (6.2) and (6.5) with $\hat{\lambda} = N/T$, it follows that

$$\hat{L} = \frac{1}{T}\int_0^T L(t)\,dt = \frac{N}{T}\frac{1}{N}\sum_{i=1}^{N} W_i = \hat{\lambda}\hat{w}$$

which is Little's equation (6.8). The intuitive and informal derivation presented here depended on the single-server FIFO assumptions, but these assumptions are not necessary. In fact, Equation (6.10), which was the key to the derivation, holds (at least approximately) in great generality, and thus so do Equations (6.8) and (6.9). Exercises 14 and 15 ask the reader to derive Equations (6.10) and (6.8) under different assumptions.

Technical note: If, as defined in Section 6.3.2, W_i is the system time for customer i during $[0, T]$, then Equation (6.10) and hence Equation (6.8) hold exactly. Some authors choose to define W_i as total system time for customer i; this change will affect the value of W_i only for those customers i who arrive before time T but do not depart until after time T (possibly customer 5 in Figure 6.8). With this change in definition, Equations (6.10) and (6.8) hold only approximately. Nevertheless, as $T \to \infty$ and $N \to \infty$, the error in Equation (6.8) decreases to zero, and, therefore, the conservation equation (6.9) for long-run measures of performance—namely, $L = \lambda w$—holds exactly.

6.3.4 Server Utilization

Server utilization is defined as the proportion of time that a server is busy. Observed server utilization, denoted by $\hat{\rho}$, is defined over a specified time interval $[0, T]$. Long-run server utilization is denoted by ρ. For systems that exhibit long-run stability,

$$\hat{\rho} \to \rho \text{ as } T \to \infty$$

Example 6.5

Per Figure 6.6 or 6.8, and assuming that the system has a single server, it can be seen that the server utilization is $\hat{\rho}$ = (total busy time)$/T = (\sum_{i=1}^{\infty} T_i)/T = (T - T_0)/T = 17/20$.

Server utilization in G/G/1/∞/∞ queues

Consider any single-server queueing system with average arrival rate λ customers per time unit, average service time $E(S) = 1/\mu$ time units, and infinite queue capacity and calling population Notice that $E(S) = 1/\mu$ implies that, when busy, the server is working at the rate μ customers per time unit, on the average; μ is called the service rate. The server alone is a subsystem that can be considered as a queueing system in itself; hence, the conservation Equation (6.9), $L = \lambda w$, can be applied to the server. For stable systems, the average arrival rate to the server, say λ_s, must be identical to the average arrival rate to the system, λ (certainly $\lambda_s \le \lambda$—customers cannot be served faster than they arrive—but, if $\lambda_s < \lambda$, then the waiting line would tend to grow in length at an average rate of

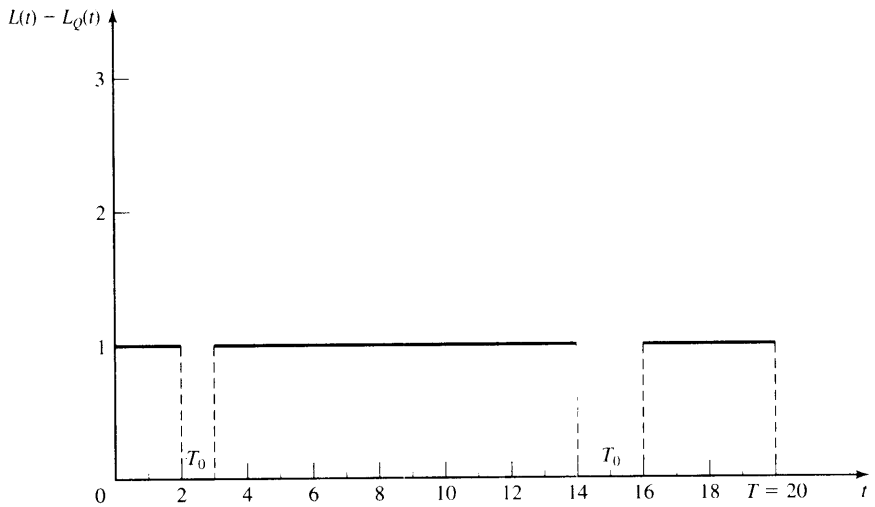


Figure 6.9 Number being served, $L(t) - L_Q(t)$, at time t.

$\lambda - \lambda_s$ customers per time unit, and so we would have an unstable system). For the server subsystem, the average system time is $w = E(S) = \mu^{-1}$. The actual number of customers in the server subsystem is either 0 or 1, as shown in Figure 6.9 for the system represented by Figure 6.6. Hence, the average number, $\hat{L}_s$, is given by

$$\hat{L}_s = \frac{1}{T}\int_0^T (L(t) - L_Q(t))\, dt = \frac{T - T_0}{T}$$

In this case, $\hat{L}_s = 17/20 = \hat{\rho}$. In general, for a single-server queue, the average number of customers being served at an arbitrary point in time is equal to server utilization. As $T \to \infty$, $\hat{L}_s = \hat{\rho} \to L_s = \rho$. Combining these results into $L = \lambda w$ for the server subsystem yields

$$\rho = \lambda E(S) = \frac{\lambda}{\mu} \tag{6.11}$$

—that is, the long-run server utilization in a single-server queue is equal to the average arrival rate divided by the average service rate. For a single-server queue to be stable, the arrival rate λ must be less than the service rate μ:

$$\lambda < \mu$$

or

$$\rho = \frac{\lambda}{\mu} < 1 \tag{6.12}$$

If the arrival rate is greater than the service rate ($\lambda > \mu$), the server will eventually get further and further behind. After a time, the server will always be busy, and the waiting line will tend to grow in length at an average rate of $(\lambda - \mu)$ customers per time unit, because departures will be occurring at rate μ per time unit. For stable single-server systems ($\lambda < \mu$ or $\rho < 1$), long-run measures of performance such as average queue

length L_Q (and also L, w, and w_Q) are well defined and have meaning. For unstable systems ($\lambda > \mu$), long-run server utilization is 1, and long-run average queue length is infinite; that is,

$$\frac{1}{T}\int_0^T L_Q(t)\,dt \to +\infty \text{ as } T \to \infty$$

Similarly, $L = w = w_Q = \infty$. Therefore these long-run measures of performance are meaningless for unstable queues. The quantity λ/μ is also called the offered load and is a measure of the workload imposed on the system.

Server utilization in G/G/c/∞/∞ queues

Consider a queueing system with c identical servers in parallel. If an arriving customer finds more than one server idle, the customer chooses a server without favoring any particular server. (For example, the choice of server might be made at random.) Arrivals occur at rate λ from an infinite calling population, and each server works at rate μ customers per time unit. From Equation (6.9), $L = \lambda w$, applied to the server subsystem alone, an argument similar to the one given for a single server leads to the result that, for systems in statistical equilibrium, the average number of busy servers, say L_s, is given by

$$L_s = \lambda E(S) = \frac{\lambda}{\mu} \tag{6.13}$$

Clearly, $0 \le L_s \le c$. The long-run average server utilization is defined by

$$\rho = \frac{L_s}{c} = \frac{\lambda}{c\mu} \tag{6.14}$$

and so $0 \le \rho \le 1$. The utilization ρ can be interpreted as the proportion of time an arbitrary server is busy in the long run.

The maximum service rate of the $G/G/c/\infty/\infty$ system is $c\mu$, which occurs when all servers are busy. For the system to be stable, the average arrival rate λ must be less than the maximum service rate $c\mu$; that is, the system is stable if and only if

$$\lambda < c\mu \tag{6.15}$$

or, equivalently, if the offered load λ/μ is less than the number of servers c. If $\lambda > c\mu$, then arrivals are occurring, on the average, faster than the system can handle them, all servers will be continuously busy, and the waiting line will grow in length at an average rate of $(\lambda - c\mu)$ customers per time unit. Such a system is unstable, and the long-run performance measures (L, L_Q, w, and w_Q) are again meaningless for such systems.

Notice that Condition (6.15) generalizes Condition (6.12), and the equation for utilization for stable systems, Equation (6.14), generalizes Equation (6.11).

Equations (6.13) and (6.14) can also be applied when some servers work more than others, for example, when customers favor one server over others, or when certain servers serve customers only if all other servers are busy. In this case, the L_s given by Equation (6.13) is still the average number of busy servers, but ρ, as given by Equation (6.14), cannot be applied to an individual server. Instead, ρ must be interpreted as the average utilization of all servers.

Example 6.6

Customers arrive at random to a license bureau at a rate of $\lambda = 50$ customers per hour. Currently, there are 20-clerks, each serving $\mu = 5$ customers per hour on the average. Therefore the long-run, or steady-state, average utilization of a server, given by Equation (6.14), is

$$\rho = \frac{\lambda}{c\mu} = \frac{50}{20(5)} = 0.5$$

and the average number of busy servers is

$$L_s = \frac{\lambda}{\mu} = \frac{50}{5} = 10$$

Thus, in the long run, a typical clerk is busy serving customers only 50% of the time. The office manager asks whether the number of servers can be decreased. By Equation (6.15), it follows that, for the system to be stable, it is necessary for the number of servers to satisfy

$$c > \frac{\lambda}{\mu}$$

or $c > 50/5 = 10$. Thus, possibilities for the manager to consider include $c = 11$, or $c = 12$, or $c = 13$, ... Notice that $c \geq 11$ guarantees long-run stability only in the sense that all servers, when busy, can handle the incoming work load (i.e., $c\mu > \lambda$) on average. The office manager could well desire to have more than the minimum number of servers ($c = 11$) because of other factors, such as customer delays and length of the waiting line. A stable queue can still have very long lines on average.

Server utilization and system performance

As will be illustrated here and in later sections, system performance can vary widely for a given value of utilization, ρ. Consider a $G/G/1/\infty/\infty$ queue: that is, a single-server queue with arrival rate λ, service rate μ, and utilization $\rho = \lambda/\mu < 1$.

At one extreme, consider the $D/D/1$ queue, which has deterministic arrival and service times. Then all interarrival times $\{A_1, A_2, \ldots\}$ are equal to $E(A) = 1/\lambda$, and all service times $\{S_1, S_2, \ldots\}$ are equal to $E(S) = 1/\mu$. Assuming that a customer arrives to an empty system at time 0, the system evolves in a completely deterministic and predictable fashion, as shown in Figure 6.10. Observe that $L = \rho = \lambda/\mu$, $w = E(S) = \mu^{-1}$, and $L_Q = w_Q = 0$. By varying λ and μ, server utilization can assume any value between 0 and 1, yet there is never any line whatsoever. What, then, causes lines to build, if not a high server utilization? In general, it is the variability of interarrival and service times that causes lines to fluctuate in length.

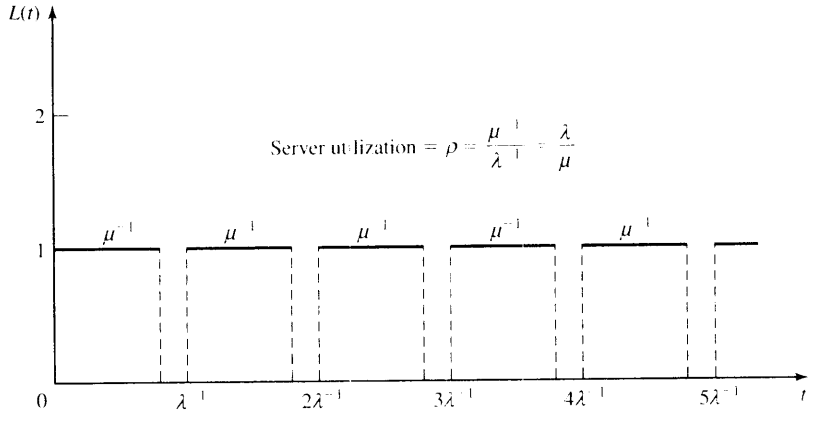


Figure 6.10 Deterministic queue ($D/D/1$).

Example 6.7

Consider a physician who schedules patients every 10 minutes and who spends S_i minutes with the ith patient, where

$$S_i = \begin{cases} 9 \text{ minutes with probability } 0.9 \\ 12 \text{ minutes with probability } 0.1 \end{cases}$$

Thus, arrivals are deterministic ($A_1 = A_2 = \cdots = \lambda^{-1} = 10$) but services are stochastic (or probabilistic), with mean and variance given by

$$E(S_i) = 9(0.9) + 12(0.1) = 9.3 \text{ minutes}$$

and

$$\begin{aligned} V(S_i) &= E(S_i^2) - [E(S_i)]^2 \\ &= 9^2(0.9) + 12^2(0.1) - (9.3)^2 \\ &= 0.81 \text{ minutes}^2 \end{aligned}$$

Here, $\rho = \lambda/\mu = E(S)/E(A) = 9.3/10 = 0.93 < 1$, the system is stable, and the physician will be busy 93% of the time in the long run. In the short run, lines will not build up as long as patients require only 9 minutes of service, but, because of the variability in the service times, 10% of the patients will require 12 minutes, which in turn will cause a temporary line to form.

Suppose the system is simulated with service times, $S_1 = 9$, $S_2 = 12$, $S_3 = 9$, $S_4 = 9$, $S_5 = 9$, ... Assuming that at time 0 a patient arrived to find the doctor idle and subsequent patients arrived precisely at times 10, 20, 30, ..., the system evolves as in Figure 6.11. The delays in queue are $W_1^Q = W_2^Q = 0$, $W_3^Q = 22 - 20 = 2$, $W_4^Q = 31 - 30 = 1$, $W_5^Q = 0$. The occurrence of a relatively long service time (here $S_2 = 12$) caused a waiting line to form temporarily. In general, because of the variability of the interarrival and service distributions, relatively small interarrival times and relatively large service times occasionally do occur, and these in turn cause lines to lengthen. Conversely, the occurrence of a large interarrival time or a small service time will tend to shorten an existing waiting line. The relationship between utilization, service and interarrival variability, and system performance will be explored in more detail in Section 6.4.

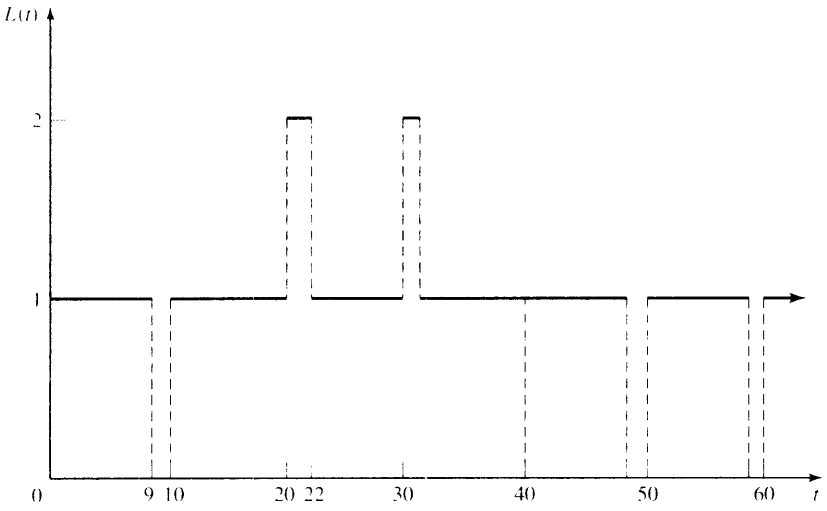


Figure 6.11 Number of patients in the doctor's office at time t.

6.3.5 Costs in Queueing Problems

In many queueing situations, costs can be associated with various aspects of the waiting line or servers. Suppose that the system incurs a cost for each customer in the queue, say at a rate of \$10 per hour per customer. If customer j spends W_j^Q hours in the queue, then $\sum_{j=1}^{N}(\$10 \cdot W_j^Q)$ is the total cost of the N customers who arrive during the simulation. Thus, the average cost per customer is

$$\sum_{j=1}^{N} \frac{\$10 \cdot W_j^Q}{N} = \$10 \cdot \hat{w}_Q$$

by Equation (6.7). If $\hat{\lambda}$ customers per hour arrive (on the average), the average cost per hour is

$$\left(\hat{\lambda}\frac{\text{customers}}{\text{hour}}\right)\left(\frac{\$10 \cdot \hat{w}_Q}{\text{customer}}\right) = \$10 \cdot \hat{\lambda}\hat{w}_Q = \$10 \cdot \hat{L}_Q/\text{hour}$$

the last equality following by Little's equation (6.8). An alternative way to derive the average cost per hour is to consider Equation (6.2). If T_i^Q is the total time over the interval $[0, T]$ that the system contains exactly i customers, then $\$10\, iT_i^Q$ is the cost incurred by the system during the time exactly i customers are present. Thus, the total cost is $\sum_{i=1}^{\infty}(\$10 \cdot iT_i^Q)$, and the average cost per hour is

$$\sum_{i=1}^{\infty} \frac{\$10 \cdot iT_i^Q}{T} = \$10 \cdot L_Q/\text{hour}$$

by Equation (6.2). In these cost expressions, $\hat{L}_Q$ may be replaced by L_Q (if the long-run number in queue is known), or by L or $\hat{L}$ (if costs are incurred while the customer is being served in addition to being delayed).

The server may also impose costs on the system. If a group of c parallel servers ($1 \le c < \infty$) have utilization ρ, and each server imposes a cost of \$5 per hour while busy, the total server cost per hour is

$$\$5 \cdot c\rho$$

because $c\rho$ is the average number of busy servers. If server cost is imposed only when the servers are idle, then the server cost per hour would be

$$\$5 \cdot c(1-\rho)$$

because $c(1-\rho) = c - c\rho$ is the average number of idle servers. In many problems, two or more of these various costs are combined into a total cost. Such problems are illustrated by Exercises 5, 12, 17, and 20. In most cases, the objective is to minimize total costs (given certain constraints) by varying those parameters that are under management's control, such as the number of servers, the arrival rate, the service rate, and the system capacity.

6.4 STEADY-STATE BEHAVIOR OF INFINITE-POPULATION MARKOVIAN MODELS

This section presents the steady-state solution of a number of queueing models that can be solved mathematically. For the infinite-population models, the arrivals are assumed to follow a Poisson process with rate λ arrivals per time unit—that is, the interarrival times are assumed to be exponentially distributed with mean $1/\lambda$. Service

times may be exponentially distributed (M) or arbitrarily (G). The queue discipline will be FIFO. Because of the exponential-distributional assumptions on the arrival process, these models are called Markovian models.

A queueing system is said to be in statistical equilibrium, or steady state, if the probability that the system is in a given state is not time dependent—that is,

$$P(L(t) = n) = P_n(t) = P_n$$

is independent of time t. Two properties—approaching statistical equilibrium from any starting state, and remaining in statistical equilibrium once it is reached—are characteristic of many stochastic models, and, in particular, of all the systems studied in the following subsections. On the other hand, if an analyst were interested in the transient behavior of a queue over a relatively short period of time and were given some specific initial conditions (such as idle and empty), the results to be presented here would be inappropriate. A transient mathematical analysis or, more likely, a simulation model would be the chosen tool of analysis.

The mathematical models whose solutions are shown in the following subsections can be used to obtain approximate results even when the assumptions of the model do not strictly hold. These results may be considered as a rough guide to the behavior of the system. A simulation may then be used for a more refined analysis. However, it should be remembered that a mathematical analysis (when it is applicable) provides the true value of the model parameter (e.g., L), whereas a simulation analysis delivers a statistical estimate (e.g., $\hat{L}$) of the parameter. On the other hand, for complex systems, a simulation model is often a more faithful representation than a mathematical model.

For the simple models studied here, the steady-state parameter L, the time-average number of customers in the system, can be computed as

$$L = \sum_{n=0}^{\infty} nP_n \tag{6.16}$$

where $\{P_n\}$ are the steady-state probabilities of finding n customers in the system (as defined in Table 6.2). As was discussed in Section 6.3 and was expressed in Equation 6.3), L can also be interpreted as a long-run measure of performance of the system. Once L is given, the other steady-state parameters can be computed readily from Little's equation (6.9) applied to the whole system and to the queue alone:

$$\begin{aligned} w &= \frac{L}{\lambda} \\ w_Q &= w - \frac{1}{\mu} \\ L_Q &= \lambda w_Q \end{aligned} \tag{6.17}$$

where λ is the arrival rate and μ is the service rate per server.

For the $G/G/c/\infty/\infty$ queues considered in this section to have a statistical equilibrium, a necessary and sufficient condition is that $\lambda/(c\mu) < 1$, where λ is the arrival rate, μ is the service rate of one server, and c is the number of parallel servers. For these unlimited capacity, infinite-calling-population models, it shall be assumed that the theoretical server utilization, $\rho = \lambda/(c\mu)$, satisfies $\rho < 1$. For models with finite system capacity or finite calling population, the quantity $\lambda/(c\mu)$ may assume any positive value.

6.4.1 Single-Server Queues with Poisson Arrivals and Unlimited Capacity: M/G/1

Suppose that service times have mean $1/\mu$ and variance σ^2 and that there is one server. If $\rho = \lambda/\mu < 1$, then the $M/G/1$ queue has a steady-state probability distribution with steady-state characteristics, as given in Table 6.3. In general, there is no simple expression for the steady-state probabilities $P_0, P_1, P_2, \ldots$. When

Table 6.3 Steady-State Parameters of the $M/G/1$ Queue

ρ	$\frac{\lambda}{\mu}$
L	$\rho + \frac{\lambda^2(1/\mu^2 + \sigma^2)}{2(1-\rho)} = \rho + \frac{\rho^2(1+\sigma^2\mu^2)}{2(1-\rho)}$
w	$\frac{1}{\mu} + \frac{\lambda(1/\mu^2 + \sigma^2)}{2(1-\rho)}$
w_Q	$\frac{\lambda(1/\mu^2 + \sigma^2)}{2(1-\rho)}$
L_Q	$\frac{\lambda^2(1/\mu^2 + \sigma^2)}{2(1-\rho)} = \frac{\rho^2(1+\sigma^2\mu^2)}{2(1-\rho)}$
P_0	$1 - \rho$

$\lambda < \mu$, the quantity $\rho = \lambda/\mu$ is the server utilization, or long-run proportion of time the server is busy. As is seen in Table 6.3, $1 - P_0 = \rho$ can also be interpreted as the steady-state probability that the system contains one or more customers. Notice also that $L - L_Q = \rho$ is the time-average number of customers being served.

Example 6.8

Widget-making machines malfunction apparently at random and then require a mechanic's attention. It is assumed that malfunctions occur according to a Poisson process, at the rate $\lambda = 1.5$ per hour. Observation over several months has found that repair times by the single mechanic take an average time of 30 minutes, with a standard deviation of 20 minutes. Thus the mean service time $1/\mu = 1/2$ hour, the service rate is $\mu = 2$ per hour and $\sigma^2 = (20)^2$ minutes^2 = 1/9 hour^2. The "customers" are the widget makers, and the appropriate model is the $M/G/1$ queue, because only the mean and variance of service times are known, not their distribution. The proportion of time the mechanic is busy is $\rho = \lambda/\mu = 1.5/2 = 0.75$, and, by Table 6.3, the steady-state time average number of broken machines is

$$L = 0.75 + \frac{(1.5)^2[(0.5)^2 + 1/9]}{2(1-0.75)}$$
$$= 0.75 + 1.625 = 2.375 \text{ machines}$$

Thus, an observer who notes the state of the repair system at arbitrary times would find an average of 2.375 broken machines (over the long run).

A closer look at the formulas in Table 6.3 reveals the source of the waiting lines and delays in an $M/G/1$ queue. For example, L_Q may be rewritten as

$$L_Q = \frac{\rho^2}{2(1-\rho)} + \frac{\lambda^2\sigma^2}{2(1-\rho)}$$

The first term involves only the ratio of the mean arrival rate, λ, to the mean service rate, μ. As shown by the second term, if λ and μ are held constant, the average length of the waiting line (L_Q) depends on the variability, σ^2, of the service times. If two systems have identical mean service times and mean interarrival times,