

# COMPUTER SIMULATION and MODELLING



Francis  
Neelankavil

# COMPUTER SIMULATION AND MODELLING

FRANCIS NEELAMKAVIL

*Department of Computer Science  
Trinity College, Dublin  
Ireland*

JOHN WILEY & SONS

Chichester New York Brisbane Toronto Singapore

Copyright © 1987 by John Wiley & Sons Ltd.

All rights reserved.

No part of this book may be reproduced by any means, or transmitted, or translated into a machine language without the written permission of the publisher.

***Library of Congress Cataloging-in-Publication Data:***

Neelamkavil, Francis.

Computer simulation and modelling.

Includes index.

1. Computer simulation. I. Title.  
QA76.9.C65N44 1986 001.4'34 86-9263

ISBN 0 471 91129 1

ISBN 0 471 91130 5 (pbk.)

***British Library Cataloguing in Publication Data:***

Neelamkavil, Francis

Computer simulation and modelling.

1. Digital computer simulation 2. System analysis

I. Title

003'.0724 QA76.9.C65

ISBN 0 471 91129 1

ISBN 0 471 91130 5 (pbk.)

Typeset by Photo-Graphics  
Printed and bound in Great Britain

COMPUTER SIMULATION  
AND MODELLING

*To  
my parents who made it all possible  
and  
my son Arun, who missed a lot of fun*

# Preface

This book is about modelling and simulation of systems with the aid of computers. Both modelling and simulation (continuous and discrete) aspects are covered in sufficient detail; however, the emphasis is on digital computer simulation of discrete systems.

Modelling and simulation is not a clearly defined subject; it has its roots in several fields and derive its strength from the concepts, theories, and techniques developed in all the major disciplines over the last several decades. It is more than an art, but not a fully developed science. Human judgement, experience and computer programming skill still play an important role in the formulation and solution of problems by this method. Earlier applications of modelling and simulation centred around the problems arising in science and engineering, and it has now become clear that modelling and simulation can make a significant contribution in solving problems in economics, management, social and behavioural sciences.

A number of books on modelling and simulation have appeared on the market during the past 15 years, and the classroom trial of these books highlighted several deficiencies in the structure, organization, presentation, and practical orientation of various aspects of computer modelling and simulation. This book describes all important aspects (concepts, techniques, tools, strengths, and weaknesses) of computer-aided modelling and simulation in a single volume suitable for adoption as a textbook for an undergraduate or a postgraduate course on the subject. Concepts and theories have very little meaning unless they are illustrated through diagrams and examples, which make both teaching and learning enjoyable. Diagrams, numerical examples, sample program listings (Pascal, CSMP, GPSS, SIMSCRIPT) and output from actual computer runs are the main features in the presentation of various topics. The emphasis is on practical problem-solving and not on theorem-proving. Unnecessary proofs and theoretical derivations are avoided and more theoretically motivated readers are referred to selected additional reading materials which in turn will lead to further sources.

This is the age of microprocessors and supercomputers. Interactive computer graphics, animation, artificial intelligence, and expert systems have become usable tools. The computer is no longer a luxury and micros are

currently accessible to the vast majority of students, researchers, and other practitioners. The use of microprocessors for modelling and simulation, and the availability of micro-based simulation software have been stressed throughout the book so as to make simulation an attractive tool for problem-solving in a wide range of disciplines.

Reading is a necessary first step in learning the subject, but what is more important is the actual construction and implementation of simulation models and interpretation of results. The summary and exercises given at the end of each chapter are designed to encourage reading and 'learning by doing', and thus gain experience and confidence in the practice of modelling and simulation.

The reader of this book is assumed to be familiar with basic algebra, calculus, statistics, and at least one of the computer programming languages such as Pascal, Fortran, Basic, etc. No previous knowledge of simulation languages is required. Students in computer science, engineering, natural sciences, operations research, management science, and social sciences will find this book particularly useful. The bulk of the material can be covered in a standard 1-year university course. People from industry and other fields, who are engaged in modelling and simulation projects, should also find this book helpful. A brief outline of topics covered in various chapters appear in Section 1.7 of Chapter 1.

Several people have aided me directly or indirectly in writing this book. In particular, I would like to thank my colleagues at the Department of Computer Science, Trinity College, Dublin and University of Kansas (1984–85) for their constructive criticisms and suggestions; Stella Coughlan for her help in the preparation of the manuscript, and all my students who served as a test population during the last so many years. Finally, I am grateful to A. Mullarney of CACI, Dublin, for his comments on SIMSCRIPT in Chapter 9.

Francis Neelamkavil

# Contents

<b>Preface</b>	xv
<b>1 Introduction and Overview</b>	1
1.1 Background	1
1.2 Principle of computer modelling and simulation	2
1.3 Simulation versus Monte Carlo simulation	3
1.4 Nature of computer modelling and simulation	6
1.4.1 Continuous versus discrete approach	7
1.4.2 Computer modelling and simulation process	8
1.4.3 The impact of microprocessors and computer graphics	11
1.5 When to use simulation	12
1.6 Limitations of simulation	12
1.7 The organization of this book	13
1.8 Summary	14
1.9 Exercises	14
References	15
<b>2 Systems</b>	17
2.1 Introduction	17
2.2 Systems concept	17
2.3 Systems approach to problem-solving	19
2.4 Characteristics of systems	21
2.5 The state of a system	22
2.6 Systems boundary and environment	24
2.7 Different views of systems	25
2.8 Summary	26
2.9 Exercises	26
References	27
Bibliography	27
<b>3 Models</b>	29
3.1 Introduction	29
3.2 What is a model?	30



3.3	Why study models?	31
3.4	Types of models	32
3.4.1	Physical models	32
3.4.2	Symbolic models	33
3.4.3	Mental models	35
3.4.4	Other models	36
3.5	Modelling methodology	37
3.5.1	Example 1	39
3.5.2	Example 2	45
3.6	Models from various disciplines	50
3.6.1	Biological model	50
3.6.2	Optimum timing problem	52
3.6.3	Economic system	55
3.6.4	Bending of beams	56
3.6.5	Mechanical system	57
3.6.6	Electrical system	59
3.6.7	Thermal system	60
3.7	Summary	62
3.8	Exercises	62
	References	65
	Bibliography	66
<b>4</b>	<b>Model Validation</b>	67
4.1	Introduction	67
4.1.1	Hypotheses, theories, and laws	67
4.1.2	Inductivism	67
4.1.3	Deductivism	69
4.1.4	Scientific method	69
4.2	Modelling as a scientific process	70
4.3	Possible approach to validation	70
4.3.1	Validity of concepts	71
4.3.2	Validity of methodology	72
4.3.3	Validity of data	73
4.3.4	Validity of results	74
4.3.5	Validity of inference	75
4.4	Validation in practice	76
4.5	Validation versus verification	76
4.6	Summary	78
4.7	Exercises	79
	References	79
	Bibliography	80
<b>5</b>	<b>Basic Probability and Statistics</b>	81
5.1	Introduction	81
5.2	Permutations and combinations	81

5.3	Sets	82
5.4	Definition of probability	82
5.5	Conditional probability	83
5.6	Bayes' theorem	84
5.7	Discrete distributions	85
5.8	Continuous distributions—probability density and distribution functions	86
5.9	Uniform distributions	87
5.10	Joint distributions	89
5.11	Expectation—discrete distributions	90
5.12	Expectation—continuous distributions	90
5.13	Variance, standard deviation, and moments	90
5.14	Correlation and covariance	91
5.15	Hypothesis testing, significance levels, and confidence intervals	94
5.16	Simple queuing models	95
5.17	Basic results from queuing theory	96
5.17.1	Poisson process	96
5.17.2	Single-server queues	97
5.17.3	Multiple-server queues	99
5.18	Summary	100
5.19	Exercises	102
	References	104

<b>6</b>	<b>Random Numbers and Random Variates</b>	<b>105</b>
6.1	Introduction	105
6.2	Generation of random numbers	105
6.3	Von Neumann's midsquare method	108
6.4	Linear congruential generators	108
6.4.1	Selection of constants	109
6.5	Multiplicative congruential generators	109
6.6	Other pseudo-random number generators	111
6.7	Pseudo-random number generators for microcomputers	112
6.8	Testing of pseudo-random numbers	112
6.8.1	Moments test	113
6.8.2	Frequency test using chi-square method	114
6.8.3	Serial test	115
6.8.4	Runs test	116
6.8.5	Gap test	117
6.8.6	Kolmogorov–Smirnov test (KS test)	118
6.8.7	Other tests	119
6.9	Generation of random variates	119
6.10	Inverse transformation method	120
6.11	Rejection method	120

6.12	Generation of continuous random variates	121
6.12.1	Uniform distribution	121
6.12.2	Exponential distribution	122
6.12.3	Normal distribution	123
6.12.4	Empirical distributions	126
6.12.5	Other distributions	128
6.13	Generation of discrete random variates	128
6.13.1	Poisson distribution	128
6.13.2	Binomial (Bernoulli) distribution	129
6.13.3	Empirical distributions	129
6.14	Summary	130
6.15	Exercises	131
	References	133
<b>7</b>	<b>Discrete System Simulation</b>	<b>135</b>
7.1	Introduction	135
7.2	Event, activity, and process	135
7.3	Representation of time	136
7.4	Analysis of discrete systems—a simple example	138
7.4.1	Approximate method	140
7.4.2	Theoretical method	141
7.4.3	Simulation	143
7.5	Generation of the time and type of next event	145
7.5.1	Standard statistical event distributions	145
7.5.2	Interpolated continuous event distributions	149
7.5.3	Discrete event distributions	149
7.5.4	List-type event distributions	149
7.6	Simulation strategies	150
7.6.1	Event scheduling	151
7.6.2	Process interaction	152
7.6.3	Activity scanning	154
7.7	Computer languages for simulation	155
7.7.1	Characteristics of a good language	155
7.7.2	General-purpose versus simulation languages	155
7.7.3	Comparison of major simulation languages	157
7.7.4	Selection of a simulation language	157
7.8	Summary	157
7.9	Exercises	158
	References	159
<b>8</b>	<b>General-purpose Simulation System (GPSS)</b>	<b>160</b>
8.1	Introduction	161
8.2	Basic concepts and overview	161
8.3	Source statement format	164

8.4	A simple GPSS simulation program—example 1	165
8.4.1	GPSS output	168
8.4.2	Possible errors	169
8.4.3	Effects of deleting selected blocks from GPSS program	169
8.5	Parameters and Savevalues	172
8.6	Standard Numerical Attributes (SNA)	174
8.7	Tabulated output	175
8.8	Indirect addressing	176
8.9	Random numbers, Variables, and Functions	177
8.9.1	Random number generators in GPSS	177
8.9.2	Arithmetic, floating point, and boolean variables	177
8.9.3	Discrete, Continuous, and List functions	179
8.10	GPSS example 2	180
8.10.1	Simple GPSS program for example 2	180
8.10.2	Another GPSS program for example 2	180
8.10.3	Third GPSS program for example 2	183
8.11	Generation and Termination of transactions (temporary entities)	183
8.12	Operations on permanent entities	185
8.12.1	Facilities	185
8.12.2	Storages	186
8.12.3	Queues	186
8.12.4	Logic switches	187
8.13	Manipulation of entity attributes	187
8.14	Mechanisms for transaction flow control	189
8.14.1	The TRANSFER block	189
8.14.2	The GATE block	190
8.14.3	The TEST block	191
8.15	Mechanisms for delaying transactions	191
8.16	Simulation control statements	192
8.17	GPSS example 3 and example 4	193
8.17.1	Example 3—use of STORAGE, GATE, and SAVEVALUE	193
8.17.2	Example 4—use of TEST, TABULATE, RESET, and CLEAR	195
8.18	GPSS on microprocessors	199
8.19	Summary	199
8.20	Exercises	199
	References	205
<b>SIMSCRIPT</b>		207
9.1	Introduction	207
9.2	A simple SIMSCRIPT simulation program: example 1	209

9.3	Source format	215
9.4	Variables and constants	215
9.5	Simulation clock	216
9.6	Expressions	216
9.7	IF, GO TO, and LOOP statements	217
9.8	Random distribution functions	218
	9.8.1 Built-in distributions	218
	9.8.2 User-defined distributions	219
9.9	SIMSCRIPT example 2	220
9.10	Creation and Termination of temporary entities	220
9.11	Creation of permanent entities	222
9.12	The concept of Sets	223
9.13	READ, LIST, and PRINT statements	224
9.14	Collection of statistics	225
9.15	SIMSCRIPT example 3	226
9.16	SIMSCRIPT on microprocessors	226
9.17	Summary	227
9.18	Exercises	228
	References	232
<b>10</b>	<b>Simulation Experiments in Practice</b>	<b>233</b>
10.1	Collection and analysis of simulation output	233
10.2	Major sources of errors in simulation	234
	10.2.1 Modelling errors	234
	10.2.2 Computer programming errors	234
	10.2.3 Sampling errors	235
	10.2.4 Errors in parameter estimation	235
10.3	Error control in simulation	236
	10.3.1 Detection and control of initial bias	236
	10.3.2 Replication of simulation runs	238
	10.3.3 Batch mean	240
	10.3.4 Regenerative method	240
	10.3.5 Time series and spectral analyses	241
10.4	Variance reduction techniques	241
	10.4.1 Antithetic sampling	242
	10.4.2 Common random numbers (CRN)	243
	10.4.3 Control variates	243
10.5	Design of simulation experiments	244
10.6	Validation of simulation results	246
10.7	Summary	247
10.8	Exercises	248
	References	249
<b>11</b>	<b>Continuous System Simulation</b>	<b>251</b>
11.1	Introduction	251
	11.1.1 Direct digital versus digital analogue simulation	251

11.1.2	General characteristics of analogue and digital computers	252
11.1.3	Hybrid computers and hybrid simulations	252
11.2	Analogue computer simulation	253
11.2.1	Basic operations and notations	253
11.2.2	Linear systems	255
11.2.3	Simultaneous equations	256
11.2.4	Nonlinear systems	256
11.2.5	Magnitude and time scaling	258
11.2.6	Analogue computer operations	259
11.2.7	Analogue computer simulation—summary of steps	259
11.3	Digital computer simulation using CSMP (Continuous System Modelling Program)	260
11.3.1	Historical background	260
11.3.2	Overview of CSMP	261
11.3.3	Symbolic names, constants, and source statement format	263
11.3.4	A simple CSMP simulation—example 1	263
11.3.5	Data structure and control statements—example 2	264
11.3.6	INITIAL, DYNAMIC, and TERMINAL segments—example 3	269
11.3.7	SORT and NOSORT options—example 4	273
11.3.8	Integration methods in CSMP	278
11.3.9	Some features of CSMP different from Fortran	279
11.3.10	CSSL on microprocessors	280
11.4	Summary	280
11.5	Exercises	281
	References	285
<b>12</b>	<b>Trends in Simulation</b>	287
12.1	Introduction	287
12.2	Some recent simulation applications	287
12.3	Simulation methodology and software	288
12.3.1	The development of specification languages	288
12.3.2	The credibility and acceptability of simulations	289
12.3.3	ROSS and ESL simulation languages	290
12.3.4	Graphics and Animation	291
12.3.5	Ada as a simulation language	292
12.3.6	Artificial intelligence and Expert systems	293
12.4	Simulation on microprocessors	294
12.5	The era of supercomputers	295
12.6	Summary	296
	References	297
<b>Index</b>		301

# 1

## Introduction and Overview

### 1.1 Background

A system is a collection of interacting elements or components that act together to achieve a common goal. Systems can be studied by direct experimentation, by building prototypes, or by building mathematical/logical models. The experimentation with the prototype or the real system is highly undesirable due to a variety (expensive, impractical, risky, or time-consuming) of reasons, and these methods are not dealt with in this book. The purpose of systems study through modelling is to aid the analysis, understanding, design, operation, prediction, or control of systems without actually constructing and operating the real thing. The mathematical/logical models which are not easily amenable to conventional analytic or numeric solutions form a subset of models generally known as *simulation models*. A given problem defined by a mathematical/logical model can have a *feasible solution*, *satisfactory solution*, *optimum solution*, or *no solution* at all. Computer modelling and simulation studies are primarily directed towards finding *satisfactory solutions* to practical problems. In Figure 1.1, points inside the shaded area are all feasible solutions and any point close to the optimum solution can be treated as a satisfactory solution.

Modelling and Simulation emerged (revived) as an identifiable numerical problem-solving technique during World War II when the so-called *Monte Carlo methods* (Hammersley and Handscomb, 1964) were successfully used by John Von Neumann and Stanislaw Ulam of Los Alamos scientific laboratory for solving neutron diffusion problems. Simulation need not necessarily involve computers. However the evolution of modern computers and high-level simulation languages popularized (Zeigler, 1979) the application of modelling and simulation for solving real-life problems in several disciplines, and the expected advances in computer technology indicate that this trend will continue.

Computer modelling and simulation was introduced into university curricula in the 1960s, and books (Tocher, 1963; Forrester, 1961) and periodicals (Ören, 1974, 1976) on the subject began to appear around the same time. The multidisciplinary character of modelling and simulation is evident from

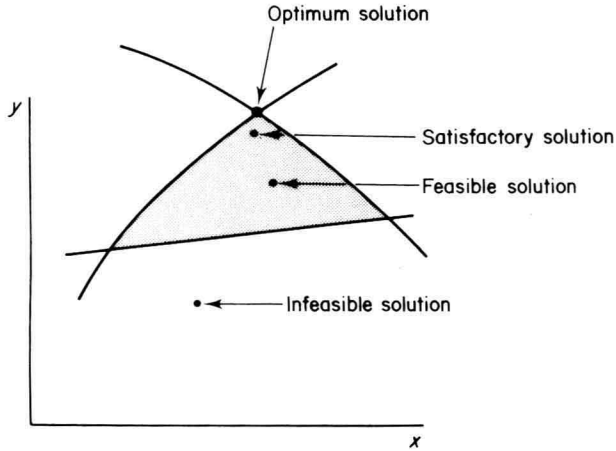


Figure 1.1 Types of solutions to a problem.

the fact that the subject is taught by different departments in different universities around the world.

## 1.2 Principle of computer modelling and simulation

The method of problem-solving by computer modelling and simulation can be best explained by a simple example. In Figure 1.2 the shaded area  $A$  bounded by the non-negative function  $y=f(x)$ , the  $x$  axis and the vertical lines at  $x=b$  and  $x=c$  is given by the definite integral

$$A = \int_b^c f(x) dx \quad (1.1)$$

where  $0 \leq f(x) \leq h$  and  $b \leq x \leq c$ .

The area  $A$  is a portion of the total area enclosed by the rectangle  $R$  with base  $(b,c)$  and height  $h$ . If a point  $P$  is dropped at random on  $R$  (this can be visualized as the throwing of a dart  $P$  from a distance on to the plane  $R$  so that the probability of hitting any point on  $R$  is the same), then the probability  $p$  of the point  $P$  falling in the area  $A$  is given by

$$p = \frac{A}{h(c-b)}$$

If the point  $P$  is dropped  $N$  times and  $K$  times it fell within the area  $A$ , then

$$p = \frac{K}{N} = \frac{A}{h(c-b)} \quad \text{when } N \rightarrow \infty$$



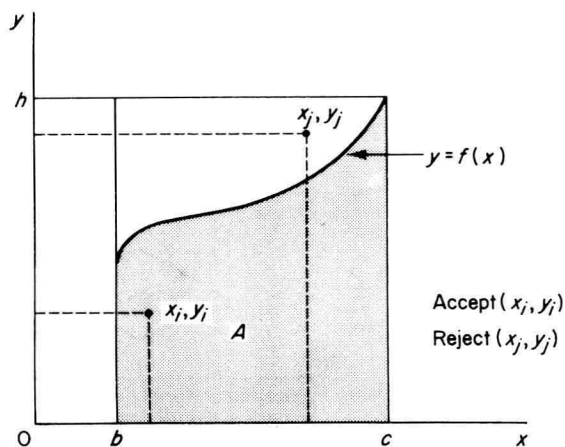


Figure 1.2 Area under the curve  $y = f(x)$ .

or

$$A = \frac{Kh(c-b)}{N} \quad (1.2)$$

Figure 1.3 shows a simple flowchart to determine the area  $A$  by computer simulation, using the model defined by equation (1.2). A Pascal program and the actual computer (VAX 11/780) output are given in Figure 1-4 where  $N=3000$ ,  $b=0$ ,  $c=h=1$ ,  $y=f(x) = \sqrt{1-x^2}$  and a standard library routine MTH\$RANDOM is generating random numbers in the range (0,1).

The computer modelling and simulation method of problem-solving (here, the evaluation of a definite integral) becomes attractive only when it is difficult, time-consuming, expensive, hazardous, or impossible to solve the problem by conventional analytic, numeric, or physical experimentation methods. It must be clear by now that simulation results are likely to be less accurate than analytic solutions.

### 1.3 Simulation versus Monte Carlo simulation

The city of Monte Carlo is well known for its gambling (roulette wheel) activities, and hence this name has come to be associated with simulation (Von Neumann gave the code name Monte Carlo for his military project at Los Alamos) which involves the generation and use of random or chance variables. Some authors use the terms *Monte Carlo simulation* and *simulation* (discrete system simulation) synonymously. However, there is a distinction between the two. Both simulation and Monte Carlo simulation (Shreider, 1966) imply that the simulation model is influenced by random events;