

Numerical methods for lattice field theory

Mike Peardon

Trinity College Dublin

August 9, 2007

Numerical methods - references

- Good introduction to the concepts: “Simulation”, Sheldon M. Ross, Academic Press ISBN 0-12-598053-1
- Detail on the theory of Markov chains: “Markov chains, Gibbs fields, Monte Carlo simulations and queues”, Pierre Brémaud, Springer ISBN 0-387-98509-3
- A classic: “The Art of Computer Programming, Volume 2” Donald E. Knuth, Addison-Wesley ISBN 0-201-48541-9.
- And another: “Numerical Recipes: The Art of Scientific Computing (3rd Edition)”, Press, Teukolsky, Vetterling and Flannery, CUP ISBN 0-521-88068-8
- Applications to field theory: “The Monte Carlo method in quantum field theory”, Colin Morningstar arXiv:hep-lat/0702020

An opening comment . . .

Monte Carlo is the worst method for computing non-perturbative properties of quantum field theories - except for all other methods that have been tried from time to time.



Churchill said:

"Democracy is the worst form of government, except for all those other forms that have been tried from time to time" (in a speech to the House of Commons after losing the 1947 general election).

Probability (1)

Outcomes, Events and Sample Spaces

Assume we have a precise description of the *outcome*, ω of the observation of a random process. The set of all possible outcomes is called the *sample space*, Ω . Any subset $A \subset \Omega$ can be regarded as representing an *event* occurring.

- The notation of *set theory* is helpful.
- A particularly useful collection of events, A_1, A_2, \dots, A_n are those that are *mutually incompatible and exhaustive*, which implies

$$A_i \cap A_j = \emptyset \text{ when } i \neq j$$

and

$$\bigcup_{k=1}^N A_k = \Omega$$

Probability (2)

- A *random variable* is a function $X : \Omega \rightarrow \mathbb{R}$ such that for $a \in \mathbb{R}$, the event $\{\omega, X(\omega) \leq a\}$ can be assigned a probability.
- A *discrete random variable* is a random variable for which $X : \Omega \rightarrow E$, where E is a denumerable set.

Axioms of probability

A probability (measure) on $(\mathcal{F}, \Omega)^a$ is a mapping $P : \mathcal{F} \rightarrow \mathbb{R}$ such that

- 1 $0 \leq P(A) \leq 1$
- 2 $P(\Omega) = 1$
- 3 $P(\sum_{k=1}^N A_k) = \sum_{k=1}^N P(A_k)$ if A_k are mutually incompatible.

^a \mathcal{F} is the collection of events that are assigned a probability; for obscure reasons it is not the same as Ω

Probability (3)

- Two events are called *independent* if

$$P(A \cap B) = P(A)P(B)$$

- Similarly, two random variables, X and Y are called independent if for all $a, b \in \mathbb{R}$,

$$P(X \leq a, Y \leq b) = P(X \leq a)P(Y \leq b)$$

- A *conditional probability* of event A given B is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- The *cumulative distribution function*, F_X describes the properties of a random variable X and is

$$F_X(x) = P(X \leq x)$$

- The *probability density function*, f_X is defined when F_X can be written

$$F_X(x) = \int_{-\infty}^x f_X(z) dz$$

Statistics (1)

Expectation

For an absolutely continuous c.d.f, and a function g , the *expectation*, $E[g(X)]$ is defined as

$$E[g(X)] = \int_{-\infty}^{\infty} g(z) f_X(z) dz$$

Mean and variance

The *mean*, μ_X of X is

$$\mu_X = E[X] = \int_{-\infty}^{\infty} z f_X(z) dz$$

and the *variance*, σ_X^2 is

$$\sigma_X^2 = E[(X - \mu_X)^2] = E[X^2] - \mu_X^2$$

Statistics (2)

The strong law of large numbers (Kolmogorov)

If $\bar{X}^{(n)}$ is the sample mean of n independent, identically distributed random numbers, $\{X_1, X_2, \dots, X_n\}$ with well-defined expected value μ_X , and variance σ_X then $\bar{X}^{(n)}$ converges almost surely to μ_X ;

$$P\left(\lim_{n \rightarrow \infty} \bar{X}^{(n)} = \mu_X\right) = 1$$

- Bernoulli: "...so simple that even the stupidest man instinctively knows it is true"

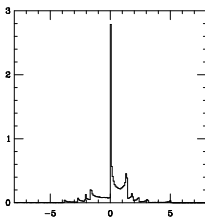
The central limit theorem (de Moivre, Laplace, Lyapunov, ...)

$$\lim_{n \rightarrow \infty} P\left(\frac{\bar{X}^{(n)} - \mu_X}{\sigma_X / \sqrt{n}} \leq z\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-y^2/2} dy$$

As more statistics are gathered, all sample means become normally distributed.

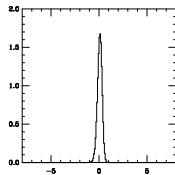
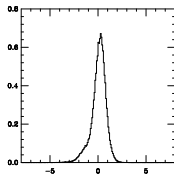
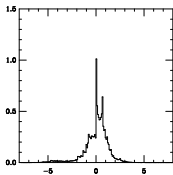
The central limit theorem

X

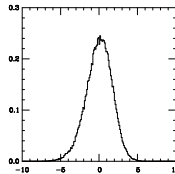
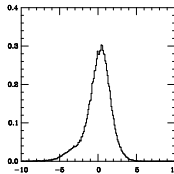
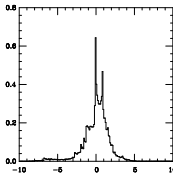


Prob. densities

$\bar{X}(n)$



$\sqrt{n}\bar{X}(n)$



$n = 2$

$n = 5$

$n = 50$

Generating (pseudo-)random numbers on a computer

- A deterministic machine like a computer is incapable of generating really random numbers.
- Randomness is faked; algorithms can be devised that generate sequences of numbers that *appear* random.
- To appear random implies statistical identities hold for large sets of these numbers. Testing randomness is done by both looking at statistical correlations (for example checking that using d adjacent entries in a sequence as co-ordinates of a d -dimensional point does not result in points that lie on certain planes in preference) and using a simulation to compute something known analytically. The best-known is Marsaglia's *Diehard* test.
- The elements of these sequences that have passed all the appropriate statistical tests are called *pseudo-random*.
- The original example is the *linear congruential sequence*. For details see eg. Knuth.

Generating (pseudo-)random numbers on a computer (2)

- Suppose we generate a sequence by applying some deterministic algorithm \mathcal{A}

$$X_0 \xrightarrow{\mathcal{A}} X_1 \xrightarrow{\mathcal{A}} X_2, \dots$$

- If any entry in the sequence can take any of m values, then the sequence must repeat itself in *at most* m iterations.
- The number of iterations before a sequence repeats itself is called the *cycle*, which may depend on the initial state, X_0 .
- This repetition of the sequence over-and-over again is definitely not random! A useful generator must thus have
 - ▶ m large - at least as large as the number of random numbers needed in the computer program, preferably much more.
 - ▶ The maximum possible cycle, independent of starting point, X_0 .

Generating (pseudo-)random numbers on a computer (3)

The best-known algorithm is:

The linear congruential sequence

The sequence, X_0, X_1, X_2, \dots with $0 \leq X_i < m$ is generated by the recursion

$$X_{n+1} = aX_n + c \pmod{m}$$

with

- X_0 the seed
- a the multiplier $0 \leq a < m$
- c the increment $0 \leq c < m$
- m the modulus $m > 0$

The period can be maximised by ensuring

- c is relatively prime to m ,
- $b = a - 1$ is a multiple of p for every prime p dividing m and
- b is a multiple of 4 if m is a multiple of 4.

See Knuth for details.

An example linear congruential sequence

- To demonstrate this in action, let's pick parameters according to the prescription above
- $m = 90 = 5 \times 3 \times 3 \times 2$ then $c = 13, b = 5 \times 3 \times 2 \rightarrow a = 31$ and seed with $X_0 = 0$
- The sequence is below. No number appears more than once.

0,	13,	56,	39,	52,	5,	78,	1,	44,	27,	40,	83,
66,	79,	32,	15,	28,	71,	54,	67,	20,	3,	16,	59,
42,	55,	8,	81,	4,	47,	30,	43,	86,	69,	82,	35,
18,	31,	74,	57,	70,	23,	6,	19,	62,	45,	58,	11,
84,	7,	50,	33,	46,	89,	72,	85,	38,	21,	34,	77,
60,	73,	26,	9,	22,	65,	48,	61,	14,	87,	10,	53,
36,	49,	2,	75,	88,	41,	24,	37,	80,	63,	76,	29,
12,	25,	68,	51,	64,	17						

Generating (pseudo-)random numbers on a computer (4)

- Many more algorithms exist now, and are used in real lattice QCD codes:
- Lüscher's `ranlux`

This generator, based on Marsaglia and Zaman's RCARRY algorithm has a long period (10^{171}) when set up properly. The quality is established by considering the dynamics of chaotic systems.
(M. Lüscher, "A portable high-quality random number generator for lattice field theory simulations", *Comput. Phys. Commun* **79** (1994), 100-110. [hep-lat/9309020](#))
- Matsumoto and Nishimura's `mersenne twister`

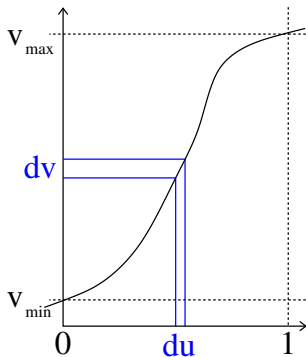
The period length is a Mersenne prime, and the most commonly-known variant has a period of $2^{19937} - 1 \approx 4 \times 10^{6001}$.
(M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", *ACM Trans. on Modeling and Computer Simulation* Vol. 8, No. 1, January (1998) 3-30)

Generating more general distributions

- We will find it useful soon to generate random numbers with more general distributions.
- Consider taking a uniform variate U and applying some function v (defined in $[0, 1]$; let us also assume it is increasing in that region). We get a new random variable, $V = v(U)$.
- What is the probability distribution of V ? A simple analysis, shown in the figure (right) shows that equating the probabilities

gives

$$f_V(v) = \frac{du}{dv} = \left(\frac{dv}{du} \right)^{-1}$$



The transformation method

- This observation suggest an algorithm, provided the function v can be found.
- v is the solution to

$$u = \int_{-\infty}^{v(u)} f_X(z) dz$$

unfortunately, there are few pdfs for which this method is efficient!

- Perhaps the most important example for QCD: the Box-Muller algorithm, which generates pairs of normally distributed random numbers.

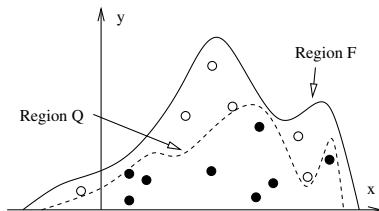
The Box-Muller algorithm

- 1 Generate two independent u.v. random numbers, $U_{1,2} \in (0, 1]$
- 2 Set $\Theta = 2\pi U_1$ and $R = \sqrt{-2 \ln U_2}$
- 3 Return $R \cos \Theta$ or $R \sin \Theta$ (or both; they are independent).

The rejection method

- Consider the following experiment, for regions F and $Q \subset F$:

- 1 Generate a random point, P with co-ordinates (X, Y) inside F
- 2 If P is also inside Q , return X .
- 3 Otherwise go back to step 1.



- What is the p.d.f. for X ? Answer: $q(x) / \int q(z) dz$ where q bounds the region Q . (There is no need to know the normalisation of q).

The rejection method

- 1 Generate X from p.d.f. f
- 2 Generate $Y = c f(X)U$ where U is a uniform variate and c is a constant such that $f(x) > q(x)$.
- 3 If $Y < q(X)$, return X otherwise go to step 1.

Monte Carlo integration

- Consider X , a random point in V with probability density function $p_X(x)$. A new stochastic variable F is defined by applying a function f to X , so $F = f(X)$.
- The expected value of F is then

$$E(F) = \int_V f(x)p_X(x)dx$$

- If N independent instances of X are generated, and then N instances of F are derived, the mean of this ensemble is

$$\langle F \rangle_N = \frac{1}{N} \sum_{i=1}^N F_i$$

The basic result of Monte Carlo

$$\lim_{N \rightarrow \infty} \langle F \rangle_N = E(F) = \int_V f(x)p_X(x)dx$$

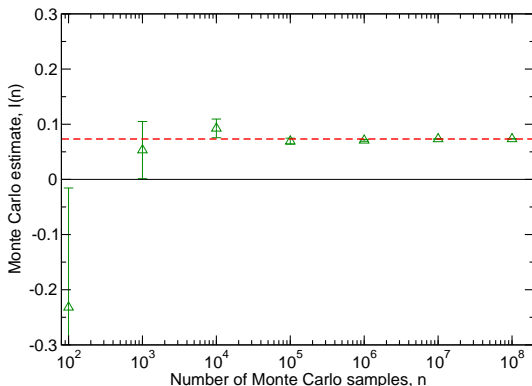
Monte Carlo integration - an example

- Estimate

$$I = \int_0^1 \frac{\sin 10\pi(1-x)}{\sqrt{x}} dx$$

using Monte Carlo.

- Generate an ensemble of uniform variate points, $\{U_1, U_2, \dots\}$.
- Then evaluate $F_i = f(U_i)$ and compute the mean of this ensemble.



Importance Sampling

- For almost all the physics problems we want to solve, we find just a tiny fraction of configuration space contributes to the integral; variance reduction will be crucial.
- The most useful variance reduction technique is *importance sampling*
- Instead of a flat sampling over the integration volume, the regions where the integral dominates are sampled preferentially
- For any $p(x) > 0$ in V , we have

$$I = \int_V f(z) Dz = \int_V \frac{f(z)}{p(z)} p(z) Dz$$

- Suppose we generate an ensemble of points, $\{X_1, X_2, \dots\}$ in V with p.d.f. p and compute the random variable

$$J_i = \frac{f(X_i)}{p(X_i)}$$

Importance Sampling

- Then (as seen above), $E[J] = I$, so the sample mean converges to the answer, independently of p but ...
- ... the variance is

$$\sigma_J^2 = E[J^2] - E[J]^2 = \int_V \frac{f(z)^2}{p(z)^2} p(z) Dz - I^2 = \int_V \frac{f(z)^2}{p(z)} Dz - I^2$$

and so this **does** depend on the choice of p .

- The optimal choice of p would minimise σ_J^2 , so a functional variation calculation yields the optimal choice to be

$$p(x) = \frac{|f(x)|}{\int_V |f(z)| Dz}$$

- In practice, it is usually difficult to draw from this sampling (since if we could, it is likely we could compute I directly!), but this suggests p should be peaked where $|f|$ is peaked.
- For QFTs, importance sampling is just a synonym for using the Boltzmann weight in the Euclidean path integral (doesn't have to be!)

Importance Sampling (3)

- Another example. Using Monte Carlo, estimate

$$I(a) = 2 \int_0^a \int_0^a e^{-(x^2+y^2)/2} dx dy$$

Note that

$$\lim_{a \rightarrow \infty} I(a) = \pi$$

Simple Monte Carlo

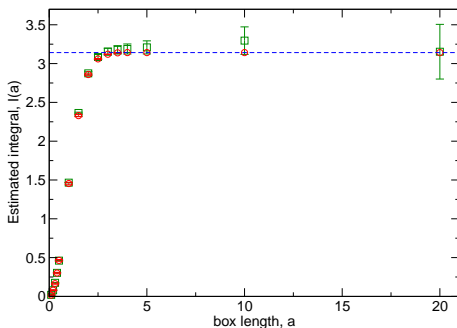
- 1 Generate independent uniform variates $X, Y \in [0, a]$
- 2 Compute $F = e^{-(X^2+Y^2)/2}$
- 3 Repeat N times, and average result, $\bar{F}^{(N)}$ return $2a^2 \bar{F}^{(N)}$

Importance sampling

- 1 Generate normally distributed $X, Y \in [0, \infty]$
- 2 $G = 1$ if $X \leq a, Y \leq a$.
- 3 Repeat N times, and average result, $\bar{G}^{(N)}$ return $\pi \bar{G}^{(N)}$

- As $a \rightarrow \infty$, the region contributing almost all the integral (near the origin) becomes fractionally smaller and smaller - variance in simple MC estimate will diverge.

Importance Sampling (4)



- The variance in the importance sampling estimator becomes zero as $a \rightarrow \infty$
- Unfortunately, so far we have only developed algorithms to generate random numbers with a general probability distribution function in very low dimensions - for field theory applications, we want to calculate integrals in many thousands of dimensions...

Markov Processes

- A more general method for generating points in configuration space is provided by considering Markov processes.
- In 1906, Markov was interested in demonstrating that independence was not necessary to prove the (weak) law of large numbers.
- He analysed the alternating patterns of vowels and consonants in Pushkin's novel "Eugene Onegin".
- In a Markov process, a system makes stochastic transitions such that the probability of a transition occurring depends only on the start and end states. The system retains no memory of how it came to be in the current state. The resulting sequence of states of the system is called a Markov chain.



A. A. Markov (1856).

Markov Processes (2)

A Markov Chain

Let $\{\psi_i\}$ for $i = 0..n+1$ be a sequence of states generated from a finite state space Ω by a stochastic process. Let $\chi_k \in \Omega$ be a state, such that $\chi_i \cup \chi_j = \emptyset$ if $i \neq j$ and $\Omega = \chi_1 \cup \chi_2 \cup \chi_3 \cup \dots \chi_m$. If the conditional probability obeys

$$P(\psi_{n+1} = \chi_i | \psi_n = \chi_j, \psi_{n-1} = \chi_{j_{n-1}}, \dots, \psi_0 = \chi_{j_0}) = P(\psi_{n+1} = \chi_i | \psi_n = \chi_j)$$

then the sequence is called a Markov Chain

- Moreover, if $P(\psi_{n+1} = \chi_i | \psi_n = \chi_j)$ is independent of n , the sequence is a *homogenous Markov chain*.
- From now on, most of the Markov chains we will consider will be homogenous, so we'll drop the label.

Markov Processes (3)

- The transition probabilities fully describe the system. They can be written in matrix form (usually called the Markov matrix);

$$M_{ij} = P(\psi_{n+1} = \chi_i | \psi_n = \chi_j)$$

- The probability the system is in a given state after one application of the process is then

$$P(\psi_{n+1} = \chi_i) = \sum_{j=1}^m P(\psi_{n+1} = \chi_i | \psi_n = \chi_j) P(\psi_n = \chi_j)$$

- Writing the probabilistic state of the system as a vector, application of the process looks like linear algebra

$$p_i(n+1) = P(\psi_{n+1} = \chi_i) = M_{ij} p_j(n)$$

Markov Processes (4)



- An example: Seattle's weather.
- It is noticed by a resident that on a rainy day in Seattle, the probability tomorrow is rainy is 80%. Similarly, on a sunny day the probability tomorrow is sunny is 40%.
- This suggests Seattle's weather can be described by a (homogenous) Markov process. From this data, can we compute the probability any given day is sunny or rainy?
- For this system, the Markov matrix is

	Sunny	Rainy
Sunny	0.4	0.2
Rainy	0.6	0.8

Markov Processes (5)

- If today is sunny, then $\psi_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, the state vector for tomorrow is then $\psi_1 = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}$, and $\psi_2 = \begin{pmatrix} 0.28 \\ 0.72 \end{pmatrix}$, $\psi_3 = \begin{pmatrix} 0.256 \\ 0.744 \end{pmatrix}$, \dots
- If today is rainy, then $\psi_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, the state vector for tomorrow is then $\psi_1 = \begin{pmatrix} 0.2 \\ 0.8 \end{pmatrix}$, and $\psi_2 = \begin{pmatrix} 0.24 \\ 0.76 \end{pmatrix}$, $\psi_3 = \begin{pmatrix} 0.248 \\ 0.752 \end{pmatrix}$,
- The vector ψ quickly collapses to a fixed-point, which must be π , the eigenvector of M with eigenvalue 1, normalised such that $\sum_{i=1}^2 \pi_i = 1$.
- We find $\pi = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix}$. This is the invariant probability distribution of the process; with no prior information these are the probabilities any given day is sunny (25%) or rainy (75%).

The Markov Matrix (1)

- The Markov matrix has some elementary properties

- 1 Since all elements are probabilities,

$$0 \leq M_{ij} \leq 1$$

- 2 Since the system always ends in Ω ,

$$\sum_{i=1}^N M_{ij} = 1$$

- From these properties alone, **the eigenvalues of M must be in the unit disk**; $|\lambda| \leq 1$, since if v is an eigenvector,

$$\sum_j M_{ij} v_j = \lambda v_i \implies \left| \sum_j M_{ij} v_j \right| = |\lambda| |v_i| \implies \sum_j M_{ij} |v_j| \geq |\lambda| |v_i|$$

$$\sum_j \left(|v_j| \sum_i M_{ij} \right) \geq |\lambda| \sum_i |v_i| \implies 1 \geq |\lambda|$$

The Markov Matrix (2)

- Also, a **Markov matrix must have at least one eigenvalue equal to unity**. Considering the vector $v_i = 1, \forall i$, we see

$$\sum_i v_i M_{ij} = \sum_i M_{ij} = 1, \quad \forall j$$

and thus v is a left-eigenvector, with eigenvalue 1.

- Similarly, for the right-eigenvectors,

$$\sum_j M_{ij} v_j = \lambda v_i \implies \sum_j v_j \sum_i M_{ij} = \lambda \sum_i v_i \implies \sum_j v_j = \lambda \sum_i v_i$$

and so either $\lambda = 1$ or if $\lambda \neq 1$ then $\sum_i v_i = 0$