

Preface

This book is not a computer simulation cookbook. Our aim is to explain the physics that is behind the “recipes” of molecular simulation. Of course, we also give the recipes themselves, because otherwise the book would be too abstract to be of much practical use. The scope of this book is necessarily limited: we do not aim to discuss all aspects of computer simulation. Rather, we intend to give a unified presentation of those computational tools that are currently used to study the equilibrium properties and, in particular, the phase behavior of molecular and supramolecular substances. Moreover, we intentionally restrict the discussion to simulations of *classical* many-body systems, even though some of the techniques mentioned can be applied to quantum systems as well. And, within the context of classical many-body systems, we restrict our discussion to equilibrium phenomena.

The book is aimed at readers who are active in computer simulation or are planning to become so. Computer simulators are continuously confronted with questions concerning the choice of technique, because a bewildering variety of computational tools is available. We believe that, to make a rational choice, a good understanding of the physics behind each technique is essential. Our aim is to provide the reader with this background.

We should state at the outset that we consider some techniques to be more useful than others, and therefore our presentation is biased. In fact, we believe that the reader is well served by the fact that we do not present all techniques as equivalent. However, whenever we express our personal preference, we try to back it up with arguments based in physics, applied mathematics, or simply experience. In fact, we mix our presentation with practical examples that serve a twofold purpose: first, to show how a given technique works in practice, and second, to give the reader a flavor of the kind of phenomena that can be studied by numerical simulation.

The reader will also notice that two topics are discussed in great detail, namely simulation techniques to study first-order phase transitions, and various aspects of the configurational-bias Monte Carlo method. The reason why we devote so much space to these topics is not that consider them to be more important than other subjects that get less coverage, but rather be-

cause we feel that, at present, the discussion of both topics in the literature is rather fragmented.

The present introduction is written for the nonexpert. We have done so on purpose. The community of people who perform computer simulations is rapidly expanding as computer experiments become a general research tool. Many of the new simulators will use computer simulation as a tool and will not be primarily interested in techniques. Yet, we hope to convince those readers who consider a computer-simulation program a *black box*, that the inside of the black box is interesting and, more important, that a better understanding of the working of a simulation program may greatly improve the efficiency with which the black box is used.

In addition to the theoretical framework, we discuss some of the practical tricks and rules of thumb that have become “common” knowledge in the simulation community and are routinely used in a simulation. Often, it is difficult to trace back the original motivation behind these rules. As a result, some “tricks” can be very useful in one case yet result in inefficient programs in others. In this book, we discuss the rationale behind the various tricks, in order to place them in a proper context. In the main text of the book we describe the theoretical framework of the various techniques. To illustrate how these ideas are used in practice we provide Algorithms, Case Studies and Examples.

Algorithms

The description of an algorithm forms an essential part of this book. Such a description, however, does not provide much information on how to implement the algorithm efficiently. Of course, details about the implementation of an algorithm can be obtained from a listing of the complete program. However, even in a well-structured program, the code contains many lines that, although necessary to obtain a working program, tend to obscure the essentials of the algorithm that they express. As a compromise solution, we provide a pseudo-code for each algorithm. These pseudo-codes contain only those aspects of the implementation directly related to the particular algorithm under discussion. This implies that some aspects essential for using this pseudo-code in an actual program have to be added. For example, the pseudo-codes consider only the x directions; similar lines have to be added for the y and z direction if the code is going to be used in a simulation. Furthermore, we have omitted the initialization of most variables.

Case Studies

In the Case Studies, the algorithms discussed in the main text are combined in a complete program. These programs are used to illustrate some elemen-

tary aspects of simulations. Some case studies focus on the problems that can occur in a simulation or on the errors that are sometimes made. The complete listing of the FORTRAN codes that we have used for the case studies is accessible to the reader through the Internet.¹

Examples

In the Examples, we demonstrate how the techniques discussed in the main text are used in an application. We have tried to refer as much as possible to research topics of current interest. In this way, the reader may get some feeling for the type of systems that can be studied with simulations. In addition, we have tried to illustrate in these examples how simulations can contribute to the solution of “real” experimental or theoretical problems.

Many of the topics that we discuss in this book have appeared previously in the open literature. However, the Examples and Case Studies were prepared specifically for this book. In writing this material, we could not resist including a few computational tricks that, to our knowledge, have not been reported in the literature.

In computer science it is generally assumed that any source code over 200 lines contains at least one error. The source codes of the case studies contain over 25,000 lines of code. Assuming we are no worse than the average programmer this implies that we have made at least 125 errors in the source code. If you spot these errors and send them to us, we will try to correct them (we can not promise this!). It also implies that before you use part of the code yourself you should convince yourself that the code is doing what you expect it to do.

In the light of the previous paragraph, we must add the following disclaimer:

We make no warranties, express or implied, that the programs contained in this work are free of error, or that they will meet your requirements for any particular application. They should not be relied on for solving problems whose incorrect solution could result in injury, damage, or loss of property. The authors and publishers disclaim all liability for direct or consequential damages resulting from your use of the programs.

Although this book and the included programs are copyrighted, we authorize the readers of this book to use parts of the programs for their own use, provided that proper acknowledgment is made.

Finally, we gratefully acknowledge the help and collaboration of many of our colleagues. In fact, many dozens of our colleagues collaborated with

¹<http://www.hpcn.tudelft.nl/frenkel.smit.html>

us on topics described in the text. Rather than listing them all here, we mention their names at the appropriate place in the text. Yet, we do wish to express our gratitude for their input. Moreover, Daan Frenkel should like to acknowledge numerous stimulating discussions with colleagues at the FOM Institute for Atomic and Molecular Physics in Amsterdam and at the van 't Hoff Laboratory of Utrecht University, while Berend Smit gratefully acknowledges discussions with several colleagues at Shell. In addition, several colleagues helped us directly with the preparation of the manuscript, by reading the text or part thereof. They are: Giovanni Ciccotti, Mike Deem, Simon de Leeuw, Toine Schlijper, Stefano Ruffo, Maria-Jose Ruiz, and Guy Verbist. In addition, we thank Klaas Esselink and Sami Karaborni for the cover figure. We thank them all for their efforts. But we stress that the responsibility for errors in the text is ours alone.