Discrete Simulation and Animation for Mining Engineers

John R. Sturgul

CRC Press
Taylor & Francis Group

Discrete Simulation and Animation for Mining Engineers

John R. Sturgul

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MIX
Paper from
responsible sources
FSC® C014174
FSC
www.fsc.org

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20150605

International Standard Book Number-13: 978-1-4822-5441-9 (Hardback)

Discrete Simulation and Animation for Mining Engineers

Preface

A bucket-wheel excavator is the largest equipment in the world.

Mining simulation and animation can be a fun course! Ask any student who has taken a well-presented course on adding animations to the systems just simulated. It is one thing to write a simulation program that will create 100 different coloured stars but quite another to have these stars appear on the screen at random with various colours and make them blink, move, and then disappear. This course will teach you both how to do simulations using the powerful GPSS/H® simulation language and, along with it, how to make your simulation *come alive* with an animation software application known as PROOF®.

A simulation study represents an actual system as closely as possible, and the animation allows the user to *see* his or her system in motion. PROOF animation is one of the best such animations on the market and easily follows from GPSS/H. It can, of course, be used with other simulation languages (and non-simulation languages for that matter).

This course is the result of many years of teaching mining students from widely diverse backgrounds. It is also the evolution of notes used for short courses in many countries, such as Argentina, Australia, Brazil, Bulgaria, Canada, Chile, Peru, Poland, China, Romania, Russia, South Africa, Spain, Tanzania, and the United States. More importantly, it is the result of modelling actual mines and mining operations in the above countries plus others as well, such as Papua New Guinea, Borneo, Namibia, Ghana, and even a few others.

It is necessary to have both the student versions of PROOF and GPSS/H installed. These can be downloaded free of cost from www.wolverinesoftware.com.

All of the examples presented here assume that you have installed both Student GPSS/H and Student PROOF.

There will be quite a few files from which to learn. Those with Student PROOF will be in *pairs*: *.LAY and *.ATF, where the "*" could be any legal name such as MYFIRST.LAY and MYFIRST.ATF. Those with Student GPSS/H will be of form *.GPS. For example, a program might be named MYPROG.GPS and the animation pairs MYPROG.LAY and MYPROG.ATF.

If you already have other versions of the software, such as PROOF Professional® or GPSS/H® Professional, this is fine. You will need to know only in which directory on your computer each file is located and how to run the programs and the animation.

With every chapter, there are examples that should be run as exercises. In some cases, I debated whether to include the listing of the GPSS/H program when the actual program is given in a file. I decided to include certain listings where the student can benefit from important parts of it. In many cases, the student will be instructed to rerun the program with designated lines changed in order to compare results.

Comments, suggestions, and perhaps even some mining examples that the reader has created that might be added to these exercises are solicited. Send them to:john.sturgul@adelaide.edu.au.

The form of this course is a bit different from that of other courses. The normal way of teaching this subject is to cover first the simulation language and then, once the students are writing their simulation programs, to gradually introduce animation. Having taught simulation and animation for many years, I have noticed that the students really come alive when they start doing the animations. No doubt this is because of the opportunity for each to be his or her own artist. It is easy to be creative and make fancy animations, especially using the software presented here.

This course starts with animation. The first six chapters refer only to animation. Each topic or point to be considered will have an animation (or animations) to show the students what is being done. The simulation language is slowly introduced and, as each new topic is studied, there will be an animation to illustrate this topic. Thus, the student will always be guided to view an animation to *see* what is being taught.

Special thanks must go to James O. Henriksen, president of Wolverine Software, for his assistance in this book. Jim developed GPSS/H while a graduate student at the University of Michigan and continued its many morphs through the years. Dr. Thomas Schriber of the University of Michigan was also very helpful with his many suggestions and for his kindness in allowing me to use several of the examples from his classic textbooks on the GPSS and GPSS/H languages. It was his textbooks that got me started in this field, for which I am eternally thankful.

For the student: I envy you in a way. You are about to embark on a journey that will give you one of the strongest tools that exists for modelling the real world. It is rare that one can learn only one subject and then immediately be able to feel that he or she can solve real-life problems. Once you finish this course and work through the many examples and exercises, you will have a tool that will provide you such skills.

I tell my students during the first lecture that they will learn many subjects during their study to become mining engineers. These include subjects such as mine ventilation, rock mechanics, open-pit slope design, and underground mine design. All are important and necessary in any mining curriculum. However, no one is an expert without many years of actual practice. On the other hand, once a student learns mine simulation and animation, he or she can immediately put this training to use on day 1.

John R. Sturgul
University of Adelaide

Additional material is available from the CRC Web site: https://www.crcpress.com/product/isbn/9781482254419

Introduction

Simulation Models in General

The GPSS/H (General Purpose Simulation System/H) computer programming language is a special language that is used primarily to simulate what can be classified as *discrete systems*. A discrete system is one where, at any given instant in time, a *countable* number of things can take place. For example, students working in a computer room with multiple PCs comprise such a system; other examples are a truck being loaded, a truck dumping, or a shovel starting to load a truck. At any one instant in time, the number of things taking place can be counted. Many engineering systems are ideal examples of discrete systems. However, air flowing through a mine, water flowing through a pipe, and the deformation of a chair under the load of a heavy person are not examples of discrete systems.

Considering only discrete systems may seem restrictive, but it is not as restrictive as may appear at first reading. Nearly all of the problems one encounters in the study of queuing theory can be represented by discrete systems. Textbooks on the subject of operations research have numerous examples that can be considered as discrete systems. Some systems that may not appear to be discrete easily can be approximated by discrete systems. These might include cars and trucks travelling from one point to another. When travelling, their motion is certainly continuous but, for the purpose of studying traffic flow, their motion can be considered using discrete simulation. Material on a conveyor belt also can be considered as being part of a discrete system. We will study these and other examples in this book.

Some Examples of Discrete Systems

There are many examples of discrete systems. In general, whenever there is a queuing situation or a potential one, the system may be considered as a discrete system. A few such systems are as follows:

People coming to a news stand with a single worker: If the worker is busy, people wait until it is their turn.

People entering a bank with multiple tellers: The customers may either form individual queues for each teller or wait in a single queue (known as a *quickline*).

Trucks working at a construction site where a single shovel loads each truck: The trucks travel to a dump area where they dump and then return to the shovel. This is an example of a *cyclic* queue. The elements of the system, in this case, the trucks, do not leave the system.

The *barbershop* problem, where people come from a large population and there is a single barber: If the barber is busy, the customers will wait in the chairs provided. If all of the chairs are taken, customers will go away and, perhaps, return later or possibly go to another barber.

Ships entering a harbour with multiple berths: The ships need to be towed into a berth with one or more tug boats.

Telephone calls arriving at a central switchboard where they need to be routed to the correct extension.

Television sets on a conveyor belt arriving at an inspection station: If the set fails inspection, it may be sent back for adjustment or, in the worst case, be discarded.

Trucks breaking down and having to come to the repair shop for repairs.

There are many situations in a mine that give rise to discrete simulation. The basic operation of a mine itself can be considered such a system. Reliability of the equipment and inventory problems are also examples of discrete systems. Following the flow of the ore from the mine to the mill, to the loading docks and then off to the markets is also an example.

A complete treatment of simulation theory is beyond the scope of this book. However, an understanding of how simulation models are constructed and what they tell us is not too difficult.

Consider a simplified version of a bank with customers arriving and tellers giving service. Ignore the fact that the bank may offer multiple services such as insurance and new accounts. In this simplified bank, customers arrive only to visit one of the multiple tellers. All the possible events that take place in this bank are discrete events or can be considered as such. Possible events might be customers arriving, customers joining a queue if all the tellers are busy, customers going to a teller who is free, and customers leaving the bank when finished. Perhaps some of the customers will leave the queue if the waiting time is too long and go to another store or stores and return later. In most cases, we shall be modelling systems that will involve some queuing, for example, when all the tellers are busy in the bank, all the petrol pumps in a petrol station are being used, and all the checkout counters at the grocery store are in use.

The classic problem of trucks working in a surface mine site with a single shovel is one that has quite a history. Civil and mining engineers have been trying to solve this problem using classical means for many years. In this case, there are n trucks at the site and a single shovel that can load only one truck at a time. The loaded trucks travel to a single dump area where they dump their loads and return to the shovel. Since the shovel can load only one truck at a time, an arriving truck sometimes must wait in a queue until the shovel is free. The *problem* is to determine the production of the system as the number of trucks increases. For constant load, dump, and haul times, the solution is trivial, but for stochastic times, the problem can be solved for the general case only by simulation. In fact, it is easily solved in only a few lines of computer code using the software presented here (Figure I.1).

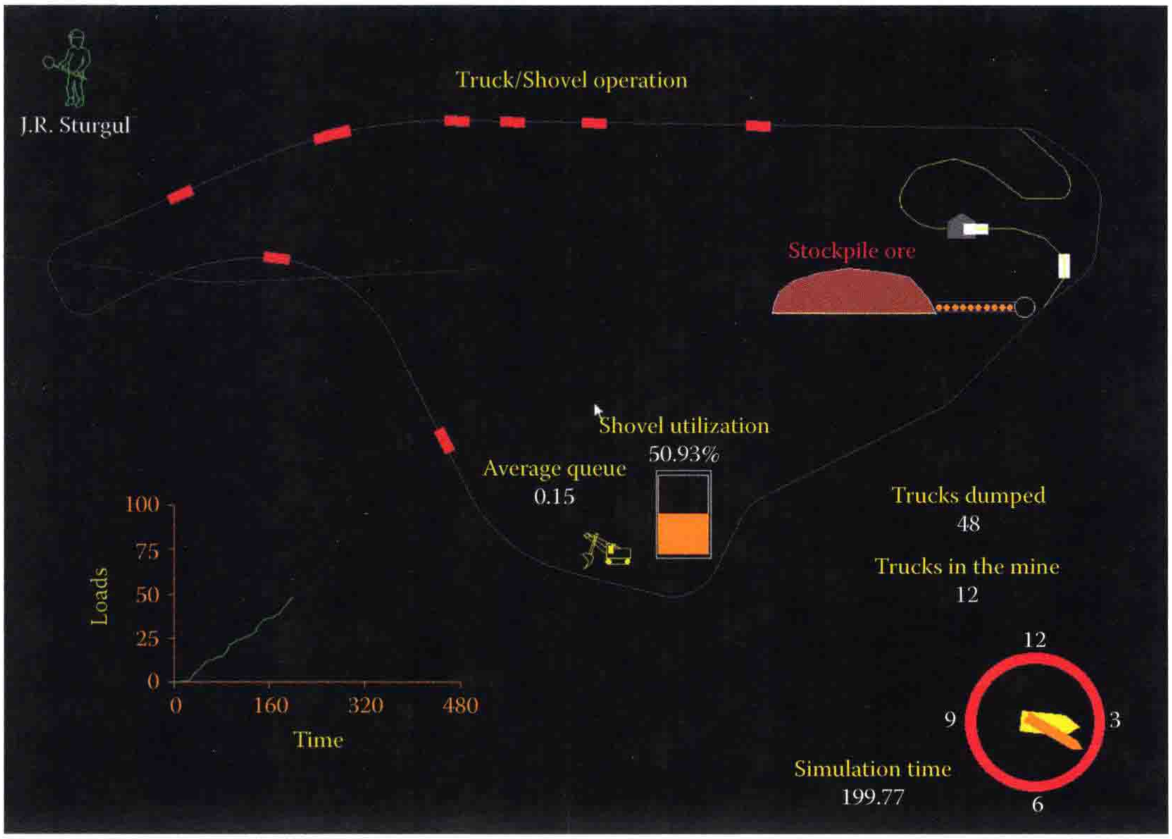


FIGURE I.1
Animation for a basic mine design problem.

The GPSS/H simulation language is excellent for simulating systems that have this type of queuing. As we shall see, it is very easy to model a great variety of very complicated systems using GPSS/H.

What Will Be Modelled?

The models we shall be studying will be primarily ones that mining engineers will encounter. However, the models might represent a bank working over a period of many months; an assembly plant that manufactures television sets; a barbershop where customers can obtain haircuts, shampoos, and manicures; or even a person doing the Saturday morning shopping. In some cases, the model may be only a small part of a large system, such as the tool crib in a large factory or the repair shop for small items.

It is important to understand that a simulation model does not in itself solve a problem. Instead, it tells the modeller how the system under study will work given certain parameters. For example, a simulation model might be constructed for a system that uses 10 trucks. The model might then be rerun using 11 trucks, then 12 trucks, and so on. It is the modeller who takes the results of the simulations and uses them to draw a conclusion. For example, the model might be a simple manufacturing system. Assuming the model is correct, the modeller might run it with four workers and determine, using the results of the simulation, that this number of workers might result in a profit of $500 per day. With 5 workers, the profit might be $526.79 per day and with 6 or more workers, the profit might be less than this figure. Hence, the conclusion to be drawn is that the system works best with 5 workers. This conclusion implicitly assumes that the model used for the simulation and all input data are correct. If the workers suddenly were given a pay raise, the model results may have to be re-evaluated to see if the same conclusion holds.

Thus, the simulation models we construct will not in themselves *solve* any problem directly but will provide information about how the system is working and, then, how it will work with certain selected parameters changed. Suppose a company has its own fleet of cars for its salesmen to use. If the cars need any service, whether it is of a routine nature or major repairs, it is done by one of two mechanics. The company is concerned that the mechanics are not able to keep up with the repairs and wonders whether it would be worth its while to hire another mechanic. Before the simulation model can be constructed, the company must define the problem to be solved in greater detail than has been given here. The following information is also needed:

1. The frequency of service and the distribution of times for the particular service.
2. What it would cost to hire a third mechanic, as well as the cost in lost sales when a car is not available.
3. When the managers bring their cars in for minor service, these cars have priority over those that are in for major service. This means that these cars are put in the front of the queue of any other cars that are waiting for service.
4. When the company owner brings his car in for service, it is given a special status and this car is immediately worked on. Thus, even if both mechanics are busy, one will put aside the car he is working on and start the repairs on the owner's car.

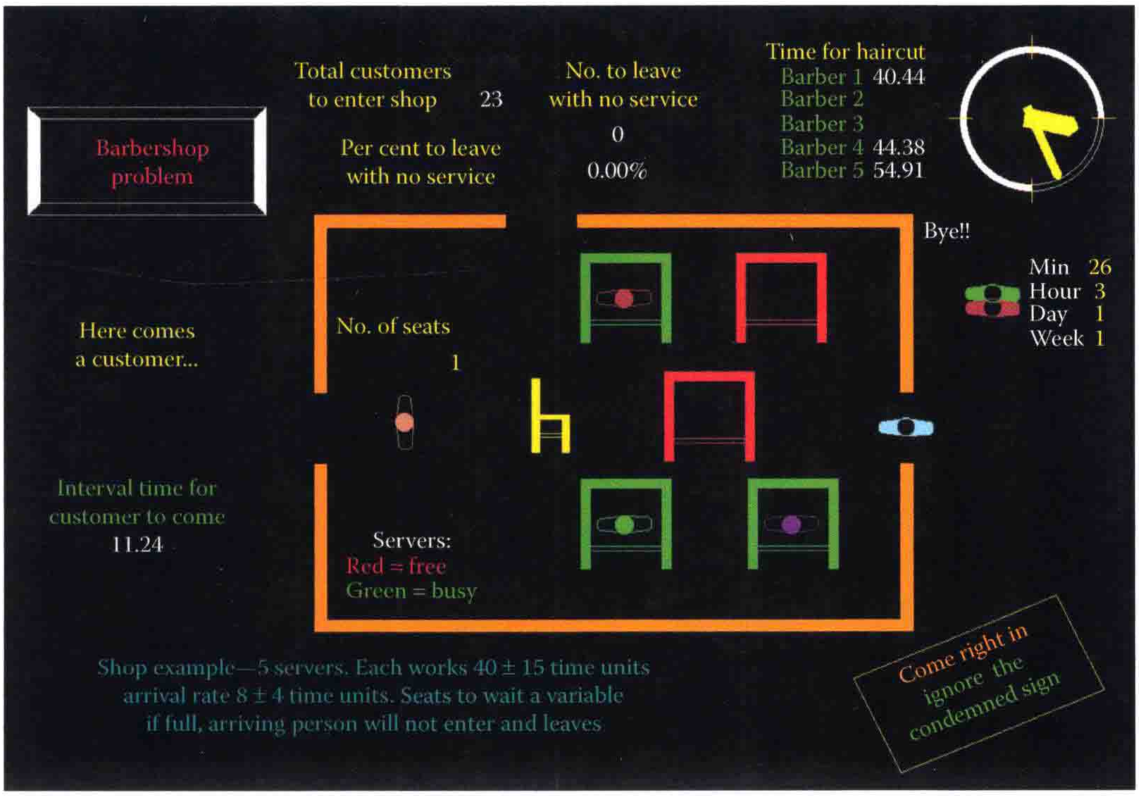
The information obtained from the simulation model might include the following:

How the system presently works. Obviously, the computer model has to accurately reflect the system as it is working before any reliability can be associated with the results from the changed model.

How the system will work with three mechanics.

Another Type of Problem to Be Studied and the Method of Solution

Consider the following system consisting of a five-person barbershop with customers arriving to have their hair cut. The shop operates 8 hours a day with no time off for coffee breaks or even for lunch. There are five chairs for the haircuts and only one chair is provided for the customers to wait if all five barbers are busy. Assuming that people do not like to stand around waiting, the system can only hold six customers. If a seventh customer arrives and finds the shop full, he will always leave. All five barbers are identical, so it can be assumed that they work at the same rate and the customers have no preference for any barber. The customers do not arrive at regular intervals, and the haircuts are not given at exact times but vary according to known statistical distributions (Figure I.2).



Shop example—5 servers. Each works 40 ± 15 time units arrival rate 8 ± 4 time units. Seats to wait a variable if full, arriving person will not enter and leaves

FIGURE I.2
Animation of barbershop problem—customers are represented by elliptical shapes.

The owner of the barbershop would like to study the shop to see whether it would be profitable to add another barber or simply add another chair for customers to wait. Perhaps it would be possible to purchase new equipment or provide training, so that the barbers can work even faster. Would the extra haircuts justify the expense of this equipment? GPSS/H will assist in building a model:

1. Predict how the system as outlined above works. The model will only be as good as the input data and the assumptions given above.
2. Once the model represents the barbershop as it presently is working, it can be modified to predict how it will work under different conditions.

This demonstrates another reason why GPSS/H is such a powerful problem-solving tool. As we will discuss, changes in GPSS/H programs often are made by changing only a few lines of code.

The fact that GPSS/H programs can so easily be changed to reflect the what-if?-*type questions a person may want to pose makes it an ideal language to use for simulation studies.*

Once the modeller is satisfied that the original model is correct, the simulation can be redone, but this time with six barbers. Alternately, the model can be run for more seats for customers. Finally, the model can be run for different combinations of haircut speeds for the barbers.

Using the cost data for the various combinations of barbers, lost customers, profit per haircut, and so on, the modeller can determine the economics of the system and make the correct choice.

A Simulation Model

The following example will illustrate a situation of a simulation model with constant arrival rates and constant service rates. Suppose a tool crib has 1 attendant to serve a large group of machinists. These machinists come for a single tool at a uniform rate of 1 every 5 minutes. It takes exactly 6 minutes to obtain the tool. Machinists earn \$20 per hour and the tool crib attendant earns \$16 per hour. The factory works an 8-hour shift but stops for a 1-hour lunch break. The crib is closed for lunch and at the end of the 8-hour shift. In order to simplify the calculations, if a mechanic is waiting for a tool either at the lunch break or at the end of the day, he or she will wait and be served. The tool crib operator does not receive extra pay for working overtime but the mechanics can include the time waiting in their actual working time. Should the company hire another tool crib attendant?

Solution

While this is a very simple situation and one that would rarely be encountered in practice, it will prove instructive to learn what simulation models can tell us. The problem will be solved first for one tool crib attendant, then two, and then for three.

The machinists arrive every 5 minutes, so there will 12 per hour arriving. In a 4-hour time period, 48 will arrive. The first arrives 5 minutes after the tool crib opens. There will be no wait for him. The second person arrives 5 minutes later and will experience a 1 minute wait until the attendant is free. Similarly, the third person has a 2 minute wait, and so on, up to the 48th person, who has a 47 minute wait. In a 4-hour period, there will be a total waiting time for the machinists of $1 + 2 + 3 + \cdots + 47$, or 1128 minutes at \$20 per hour. This represents a loss of $(1128/60) \times \$20$ or \$376. For the two 4-hour periods in a day, this represents a loss of \$752. If two (or more) tool crib attendants are working, there will never be a wait for a free attendant (only the 6 minute wait for the tool). Table I.1 summarizes the results from considering the case of three attendants.

Clearly, it is advantageous to hire one additional attendant. The result of this simple model may or may not be useful to a company, depending on the original hypotheses, which were as follows:

Having an arrival rate of one machinist every 5 minutes. In practice, the arrival rate will be at random times. There may be an average rate of so many per hour but, in general, the arrival time for a particular machinist will be random.

Service rate of the tool crib attendants is constant. Here, too, in practice, the service rate will normally be random.

The tool crib closed every 4 hours. Anyone waiting for service immediately left when the 4 hours was up. In practice, the machinist about to be served still will obtain service.

TABLE I.1

Tool Crib Simulation

Number of attendants	1	2	3
Number of machinists who arrive	48	48	48
Total time waiting for attendant	1128	0	0
Cost of lost time per 4 hours	\$376	0	0
Pay to attendant per 4 hours	\$64	\$128	\$192
Total cost per 8 hours	\$880	\$128	\$192

The length of the queue tended to grow to an eventual size of 47. This is not realistic. If the queue is too long, an arriving machinist will tend to leave and come back later when the line is shorter.

Each arriving machinist wanted only one tool. In practice, the number of tools needed may be 2, 3, or more.

It will be shown that, using the GPSS/H language, a model can be constructed easily to include all of the above possible changes to the original assumptions. Problems such as this are quickly and easily solved with GPSS/H.

Some Comments on Queuing Theory

The example of the queuing problem for the tool crib has an exact solution. There are very few such exact solutions available for most real-life problems. For example, systems involving cyclic queues and/or those involving queues for a finite population for the general cases of random arrival or service times have no solutions. Cyclic queues are those where the system under study has elements that do not leave, such as ships sailing from port to port. Here the ships might be loaded, haul, unload, perhaps load again, and travel to another port. Whenever there is a finite population, as soon as one element is doing a particular thing, the statistical distribution governing rates will change. Thus, if a company has a fleet of 10 cars to be studied, if 2 are being serviced, the probability of another car coming for service is no longer the same as when all 10 were up and running.

In general, in order to study complex systems where queuing takes place, it is necessary to build computer simulation models. It will be shown that GPSS/H is an ideal computer programming language to model such systems. In fact, one of the features of GPSS/H is that, as one learns the language, one automatically learns how to build complex simulation models. Before we actually start learning the software for simulation and animation, it might be instructive to review a few basic concepts from queuing theory. These have to do with the possible arrival distributions, service distributions, number of servers either in series or in parallel, the population size, and the queue discipline. Table I.2 gives the possibilities to consider.

TABLE I.2

Possibilities for Queuing System

1. Population
    - Infinite
    - Finite
2. Arrival time distribution
    - Constant
    - Poisson
    - Erlang
    - Uniform
    - Arbitrary
    - Normal

(Continued)

TABLE I.2 (*Continued*)

Possibilities for Queuing System

3. Service time distribution
    - Exponential
    - Erlang
    - Constant
    - Uniform
    - Arbitrary
    - Changing with the time of the day or queue length
4. Service facility
    - Single
    - More than one in parallel
    - More than one in series
    - Variable number, both in parallel and in series
5. Queue discipline
    - FIFO (first in—first out)
    - Random
    - LIFO (last in—first out)
    - Priority of one type of customer over another for the position in the queue
    - Ability of one customer to pre-empt another one being served
    - Priority of shortest or longest service time being served first or last
    - Balking (customer refuses to join if queue too long)
    - Switching from queue to queue
    - Leaving (customers will leave if waiting too long)
    - Being a member of more than one queue (a person can be in a shopping centre and take a number for meat service at the same time as he or she is waiting for his or her vegetable number to be called).

The following may come as a surprise to the person who has not formally studied queuing theory: it is *not* possible to obtain exact solutions to most of the above situations (although a lot of very fine mathematicians have tried). However, several problems do have solutions and these can be found in textbooks on operations research or queuing theory, such as the one by Phillips et al. (1976). As one learns how to construct simulation models, it is instructive to compare the results from the simulation model with what one expects to obtain from an exact solution.

Simulation versus Mathematical Solution

To illustrate a comparison of a simulation model with one that has an exact solution, consider the case of a store where customers arrive on the average of 24 per hour. The interarrival times are given by sampling from the Poisson distribution. Thus, after each arrival, the next arrival time is obtained by sampling from the Poisson distribution with the same mean as before. The single clerk in the store can handle a customer on the average of 1 every

2 minutes. The distribution for this service is exponential. Service is first-come, first-served. The customers do not mind waiting if there is a queue. It is desired to simulate the store for 50 days or 10 weeks of operation, where a day is 8 hours in duration and the store operates continuously. Compare the results with those obtained by an exact solution.

Solution
The problem will be recognized as a standard one that is discussed in any text on queuing theory. The exact mathematical solution is available and equations can be found for determining the probability of the clerk being idle, the probability of any number of customers being in the store, the expected number of customers in the store, the average time for a customer to wait in the queue, to be in the store, and so on.

GPSS/H is used to solve such problems. The simulation model used Monte Carlo simulation. This technique uses a random number generator to simulate both arrival times and service times. The simulation starts at simulated time $t = 0$ and runs until the program reaches a point in simulated time that the programmer feels is enough to yield correct results. First, a basic time unit needs to be selected. This is normally taken as the smallest time as given by the statement of the problem. For the example here, a time unit of 1 minute is selected. Thus, the customers will arrive on the average of every 2.5 time units. The clerk can handle a customer every 2 time units. The simulation is then done for times of 8, 50, 100, 200, and 400 hours, and so on. These have to be converted to minutes since the basic time unit is a minute.

Since the exact solution assumes steady-state conditions, the simulation is run for 4 hours (240 time units) and then is stopped. All relevant statistics, except for the customers in the system, are discarded. Then, the simulation is restarted and run for the desired simulated time. A selected portion of the output from this program for the simulated time of 400 hours is as follows:

Customers serviced	9605
Percent time clerk busy	801
Average number of customers in system	4.122
Average time in system	10.2 minutes

The theoretical values can be found by use of simple formulas that can be found in any book on operations research or queuing theory.

Customers served	9600 (customers arrive on average of 24/hour for 400)
Percent time clerks busy	800 (this is 24/30)
Average number of customers in system	4
Average time in system	10 minutes

As can be seen, the above results compare quite favourably with those obtained by the simulation. It is important in both interpreting and using the results of a simulation that the simulation has been allowed to run for a long enough period or the results may not be accurate. In performing a simulation, one would like to obtain results that can be reproduced nearly identically if other simulations are done with different random numbers. There is no set answer to the question of how many simulations are enough, as the proper number of time units to simulate for is a function of several variables. One is the nature of the simulation, that is, is the population infinite or finite? In the case just considered of

TABLE I.3

Results of Re-Running Simulation for Store

Simulate Time (hours)	Customers Served per Hour	Percent Time Clerk Busy	Avg. No. of Customers in System	Avg. Time Customers in System
8	26.9	92.2	4.34	9.58
50	23.6	81.7	4.105	7.87
100	24.0	82.6	4.061	10.63
200	23.9	79.7	4.810	11.52
400	24.2	80.3	4.122	10.2
Theoretical	24.0	80.0	4.00	10.2

an infinite population and Poisson arrival times with exponential service times, a large number of simulations have to be performed. In the case of a system where the parameters being simulated cycle through the system (such as workers in a factory) not quite so many simulations may be needed. The nature of the queue and the service facilities are also important. In addition, if the statistical distributions are relatively uniform, such as a normal distribution with small standard deviation, the simulations tend to achieve a level of stability rapidly. This last result is important (and comforting) for the person doing simulations who has a lot of data that is normally distributed. This is often the case for working times in a factory, truck haulage rates along a road, manufacturing times, and so on. If the statistical distributions are non-symmetric, the number of simulations required can be high. This will be demonstrated by means of an example later. First, let us again consider the example just solved.

Suppose, however, that, for the simple queuing system just studied, the simulation was done for less than 400 hours. What would the results have been? The answer depends, in part, on the sequence of random numbers. But it is instructive to redo the simulation for less than 400 hours and examine the results. Table I.3 summarizes the results from these different simulations.

As can be seen, the results for simulating for 8 hours are quite different from the theoretical ones. Simulating for 200 hours yields results that come close to the theoretical ones except for the average number of customers in the system. After 400 hours, the simulated results are quite close to the expected ones. If this problem was for a real store, the simulation may well have been run for an even longer time.

Simulation with Non-Symmetrical Distributions

Whenever the statistical distributions are non-symmetric (the Poisson is non-symmetric, the normal or Gaussian distribution is symmetrical), the number of simulations may have to be very large. This is easy to understand, since it is desirable to model a system over every possible situation and, in theory, repeat the simulation until the various parameters being studied do not change. To illustrate this concept of non-symmetric distribution, consider a simple example. Suppose a person is modelling his behaviour on a day-to-day basis, weekends not included. Each day this person stops at the local casino and bets $2 on number 7 on a roulette wheel. He makes only this bet and, whether he wins or loses, will leave. The probability of winning is 1/38 (the wheel has numbers running from 1 to 36 as

TABLE I.4

Results of Studying Simulation of Roulette Wheel

No. of days	First Set Random Nos.	Second Set Random Nos.	Third Set Random Nos.	
	Obs. Wins	Obs. Wins	Obs. Wins	Exp. Wins
380	8(−20%)	13(+30.0%)	5(−50%)	10
3,800	95(−5.0%)	95(−5.0%)	95(−5%)	100
38,000	1,019(+1.9%)	1,019(+1.9%)	1,017(+1.7%)	1,000
380,000	10,025 (+2%)	10,029(+.2%)	10,021(+.21%)	10,000

well as a zero (0), and double zero (00). How many simulated days are needed to produce satisfactory results for the simulation? Certainly not 38, as the expected number of wins is only 1. How about 380 or 3800? To study this, a short GPSS/H program was written. The simulation was performed for 380; 3,800; 38,000; and, finally, 380,000 days. It was then run for three different sequences of random numbers. Table I.4 summarizes the results of these three simulations.

As can be seen, the observed number of wins versus the expected number of wins start to approach each other only after a very large number of simulations. In fact, the results for 380 simulations give results that vary from the expected number of wins by as much as 50%. Thus, for situations such as the above, one must always be aware that a very large number of simulations may be needed (there's no guarantee that even 380,000 is enough, depending on the problem!). Fortunately, for most situations, the simulation can be performed successfully with a reasonable number of simulations.

Why Do a Simulation?

It always comes as a surprise to mining students to learn that it is rarely possible to obtain exact solutions to any but the most elementary problems that lead to queuing situations. We all experience queuing daily whenever we enter a bank that has many customers and we have to wait for a teller or shop in a large grocery store and wait in the checkout line, and so on. One would think that such problems can be solved quite easily, but this is not the case. Although queuing theory has been studied by mathematicians for many years, very few complex problems have been solved.

A computer simulation can rapidly and accurately solve most elementary and complex situations where one encounters queuing situations. Many mining engineers still do design work using average values, such as the average time a truck takes to be loaded, the average time to travel from point A to point B, and the average time to dump. It will not take long for the student to realize, however, that this type of design leads to incorrect results. For example, if a different person enters a barbershop with a single barber for a haircut exactly every 5 minutes and the time to give each person a haircut is also exactly 5 minutes, then the system is *perfect* in the sense that the barber will always be busy and no person will ever have to wait for the barber. However, if the interarrival time for each customer is obtained by sampling from the Poisson (exponential distribution) with a mean of 5 minutes and the time for a haircut is also exponential with a mean of 5 minutes, then the expected queue of waiting customers becomes infinite! This is one of the few queuing