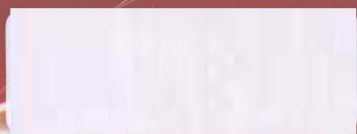


牛津大学研究生教材

统计物理学中的 蒙特卡罗方法

M·E·J·纽曼 G·T·巴克马 著



世界图书出版公司

www.wpcbj.com.cn

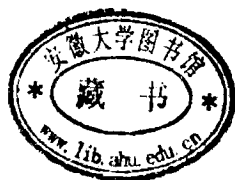
统计物理学中的 蒙特卡罗方法

M · E · J · 纽曼

Santa Fe Institute

G · T · 巴克马

*Institute for Theoretical Physics
Utrecht University*



图书在版编目 (CIP) 数据

统计物理学中的蒙特卡罗方法 = Monte Carlo Methods in Statistical Physics: 英文/(荷) 纽曼著. —影印本. —北京: 世界图书出版公司北京公司, 2012. 3
ISBN 978 - 7 - 5100 - 4417 - 5

I. ①统… II. ①纽… III. ①蒙特卡罗法—应用—统计物理学—英文
IV. ①O414. 2

中国版本图书馆 CIP 数据核字 (2012) 第 030231 号

Copyright © 1999 by Oxford University Press, Inc.

THIS BOOK IS BASED ON MONTE CARLO METHODS IN STATISTICAL PHYSICS. THIS SPECIAL CHINESE VERSION is published by arrangement with Oxford University Press for sale/distribution in The Mainland (part) of the People's Republic of China (excluding the territories of Hong Kong SAR, Macau SAR and Taiwan Province) only and not for export therefrom.

本书得到牛津大学出版社的授权在中国大陆地区 (不包括香港, 澳门和台湾) 重印发行, 不得出口。

书 名: Monte Carlo Methods in Statistical Physics

作 者: M. E. J. Newman, G. T. Barkema

中 译 名: 统计物理学中的蒙特卡罗方法

责任编辑: 高蓉 刘慧

出 版 者: 世界图书出版公司北京公司

印 刷 者: 三河市国英印务有限公司

发 行: 世界图书出版公司北京公司 (北京朝内大街 137 号 100010)

联系电话: 010 - 64021602, 010 - 64015659

电子信箱: kjb@wpcbj.com.cn

开 本: 24 开

印 张: 21

版 次: 2013 年 3 月

版权登记: 图字: 01 - 2011 - 6314

书 号: 978 - 7 - 5100 - 4417 - 5/O · 943

定 价: 59.00 元

影印版前言

本书作为牛津大学出版社研究生教材系列丛书选定的计算物理方面的一部经典教材，主要内容为蒙特卡罗方法在经典统计物理中的应用。它出版于1999年，但至今仍然受到物理类相关学科的师生与研究人员的青睐。现在我们将它推荐给读者正是考虑到，与类似书籍相比较，这本书对于学习相关课程所具有的独特价值。对于本书的详细介绍，包括内容的取舍及其特色、选材遵循的指导原则、与同类图书的比较以及使用的建议等，作者在序言中都有详细的说明，这里不打算赘述。我们只强调指出，该书几乎涵盖了到二十世纪末发展起来的经典统计物理应用领域所涉及的蒙特卡洛模拟所有的重要算法和技巧。作者特别注重各种算法的物理背景的介绍，特别是用了很大篇幅详细地阐述了 Metropolis 算法及其在 Ising 模型中的成功应用范例。作者明确地指出本书强调读者应该透彻地理解各种算法背后的原理和基本概念，而不要拘泥于读懂和编写具体程序。另外本书特别重视如何提高编程效率，对于实际的应用很有指导意义。

蒙特卡洛方法，也称统计模拟方法，随机抽样技术。它是以概率和统计理论方法为基础的一种随机模拟方法，对于很多实际遇到的不能解析处理的复杂问题的近似求解非常有效。通常的做法是将所求解的问题同一定的概率模型相联系，使用随机数（或更常见的伪随机数）用电子计算机实现统计模拟或抽样，或者说用计算机代替人工进行抽样的模拟试验，当试验的次数足够多时可以获得问题的相当精确的近似解。由于这种方法相比其他计算方法编程容易、使用方便、图像清晰、误差估计简单，而且节省了大量的真实实验所消耗的宝贵资源，随着计算机的飞速发展其应用领域扩展极为迅速，受到人们的广泛重视。而今已经成为金融工程学，宏观经济学，计算物理学（如粒子输运计算、量子热力学计算、空气动力学计算以及高能物理理论与实验研究等）以及计算数学等极为重要的工具。

在本书中作者还简略地介绍了蒙特卡洛方法的发展历史，对于读者很有参考价值。作者指出这一方法的前身，“统计抽样”方法发明于十八世纪，远在计算机发明之前。其中最著名的是1777年法国 Georges - Louis Leclerc, Conite de Buffon 所设计的投针试验，通过简单的公式近似计算圆周率 π 。到

1901 年 Mario Lazzarini, 用了一根 2.5 厘米的针投了 3408 次得到了 3.1415929, 这已经是 π 的很好的近似值。随着十九世纪末机械计算机的发明使得利用随机抽样方法第一次用于物理学研究成为现实。但真正促进这一方法大发展的应该归功于两次世界大战。特别是著名的曼哈顿工程进行的原子弹的研制成为极大的推动力。当时刚刚发明的电子计算机使这种随机模拟技术提高到了一个全新的水平。Nick Metropolis, Stanislaw Ulam and John von Neumann 利用新发明的电子计算机在洛斯阿拉莫斯国家实验室为核武器计划完成了关于中子输运的重要工作, 发明了蒙特卡洛方法。蒙特卡洛是摩纳哥的一个赌城的名字, 因为 Ulam 的叔叔经常在那里的赌场输钱, 而蒙特卡洛方法有正是以概率为基础的方法。所以在 1949 年在他们的一篇关于用统计技术研究积分微分方程的文章中第一次以“蒙特卡洛”方法作为标题。从而开辟了一个随机模拟方法应用的新时代。

本书的两位作者是著名的计算物理学家, 他们撰写本书的主要目的是为对蒙特卡罗模拟方法应用于经典统计力学感兴趣的读者提供一部相关课程的教材。但由于涵盖范围广泛, 叙述由浅入深非常细致深入, 注意推导详尽, 很适合有一定统计物理基础的读者用来自学蒙特卡罗方法, 对于想要提高对于新发展起来的那些复杂的模拟技术的有经验研究人员也是一部很有价值的参考书。

中国科学院大学 丁亦兵

前 言

本书所针对的是对蒙特卡罗模拟方法应用于经典统计力学感兴趣的读者。它将适合用做模拟方法，或统计物理课程的教科书。对于希望自学蒙特卡罗方法的读者，以及想要学会近十年左右出现的那些复杂的新模拟技术的有经验研究人员也都将会是良好的选择。

本书主要目的是阐述如何有效地进行蒙特卡罗模拟。对许多人来说，蒙特卡罗模拟仅仅是应用 Metropolis 算法解决手头上的问题。尽管这个著名的算法非常容易编制程序，但以最有效的方式使模拟实现的情况却极为罕见。Metropolis 算法无疑是重要的，我们肯定会详细地讨论它（第三章将做这件事），但是我们将证明：对许多问题，只需做一些用纸笔即可完成的简单工作，就可以得到一个更好的算法，在某些情况下可以快上几千倍或者上百万倍。近年来，在文献中集中涌现了一批有趣的蒙特卡罗新算法，其中，许多都是专门设计用来提高统计物理中特定类别问题的模拟速度的。其中，我们描述了簇算法、多重网格法、序参量守恒模型的非局域算法、熵抽样算法、模拟退火算法和连续时间蒙特卡罗方法。本书被分成了两大部分，分别涵盖了平衡模拟和非平衡模拟，而且自始至终我们都给出了一些指针，指向如何使这些算法最有效地运行。在本书的最后，我们用几章的篇幅，论述了运行蒙特卡罗模拟遇到的一般关键性问题。我们不仅相当详细地阐述了蒙特卡罗模拟中特有的许多技巧，诸如单直方图和多重直方图方法、有限尺度标度和蒙特卡罗重整化群等；还仔细地阐述了了数据分析方法，包括估算可观测量、平衡与相关时间、相关函数、标准误差的一般方法。

本书采用的做法是通过例子讲授。我们尝试尽可能多的包含现今仍在使用的一些重要的蒙特卡罗算法，而且对每一种算法都引用一个或多个具体模型进行说明。例如，Metropolis 算法是通过对伊辛模型的模拟应用说明的。然而，我们并没有认为读者熟悉所研究的这些模型，而是在与之相应的那一章开始处简略概述了每一种模型的物理背景。我们所假定的一切只是读者具备典型的初学物理的大学生水平的统计力学与热力学的基础知识。如果读者想要自己实际编写一个蒙特卡罗程序，那么适当的、熟练的计算机编程语言，诸如 FORTRAN 或 C 语言，肯定是必不可少的，但无此也不影响对本书的理

解。在本书正文中我们已经避免了给出实际的计算机编码实例。采用这种做法有两个原因。不这样做就会要求读者熟知某种特殊语言，或者如果他或她还没有熟悉这种语言就需要学会它；此外，我们并不相信把程序代码包括进来对于理解一种算法会有多大帮助。通过利用所涉及的物理和数学来工作，比试图通过读别人的程序来学习，对获取算法背后原理的清晰概念要好得多。而在清楚地理解那些基本原理之后，你应当不会遇到什么困难就能编写出自己相应的程序。然而，检查别人的程序在一个方面可能是有用的：即它是学习编程技巧和编写高效代码技术的一个好方法。正因如此，在本书结尾的一个附录中，我们包括了一些较为常见的蒙特卡罗算法的程序。这些程序是用 C 语言编写的，若代之以最常用于科学编程的 FORTRAN 语言，会更快。

在本书每一章的末尾我们还包含了许多习题，提供给读者，如果他或她希望的话，去完成它们。这些习题中有一些是纯解析的，可以用书面完成。其它的需要读者编写一个简短的计算机程序。解析类习题的答案在本书末尾给出了。而要求你编写一个计算机程序的那些习题具有许多同样好的解答，因此对于这些习题我们大部分采用给出提示而不是答案。

在编写本书的过程中，很多人和机构给予了热心帮助，在此，我们感谢牛津大学出版社编辑 Sönke Adlung、Donald Degenhardt 和 Julia Thompson 的帮助和耐心，也非常感谢康奈尔大学、牛津大学、普林斯顿高等研究院、于利希研究中心、圣塔菲研究所和乌德勒支大学，在我四年准备书稿过程中所提供的友好款待。最后感谢许多同事和朋友们给我们的建议和鼓励，他们是 James Binney、Geoffrey Chester、Eytan Domany、Peter Grassberger、Harvey Gould、Daniel Kandel、Yongyut Laosiritaworn、Jim Louck、Jon Machta、Nick Metropolis、Cris Moore、Richard Palmer、Gunter Schütz、Jim Sethna、Kan Shen、Alan Sokal 和 Ben Widom。对于任何在书中可能存在的一些没有发现的错误，当然，其责任属于作者。从眼光敏锐的读者所获知的任何这类问题，我们都将深表谢意。

Mark Newman 于美国新墨西哥州圣塔菲

Gerard Barkema 于荷兰乌德勒支

1998 年 6 月

目 录

I 平衡蒙特卡罗模拟

1 引言	3
1.1 统计力学	3
1.2 平衡	7
1.2.1 涨落、相关性和响应	10
1.2.2 例子：伊辛模型	15
1.3 数值方法	18
1.3.1 蒙特卡罗模拟	21
1.4 蒙特卡罗方法的简要历史	22
习题	29
2 热平衡蒙特卡罗模拟的基本原理	31
2.1 估计量	31
2.2 重要抽样	33
2.2.1 马尔科夫过程	34
2.2.2 遍历性	35
2.2.3 细致平衡	36
2.3 接受比	40
2.4 连续时间的蒙特卡罗	42
习题	44
3 伊辛模型与 Metropolis 算法	45
3.1 Metropolis 算法	46
3.1.1 Metropolis 算法的实现	49
3.2 平衡化	53
3.3 测度	57
3.3.1 自相关函数	59

3.3.2	相关时间和马尔科夫矩阵	65
3.4	误差计算	68
3.4.1	统计误差估计	68
3.4.2	阻塞法	69
3.4.3	自助 (Bootstrap) 方法	71
3.4.4	刀切法	72
3.4.5	系统误差	73
3.5	熵的测量	73
3.6	相关函数的测量	74
3.7	一个实际计算	76
3.7.1	相变	82
3.7.2	临界涨落与临界慢化	84
	习题	85
4	伊辛模型的其它算法	87
4.1	临界指数及其测度	87
4.2	Wolff 算法	91
4.2.1	一个聚类算法的接受比	93
4.3	Wolff 算法的性质	96
4.3.1	相关时间和动力学指数	100
4.3.2	动力学指数和磁化率	102
4.4	伊辛模型的高级算法	106
4.4.1	斯文登森 - 王算法	106
4.4.2	Niedermayer 算法	109
4.4.3	多重网格法	112
4.4.4	侵入聚类算法	114
4.5	其它自旋模型	119
4.5.1	Potts 模型	120
4.5.2	Potts 模型的聚类算法	125
4.5.3	连续自旋模型	127
	习题	132
5	序参量守恒的伊辛模型	133
5.1	川崎 (Kawasaki) 算法	138
5.1.1	界面的模拟	140
5.2	更有效的算法	141

5.2.1 一个连续时间的算法	143
5.3 平衡晶体形态	145
习题	150
6 无序自旋模型	151
6.1 玻璃系统	153
6.1.1 随机场伊辛模型	154
6.1.2 自旋玻璃	157
6.2 玻璃系统的模拟	159
6.3 熵抽样方法	161
6.3.1 测量	162
6.3.2 内能与比热	163
6.3.3 熵抽样方法的实现	164
6.3.4 例子: 随机场伊辛模型	166
6.4 模拟回火	169
6.4.1 方法	169
6.4.2 变异	174
习题	177
7 冰模型	179
7.1 冰与冰模型	179
7.1.1 质子的排列	182
7.1.2 冰的残差熵	183
7.1.3 三色模型	186
7.2 正方形冰的蒙特卡罗算法	187
7.2.1 标准冰模型算法	188
7.2.2 遍历性	189
7.2.3 细致平衡	191
7.3 一个替代算法	191
7.4 三色模型的算法	193
7.5 正方形冰的算法比较	196
7.6 含能冰力模型	201
7.6.1 含能冰模型的圈算法	202
7.6.2 含能冰模型的聚类算法	205
习题	209

8 蒙塔卡罗数据分析	210
8.1 单矩形图方法	211
8.1.1 单直方图方法	217
8.1.2 外推到其余变量	218
8.2 多直方图方法	219
8.2.1 实现	226
8.2.2 内插其余变量	228
8.3 有限尺度标度	229
8.3.1 临界指数的直接测量	230
8.3.2 有限尺度标度方法	232
8.3.3 有限尺度标度方法的困难	236
8.4 蒙特卡罗重整化群	240
8.4.1 实空间重整化群	240
8.4.2 临界指数的计算: 指数 ν	246
8.4.3 其它指数的计算	250
8.4.4 指数和	251
8.4.5 更精确的变换	252
8.4.6 指数的测量	256
习题	258

II 偏离平衡模拟

9 偏离平衡蒙特卡罗模拟	263
9.1 动力学	264
9.1.1 选择动力学	266
10 伊辛模型的非平衡模拟	268
10.1 相分离和伊辛模型	268
10.1.1 普通伊辛模型的相分离	271
10.1.2 COP 伊辛模型的相分离	271
10.2 相区尺寸的测量	274
10.2.1 相关函数	274
10.2.2 结构因子	277
10.3 三维伊辛模型的相分离	278
10.3.1 一个更有效的算法	279

10.3.2	连续时间算法	280
10.4	替代动力学	282
10.4.1	体扩散与表面扩散	283
10.4.2	一个体扩散算法	284
习题		288
11	表面科学中的蒙特卡罗模拟	289
11.1	动力学、算法和能垒	292
11.1.1	单吸附原子的动力学	293
11.1.2	多吸附原子的动力学	296
11.2	实现	301
11.2.1	川崎算法和数能带算法	301
11.2.2	查表算法	302
11.3	例子：分子束外延	304
习题		306
12	雷普顿模型	307
12.1	电泳	307
12.2	雷普顿模型	309
12.2.1	投影雷普顿模型	313
12.2.2	模型的参数值	314
12.3	雷普顿模型的蒙特卡罗模拟	315
12.3.1	改进的算法	316
12.3.2	进一步改进的算法	318
12.3.3	雷普顿模型的代表结构	320
12.4	蒙特卡罗模拟结果	322
12.4.1	零电场的模拟	323
12.4.2	非零电场的模拟	323
习题		327
III	实 施	
13	格与数据结构	331
13.1	计算机上的代表格	332
13.1.1	正方格与立方格	332
13.1.2	三角形格、蜂窝格和 Kagomé 格	335

13.1.3	面心立方格、体心立方格和金刚石格	340
13.1.4	一般格	342
13.2	数据结构	343
13.2.1	变量	343
13.2.2	数组	345
13.2.3	链表	345
13.2.4	树	348
13.2.5	缓冲区	352
	习题	355
14	并行计算机上的蒙特卡罗模拟	356
14.1	简单的并行算法	358
14.2	更复杂的并行算法	359
14.2.1	伊辛模型上的 Metropolis 算法	359
14.2.2	伊辛模型上的聚类算法	361
	习题	362
15	多自旋编码	364
15.1	伊辛模型	365
15.1.1	一维伊辛模型	365
15.1.2	二维伊辛模型	367
15.2	多自旋编码算法的实现	369
15.3	真值表和卡诺图	369
15.4	雷普顿模型的一个多自旋编码算法	373
15.5	同步更新算法	379
	习题	380
16	随机数	382
16.1	均匀分布随机数的生成	382
16.1.1	真随机数	384
16.1.2	伪随机数	385
16.1.3	线性同余生成器	386
16.1.4	改进的线性同余生成器	390
16.1.5	移位寄存器发生器	392
16.1.6	延时斐波那契生成器	393
16.2	非均匀分布随机数的生成	396

16.2.1	变换方法	396
16.2.2	高斯分布随机数的生成	399
16.2.3	舍选法	401
16.2.4	混合法	404
16.3	生成随机比特发生器	406
习题		409
参考文献		410
附录		417
A	习题解答	417
B	编程实例	433
B.1	伊辛模型上的算法	433
B.1.1	Metropolis 算法	433
B.1.2	多自旋编码 Metropolis 算法	435
B.1.3	Wolff 算法	437
B.2	COP 伊辛模型上的算法	438
B.2.1	非局域算法	438
B.2.2	连续时间算法	441
B.3	Potts 模型上的算法	445
B.4	冰力模型上的算法	448
B.5	随机数生成器	451
B.5.1	线性同余生成器	451
B.5.2	混合同余生成器	452
B.5.3	延时斐波那契生成器	452
索引		455

Part I

Equilibrium Monte Carlo simulations

1

Introduction

This book is about the use of computers to solve problems in statistical physics. In particular, it is about **Monte Carlo methods**, which form the largest and most important class of numerical methods used for solving statistical physics problems. In this opening chapter of the book we look first at what we mean by statistical physics, giving a brief overview of the discipline we call **statistical mechanics**. Whole books have been written on statistical mechanics, and our synopsis takes only a few pages, so we must necessarily deal only with the very basics of the subject. We are assuming that these basics are actually already familiar to you, but writing them down here will give us a chance to bring back to mind some of the ideas that are most relevant to the study of Monte Carlo methods. In this chapter we also look at some of the difficulties associated with solving problems in statistical physics using a computer, and outline what Monte Carlo techniques are, and why they are useful. In the last section of the chapter, purely for fun, we give a brief synopsis of the history of computational physics and Monte Carlo methods.

1.1 Statistical mechanics

Statistical mechanics is primarily concerned with the calculation of properties of condensed matter systems. The crucial difficulty associated with these systems is that they are composed of very many parts, typically atoms or molecules. These parts are usually all the same or of a small number of different types and they often obey quite simple equations of motion so that the behaviour of the entire system can be expressed mathematically in a straightforward manner. But the sheer number of equations—just the magnitude of the problem—makes it impossible to solve the mathematics exactly. A standard example is that of a volume of gas in a container. One