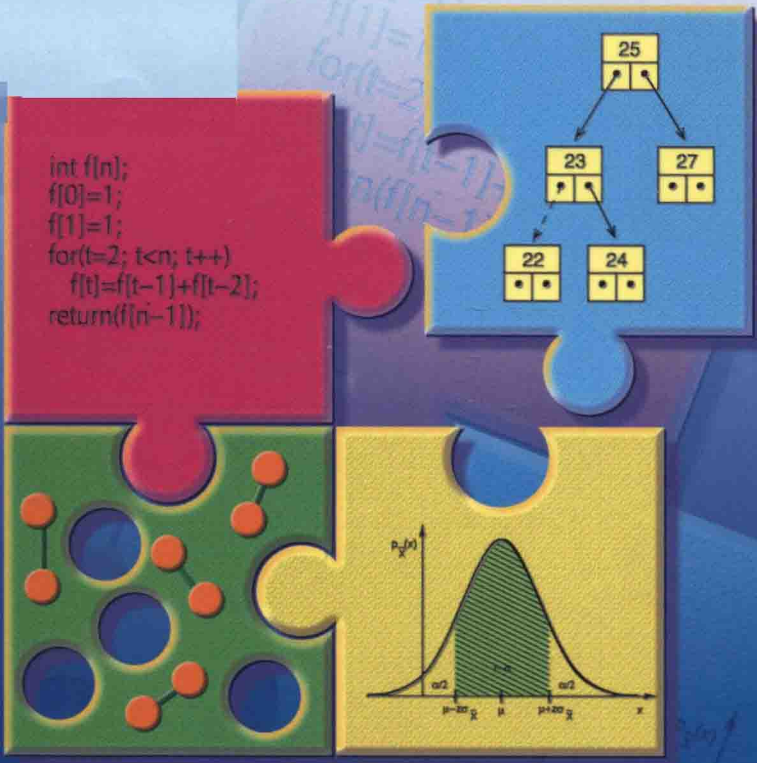
Alexander K. Hartmann

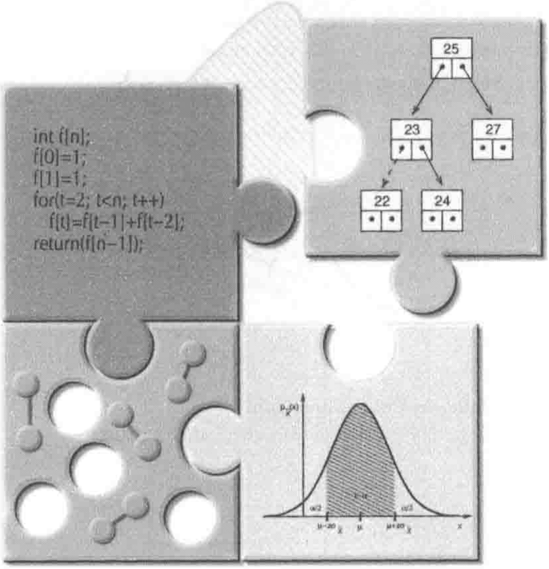


```
int f[n];
f[0]=1;
f[1]=1;
for(t=2; t<n; t++)
   f[t]=f[t-1]+f[t-2];
return(f[n-1]);
```

BIG PRACTICAL GUIDE TO Computer Simulations

Second Edition

World Scientific



```
int f[n];
f[0]=1;
f[1]=1;
for(t=2; t<n; t++)
   f[t]=f[t-1]+f[t-2];
return(f[n-1]);
```

BIG PRACTICAL GUIDE TO Computer Simulations

Second Edition

Alexander K Hartmann
University of Oldenburg, Germany

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

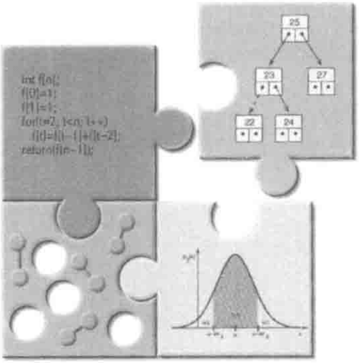
World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

BIG PRACTICAL GUIDE TO COMPUTER SIMULAITONS
2nd Edition

ISBN 978-981-4571-76-0
ISBN 978-981-4571-77-7 (pbk)

Printed in Singapore



```
int f[n];
f[0]=1;
f[1]=1;
for(i=2; i<n; i++)
  f[i]=f[i-1]+f[i-2];
return(f[n-1]);
```

BIG PRACTICAL GUIDE TO

Computer Simulations

Second Edition

To my family

Preface to the 2nd edition

After the first edition of this book was published, I received a lot of supporting comments from various people. Nevertheless, many readers recommended some further subjects to be included in the book, in particular related to new developments or tools. Furthermore, several misprints and little mistakes were reported by them. Also, for some paragraphs, I decided to rewrite them a bit or expand them by including more examples. Hence, I did not hesitate for long, when World Scientific asked me to write a second edition. You are holding the result of this effort in your hands.

The main extensions of the second edition are as follows:

- An introduction to the programming language *Python* has been added, see Sec. 2.3. This language is half-way between a script language (since it is interpreted) and a full high-level programming language (it offers, e.g. complex data structures, object-oriented elements and many libraries for various purposes). Thus, when using *Python*, on the one hand it is easy to write scripts for various tasks like data analysis. On the other hand, small to medium size simulation project can be implemented rapidly.
- Since simulation and analysis of *networks and graphs* has become more and more important in various fields of science, the Sec. 6.8 on graphs has been extended by a subsection (6.8.4) on the calculation of the *connected component*. This is one of the most-often performed analyses for graphs and the basis of many complex algorithms, e.g. to obtain shortest paths, diameters, or centrality measures.
- The Sec. 8.1.1 on how to *draw random numbers* for discrete random variables now includes the fastest available algorithm. It allows one to draw each random number according to the given distribution in

constant time $O(1)$. This is much faster compared to the standard approach which takes $O(\log N)$, where N is the number of the possible outcomes.

- The Sec. 8.2.4 covers the *rejection method*, which allows one to draw random numbers according to a given probability density, usually applied for non-invertible distribution functions. The section has been extended by the general approach to border the target distribution by arbitrary probability density functions. This enlarges strongly the range of distributions which can be realized by this approach.
  Furthermore, as an example for generating random points in higher dimensions, drawing points uniformly on the surface of *d-dimensional hyper sphere* is included now.
- In Chap. 8 on data analysis, in the Sec. 8.5 on hypothesis test and measuring correlations in data sets, a subsection on the *Receiver-operator characteristic* (ROC) has been included. The ROC is an advanced yet widespread approach which allows for the determination of optimal parameters for hypothesis tests.
- Furthermore a subsection (8.5.5) on the *principal-component analysis* has been added. This is the simplest approach to search for the most important "directions" in high-dimensional data.
- *Data clustering* methods are an important tool in the analysis of big data sets, since they allow for looking for structure in overwhelming amounts of data. This is a very sophisticated way of data analysis, much beyond the level of averages, histograms and curve fitting. Several approaches, ranging from the most-simple k-means method to the advanced hierarchical clustering algorithms are now explained extensively in Sec. 8.5.6.
- The Sec. 8.6.2 on *fitting* data using *gnuplot* has been extended by how to restrict ranges of fitting parameters (which is not possible by default) and how to fit functions to multiple sets of data simultaneously.
- In the first edition, *gnuplot* was only used to generate "quick and dirty" plots of data. In Sec. 9.2.1 it is now shown how one can generate publication-quality plots using *gnuplot*.
- The Sec. 9.2.4 on the ray tracer *Povray* has been extended by some paragraphs which explain how this ray tracer can be used to generated three-dimensional pictures of simulation snapshots. By using sequences of such figures, small movies can be generated easily.

- In the first edition, the Sec. 9.3.1 on LaTeX, a standard typesetting tool to generate high-quality documents as large as complete books (like this one), was a rather short introduction giving few examples. For the second editions this section has been expanded substantially, such that it now serves as a *complete introduction to LaTeX*, presenting the most-frequently used LaTeXelements in scientific publications, such that all necessary knowledge to generate standard to advanced manuscripts (from the typesetting point of view) is covered.

These new sections include and discuss, following the hands-on approach of the first edition, all necessary source codes such that the reader can apply the new knowledge to her or his case as quickly as possible. Thus, I hope that this new edition will allow you to learn all technical aspects of scientific computer simulations as fast as possible, ranging from the initial idea over the implementation, performing the simulations, doing the data analysis, up to the publication of your results. Thus, using this book will save you a lot of time which you can use to concentrate on the actual scientific problem you want to solve. Finally, since the new edition comprises almost all non-problem specific knowledge you need in computational science, I decided to rename the book to "Big Practical Guide to Computer Simulations".

This new edition would not have been possible without the help of many people. I am grateful to the following persons for communicating mistakes, making useful suggestions and providing extensions of the book: Pia Backmann, Jan Christoph Bernack, Gunnar Claussen, Timo Dewenter, Florian Effenberg, Pascal Fieth, Nikolai Gagunashvili, Hendrike Heidemann, Iwo Ilnicki, Simon Knowles, Karsten Looschen, Markus Manssen, Andreas Mohrs, Zacharais Njam Mokom, Oliver Melchert, Marc Mézard, Tudor Mitran, Christoph Norrenbrock, Tom Seren, Hendrik Schawe, Sagar Sinha, Verena Sterr, Sebastian von Ohr, and A. Peter Young. Furthermore, Oliver Melchert provided a Python script for data resampling which is included in the new edition.

Alexander K. Hartmann
Oldenburg, July 2014

Preface to the 1st edition

You have decided to become an expert in computer simulations, congratulations! This is a good decision, because computational and simulation methods become more and more important in all areas of science, humanities, economy, engineering and mathematics. This book will help you a great deal in learning the most important basics which you need during all phases of a simulation research project, from the phases of program design, implementation, and debugging, through running the simulations, organizing the data and analyzing the results, to the final phase where you want to present and publish your results.

Note that nowadays thousands of different problems exist which are being investigated by computer simulations. One studies, for example, the diffusion of chemicals in soils, the folding of proteins in cells, the communication of neurons in the brain, the deformation of cars in accidents, the behavior of brokers working at the stock market, the evolution of the weather during the next days or weeks, the turbulent behavior of flowing water in a turbine, the movement of electrons in semiconductors, the patterns of words in languages, or the traffic of pedestrians in crowded shopping malls, to name only a few. Consequently, there are many algorithms to treat the variety of problems, for example, finite differences, finite elements, integration methods, matrix inversion, eigenvalue determination, equation solvers, molecular dynamics simulations, Monte Carlo methods, density functional approaches, graph algorithms, optimization methods, and so on. Which methods are suitable for your problems depends heavily on the problems you want to solve. Since there are way too many algorithms available for all these different problems, and because everybody needs usually a different one, these special-purpose algorithms are *not* covered in this book.

This book instead covers methods, techniques and algorithms, which you *always* have to apply, independent of the actual simulation research project you are considering. Here, *practical* aspects of conducting research via computer simulations are discussed. An overview is given towards the end of this preface. After reading this book, you only need some additional information about the specific project your are considering, usually provided in scientific papers, and maybe you need a second special-purpose book which you have to get. Then you are ready to start!

The book addresses people who have no or little experience with computer simulations. This book is in particular suited for students who want to start a project, like a PhD thesis, in the field of computer simulations. But also researchers who have conducted already some simulation projects may find a lot of the advanced material helpful. It is assumed that the reader is familiar with an operating system such as UNIX (e.g. Linux), a high-level programming language such as C, JAVA, Fortran or Pascal and has some experience with at least tiny software projects.

Throughout the book, because of the limited space, usually only short introductions to the specific areas are given, as "ready-to-use recipes". The material usually is presented here in a learning-by-example manner. Nevertheless, the material is extensive enough to provide a fundamental set of tools to perform all standard tasks when creating and performing simulations. In addition, references to more specialized literature are cited, allowing specific subjects to be studied more extensively. Most examples of code are in C/C++. Many examples, also solutions to exercises, are available on the website. This is indicated by a small `GET SOURCE CODE` box in the text. Also some freely available documentation is contained on the website. For details, see the appendix.[1]

`GET SOURCE CODE`
DIR: `c-programming` FILE(S): `first.c`

Next, I give you some idea, how the book was realized. In fact, the work on this book started when I was doing simulations for obtaining my first university degree. I had to develop new algorithms and implement them. I had to do many large-scale simulations on a parallel computing cluster. The data had to be analyzed, usually in many different ways, and over-and-over again when new data became available. Finally, the results had to be presented and summarized in a scientific paper. These basic steps remained my main occupation during my PhD and during my first

[1] All supplementary materials can be downloaded from the website: http://www.worldscientific.com/r/9019-supp.

post-doc projects. I became more and more experienced and refined my approaches. I also improved my ways to work by reading books about software engineering, algorithms, data structures and data analysis, as well as by learning to use many programs. After the first-post doc years, I began to supervise students. Hence, I started to pass on my knowledge, trying to help other people to avoid many pitfalls and to devise highly efficient programs. Some of my experiences found their way to the last chapter of the book of Heiko Rieger and myself with the title "Optimization Algorithms in Physics". That chapter is in fact a very short version of the present book. Thus, it served as a seed of the this book and some material of the old chapter appears occasionally here.[2] During the years, I supervised more and more students, and even gave a university course on the practical aspects of computer simulations. This course contained a lot of new material compared to the book chapter and in fact it served as a seed for the present book. Nevertheless, the material I used for supervising students was still collected from several different sources, often not quite compatible with each other. I started to feel that it would help my work, and also increase the efficiency, if I wrote a full book about practical aspects of computer simulations, which should contain "all" needed material comprehensively. After two more years, I received an email from *World Scientific*, where my course web page was noticed. They asked me whether I would like to write a book, based on the course. Now it was not difficult to come to the decision indeed to realize the book. You are now holding the result in your hands.

Note that this book contains a very personal view of which tools are considered useful. Very often, I present several independent tools, such as tools for editing, compiling and analyzing programs rather than one all-purpose environment which usually contains a framework just integrating these basic tools. Nevertheless, most of the tools introduced are standard programs and available on all computer systems (for Microsoft operating systems they sometimes have other names).

Here, I give an overview over the contents of the book. First, a short introduction to C programming is given. Also related topics like macros, *make files* and shell scripts are touched. In the second chapter, the main ideas of software engineering are explained and several hints allowing the construction of efficient and reliable code are stated. In Chap. 4, three very

[2] taken from A.K. Hartmann and H. Rieger, *Optimization Algorithms in Physics*, pp. 293–357, 2002, Copyright Wiley-VCH Verlag GmbH & Co. KGaA. Reproduced with permission.

useful debugging tools are presented, which will help you to hunt down bugs in your programs quickly. In Chap. 5, a short primer on object-oriented software development is presented. In particular, it is shown that this kind of programming style can be achieved with standard procedural languages such as C as well, but also how C++ can be used. Next, basic types of algorithms and advanced data structures are explained. These can be used as auxiliary tools to create highly-professional, efficient simulation programs. In the subsequent chapter, the benefit of using libraries like the *Standard Template Library* and the *GNU Scientific Library* is explained and it is shown how you can build your own libraries. In Chap. 8, aspects of probability theory, random-number generation, data analysis, plotting data and curve fitting are covered. In the last chapter, an introduction to information retrieval and literature search in the Internet and to the preparation of presentations and publications is given.

I am indebted to all my colleagues for countless hours of joyful collaborations, which laid the foundations of this book. I am very grateful to Angelika Sievers for thoroughly reading all chapters of the book while checking for typos, language and grammar mistakes. Finally, I would like to thank Björn Ahrens, Luis Apolo, Bernd Burghardt, Niels Hoelzel, Magnus Jungsbluth, Reinhard Leidl, Oliver Melchert, Axel Schulz, Bruno Sciolla and Stefan Wolfsheimer for critically reading manuscript chapters and for giving me many useful comments which helped me to improve the book and to remove many typos.

Alexander K. Hartmann
Oldenburg, November 2008

Alexander Karl Hartmann is Professor of Computational Physics of the Department of Science at the University of Oldenburg (Germany). He obtained a M.S. in Computer Science at the University of Hagen and a M.S. in Physics at the University of Duisburg. After receiving his Ph.D. in Theoretical Physics at the University of Heidelberg, he worked at the University of Göttingen, the University of California in Santa Cruz and the Ecole Normale Supérieure in Paris. In 2007, he moved to Oldenburg. His research interests comprise computer simulations, disordered systems, combinatorial optimization, large deviations and Bioinformatics. Professor Hartmann is author or co-author of 8 books and more than 120 scientific publications.

Contents