I. Squaring the circle  
   A. Problems

II. Generating functions  
   A. Problems

III. Characteristic functions  
   A. Problems

IV. Metropolis-Hastings algorithm  
   A. Finite Markov chains  
      1. Problems
   B. Finite Markov chains (continued)
   C. Detailed balance
   D. The algorithm

V. One-dimensional toy model  
   A. The transfer matrix
   B. Problems

VI. Appendix

I. Squaring the circle

Define the following random/stochastic variable

\[ P = (X^2 + Y^2 < 1), \] (1)
defined in terms of random variables $x$ and $y$.

Define

$$\Theta(P) = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{if } P \text{ is false} \end{cases} \quad (2)$$

To obtain an estimate of $\pi$ we use

$$\langle \Theta(P) \rangle = \frac{\pi}{4} \quad (3)$$

The Monte Carlo estimator to be used is

$$\langle \Theta(P) \rangle \approx \frac{1}{N} \sum_{i=1}^{N} \Theta(P(X_i, Y_i)) \equiv \Theta(P(X, Y)) \quad (4)$$

with the $X_i$ and $Y_i$ independent and $U(-1, 1)$ distributed. In general, $U(a, b)$ denotes the probability density function uniform on the interval $(a, b)$.

Comment about the notation: $\langle X \rangle$ denotes the expectation value of a stochastic variable. If $X$ has a probability density function $\rho$, i.e., $Pr(x < X \leq x + dx) = \rho(x)dx$, where $Pr(E)$ denotes the probability of the occurrence of event $E$,

$$\langle X \rangle = \int x\rho(x) \, dx. \quad (5)$$

For a random variable assuming discrete values $x_1, x_2, \ldots$ and $Pr(X = x_i) = \rho_i$

$$\langle X \rangle = \sum_i x_i\rho_i. \quad (6)$$

$\overline{X}$ denotes for the average of $N$ instances $X_i$ of $X$:

$$\overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i, \quad (7)$$

A. Problems

1. Show that

$$\text{var}X = \langle X^2 \rangle - \langle X \rangle^2 \quad (8)$$

2. Show that

$$\text{var}(\alpha X) = \alpha^2 \text{var}X \quad (9)$$
3. (a) Is

\[ \langle X + Y \rangle = \langle X \rangle + \langle Y \rangle \]  

always valid or only if \( X \) and \( Y \) are independent?

(b) Suppose that \( X \) and \( Y \) are independent random variables, \( i.e., \langle XY \rangle = \langle X \rangle \langle Y \rangle \).

Show that

\[ \text{var}(X + Y) = \text{var}X + \text{var}Y \]  

(11)

4. Suppose \( \langle X \rangle = \mu \) and \( \text{var}X \equiv \langle (X - \mu)^2 \rangle = \sigma^2 \). Show that

(a) \[ \langle \overline{X} \rangle = \mu \]  

(12)

(b) \[ \langle (X - \mu)^2 \rangle = \sigma^2 \]  

(13)

(c) \[ \langle (X - \overline{X})^2 \rangle = \frac{N - 1}{N} \sigma^2 \]  

(14)

Eqs. (12) and (13) show that the left-hand sides of these equations are unbiased estimators of \( \mu \) and \( \sigma^2 \). Eq. (14) shows that its left-hand side is biased and underestimates the variance \( \sigma^2 \). Try to understand why the use of the fluctuating \( \overline{X} \) instead of constant \( \mu \) on the left-hand side tend to do this.

5. (a) According to the central limit theorem, the uncertainty of the estimator in Eq. (7) is \( \sigma/\sqrt{N} \). Use Eqs. (14) and (8) to obtain an estimate of the uncertainty of \( \pi \) obtained from the Monte Carlo method based on Eq. (4).

(b) Repeat the Monte Carlo-\( \pi \) experiment 10 to 50 times and use the “standard physics lab-recipe” to compute the uncertainty of your estimate.

(c) Consider a sequence of \( n \) independent experiments (Bernoulli trials) each resulting in 1 with probability \( p \) and 0 with probability \( q = 1 - p \). The probability of obtaining 1 precisely \( k \) times is given by the binomial distribution

\[ B(k; n, p) = \binom{n}{k} p^k q^{(n-k)} \]  

(15)

This distribution has variance \( npq \). This result to estimate the uncertainty of a Monte Carlo-\( \pi \) experiment consisting of \( N \) Monte Carlo steps.

Suggestion: look at the Mathematica documentation for BinomialDistribution.
II. GENERATING FUNCTIONS

Consider discrete random variable $X$ assuming values $0, 1, \ldots$ with probability $\rho_i$. Define the so-called generating function $g$ defined by

$$g(s) = \sum_k \rho_k s^k$$

Clearly,

$$\left( \frac{dg}{ds} \right)_{s=1} = \sum_k k \rho_k = \langle X \rangle,$$

and

$$\left( \frac{d^2 g}{ds^2} \right)_{s=1} = \sum_k k(k-1) \rho_k = \langle X^2 \rangle - \langle X \rangle^2.$$  \hfill (18)

A. Problems

1. Use Eqs. (17) and (18) to prove that if $X$ has probability distribution $B(k; n, p)$

$$\langle X \rangle = np$$

$$\text{var}X = np(1-p)$$

2. Consider independent random variables $X_i, i = 1, \ldots, N$. Assume

$$X_i = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } q = 1-p \end{cases}$$

Use Eqs. (10) and (11) to derive Eqs. (19) and (20).

III. CHARACTERISTIC FUNCTIONS

The characteristic function of a continuous random variable $X$ with probability density function $\rho$ is defined by

$$\chi(k) = \langle e^{ikX} \rangle = \int_{-\infty}^{\infty} e^{ikx} \rho(x) \, dx$$

The probability density function can be recovered from the characteristic function by using the Dirac $\delta$-function

$$\delta(x - x') = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ik(x-x')} \, dk,$$
which has the property that

$$f(x) = \int_{-\infty}^{\infty} \delta(x - x') f(x') \, dx'$$  \hfill (24)

Multiply Eq. (22) by $\exp(-ikx)$ and integrate over $k$ and use Eq. (23); this gives

$$\rho(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} \chi(k) \, dk.$$  \hfill (25)

If $X_i$ has characteristic function $\chi_{X_i}$ ($i = 1, 2$) and if the $X_1$ and $X_2$ are independent then the characteristic function of $X_1 + X_2$ is $\chi_1 \chi_2$. This follows immediately from

$$\langle AB \rangle = \langle A \rangle \langle B \rangle$$  \hfill (26)

for any two independent random variables.

### A. Problems

1. Consider independent, $U(-1, 1)$ distributed random variables $X_i$. Make histograms for $\frac{1}{n} \sum_{i=1}^{n} X_i$. You can use the following Mathematica instructions for that

```mathematica
Manipulate[
    Histogram[Sum[RandomReal[{-1, 1}, 10000], {n}] / n, {0.02}],
    {n, Range[1, 6]}
]
```

2. (a) Find $\chi_1(k)$ the characteristic function of a $U(-1, 1)$ distributed random variable.

(b) In terms of $\chi_1$ write the characteristic function of

$$S_2 = (X_1 + X_2)/\sqrt{2}$$  \hfill (27)

with $X_i$ independent, $U(-1, 1)$ distributed random variable.

(c) Use the result of (2b) to calculate the probability density function of $X$.

(d) Similarly, find $\chi_n$ the characteristic function of

$$S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i$$  \hfill (28)

(e) Numerically perform the integration required to obtain from $\chi_n$ the probability density function of $S_n$. You can use `NIntegrate` for this purpose. Plot the probability density functions of $S_n$ for various $n$ in the same graph.
IV. METROPOLIS-HASTINGS ALGORITHM

A. Finite Markov chains

A sequence of trials with \( n \) possible outcomes \( E_1, \ldots, E_n \) is called finite stationary Markov chain if the probability of any sequence \((E_{j_1}, \ldots, E_{j_t})\) satisfies

\[
P_r((E_{j_1}, \ldots, E_{j_t})) = P_{j_{t-1}j_t} P_r((E_{j_1}, \ldots, E_{j_{t-1}})),
\]

\[
P_r((E_{j_t})) = \rho_{j_t}.
\]

Think of the indices \( t = 1, \ldots, T \) as labeling discrete time. The important thing to notice is that the first factor on the right of Eq. (29) depends only on the events \( E_{j_t} \) and \( E_{j_{t-1}} \) and not on the previous events at times \( t = 1, \ldots, t-2 \). Using Eq. (29) repeatedly gives

\[
P_r((E_{j_1}, \ldots, E_{j_t})) = P_{j_{t-1}j_t} P_{j_{t-2}j_{t-1}} \cdots P_{j_2j_1} \rho_{j_1}.
\]

The stationary distribution is given by the probability \( \rho_i \geq 0 \) and the \( P_{ij} \geq 0 \) are transition probabilities; these probabilities satisfy

\[
\sum_{i=1}^{n} \rho_i = 1 \tag{32}
\]

\[
\sum_{i=1}^{n} P_{ij} = 1 \tag{33}
\]

for all \( j = 1, \ldots, n \). Note that in the event sequences time increases from left to right, while in the transition probabilities the reverse is true. The latter is the convention commonly used in physics, but not in mathematics.

We can also consider non-stationary Markov chains. In that case, if \( \rho_i^{(t)} = \Pr(E_i \text{ at time } t) \), one has for time \( t + 1 \)

\[
\rho_i^{(t+1)} = \sum_{j=1}^{n} P_{ij} \rho_j^{(t)} \tag{34}
\]

In general, a matrix \( P = (P_{ij})_{i,j=1}^{n} \) with \( P_{ij} \geq 0 \) that satisfies Eq. (33) is called a Markov or stochastic matrix.

1. Problems

1. Consider an two-state Markov process with states 1 and 2. Use RandomReal[] to generate a 2 \( \times \) 2 a random Markov matrix. Make sure to multiply each column of the
random matrix so that Eq. (32) is satisfied. Start the system in state 1 at time $t = 1$ and perform 10 random steps as defined by the Markov matrix. Repeat this process many times from $t = 1$. Verify that the frequencies with which states 1 and 2 occur are given by Eq. (34) with $\rho_i^{(1)} = \delta_{i1}$.

**B. Finite Markov chains (continued)**

A square matrix is called irreducible if it cannot be brought into upper- or lower- triangular block form by a permutation. The Perron-Frobenius theorem for an irreducible Markov matrix states that the dominant eigenvalue, the one largest in magnitude, is non-degenerate and has left and right eigenvectors with non-vanishing elements all of which have the same sign.

Eq. (33), shows that a Markov matrix has dominant eigenvalue 1 and a corresponding eigenvector with elements that can be chosen to all equal unity. Because of the Perron-Frobenius theorem, a irreducible Markov matrix has a unique equilibrium (or stationary or invariant) state which satisfies

$$\rho_i^{(\infty)} = \sum_{j=1}^{n} P_{ij} \rho_j^{(\infty)}$$  \hspace{1cm} (35)

You can derive this assuming that $P$ is not defective, i.e. has a complete set of eigenvectors, and expanding any initial state in eigenstates. This immediately shows that for long times the spectral weights of all but the dominant state become exponentially small. Check for yourself that the Perron-Frobenius theorem implies that the spectral weight of the dominant eigenvector cannot vanish.

**C. Detailed balance**

For most of the Monte Carlo methods we shall consider, the equilibrium state $\rho$ is given as the distribution to be sampled and a Markov matrix is to be constructed with this distribution as its equilibrium state. We can take Eq. 33, the condition that the probability is unity to go from any state to some other state, and re-write it as

$$\rho_i = \sum_{j=1}^{n} P_{ji} \rho_i.$$  \hspace{1cm} (36)
The equilibrium condition reads
\[ \rho_i = \sum_{j=1}^{n} P_{ij} \rho_j. \] (37)

Subtract Eqs. (36) and (37) to obtain
\[ \sum_{j=1}^{n} (P_{ij} \rho_j - P_{ji} \rho_i) = 0. \] (38)

Sufficient to satisfy Eq. (38) is
\[ P_{ij} \rho_j = P_{ji} \rho_i, \] (39)
the so-called detailed balance condition, because this condition guarantees that the probability flux from state \( i \) to \( j \) equals the reverse flux from \( j \) to \( i \). Of course this is not a necessary condition: equilibrium can also be maintained by the a process in which state \( i \) feeds into state \( j \), which feeds into state \( k \) which finally feeds back to state \( i \). Clearly, a process in which equilibrium is maintained as in this last example is not time-reversal symmetric, and we shall now show that detailed balance follows from time-reversal symmetry.

As we saw \( P_{ij} = \Pr(E_j \text{ is followed by } E_i) \). What is \( \tilde{P}_{ij} = \Pr(E_j \text{ is preceded by } E_i) \)?

Clearly,
\[ P_{ij} = \frac{\Pr((E_j,E_i))}{\Pr(E_j)}. \] (40)

Similarly,
\[ \tilde{P}_{ij} = \frac{\Pr((E_i,E_j))}{\Pr(E_j)}. \] (41)

Reverse \( i \) and \( j \) in this last equation and calculate the ratio
\[ \frac{\tilde{P}_{ji}}{P_{ij}} = \frac{\Pr(E_j)}{\Pr(E_i)} = \frac{\rho_j}{\rho_i}, \] (42)
that is
\[ P_{ij} \rho_j = \tilde{P}_{ji} \rho_i, \] (43)

Clearly, this last equation becomes the condition of detailed balance if the Markov process is time-reversal invariant, \( i.e., \frac{\tilde{P}_{ji}}{P_{ji}} = 1 \).

**D. The algorithm**

To construct a Markov matrix that satisfies detailed balance, Eq. (39), write
\[ P_{ij} = A_{ij} T_{ij} \] (44)
Algorithm 1 Metropolis-Hastings algorithm

Start from $E_j$ at time $t$.

1. Propose $E_i$ with probability $T_{ij}$.

2. Accept $E_i$ as new state at time $t + 1$ with probability $A_{ij}$, where

$$A_{ij} = \min\left(\frac{T_{ji}\rho_i}{T_{ij}\rho_j}, 1\right).$$  \hspace{1cm} (45)

3. If $E_i$ is rejected in step 2, the old $E_j$ becomes the new $E_i$ at $t + 1$: $E_i = E_j$.

and use the steps of Algorithm 1 to generate the state $E_j$ at time $t + 1$ from $E_i$ at time $t$.

Verify that Eq. (45) guarantees that the condition of detailed balance is indeed satisfied. In fact, if $f$ is any function that satisfies

$$f(x) = xf(1/x) \text{ and } 0 \leq f(x) \leq 1$$ \hspace{1cm} (46)

this will work with the choice

$$A_{ij} = f\left(\frac{T_{ji}\rho_i}{T_{ij}\rho_j}\right).$$ \hspace{1cm} (47)

It is easy to construct many rational functions $f$ that satisfy (46). In particular, the function $f(x) = x/(1 + x)$ satisfies the requirements, and indeed it has been used, but it clearly produces a lower acceptance than Eq. (45) and therefore is a less efficient choice.

Some final comments are in order. First of all, as is clear from Eq. (47), the probability density function $\rho$ may be a relative probability density function, i.e., known apart from an overall constant. Secondly, in our formulation so far we have used finite Markov chains. In more general cases, the single discrete index $i$ we used above can be replaced by one or several continuous variables that label events. Without further justification we assume that the same equilibrium properties are valid for Markov processes with a discrete or continuous infinity of states. This assumption is not correct in general, i.e., even if every state can be reached from every state, the existence of a unique invariant state and exponential relaxation are not guaranteed, but we blithely ignore such complications.
V. ONE-DIMENSIONAL TOY MODEL

Consider a one-dimensional system defined on a line with sites 1, 2, \ldots, n. Each site has a height variable $h_i$ associated with it. $h_i$ can assume values 0, ±1. The relative probability of finding the system in state $h_1, h_2, \ldots, h_n$ is

$$ P_{h_1, h_2, \ldots, h_n} = \exp[\kappa (\sum v_{h_i} - \sum_{i=1}^{n} |h_{i+1} - h_i|)] \quad (48) $$

with \( \kappa \). $v_0 = 1$ and $v_{±1} = 1$. We identify $h_{n+1} \equiv h_1$, i.e., periodic boundary conditions.

The term relative probability means that the probabilities do not add up to unity. You can think of this system as an interface that fluctuates about zero with steps of ±1. The second term has the tendency of making the interface flat, while the first term imposes a preference for the interface to be at height ±1. As $\kappa$ becomes big, only two equivalent configurations survive: $h_1 = \cdots = h_n = ±1$.

We want to calculate $p(h_1)$, the probability of finding the system in a state with $h_1 = 0, ±1$ while $h_i$ assumes any value for $i > 1$.

Clearly,

$$ p_{h_1} = \frac{\sum h_2 \cdots \sum h_n P_{h_1, h_2, \ldots, h_n}}{\sum_{h_1} \sum h_2 \cdots \sum h_n P_{h_1, h_2, \ldots, h_n}} \quad (49) $$

The denominator is included to make sure that the sum over all probabilities is equal to unity.

A. The transfer matrix

Using the periodic boundary conditions, we can rewrite $P$ in Eq. (48) as follows

$$ P_{h_1, h_2, \ldots, h_n} = T_{h_1 h_2} T_{h_2 h_3} \cdots T_{h_n h_1} \quad (50) $$

where $T$ is called the transfer matrix an is defined by

$$ T_{hh'} = \exp\left(\kappa \frac{v_h + v_{h'}}{2} - \kappa |h - h'|\right). \quad (51) $$

In terms of the transfer matrix $T$ Eq. (49) can be written as

$$ p_{h_1} = \frac{(T^n)_{h_1 h_1}}{\text{Tr} T^n} \quad (52) $$

In other words, to calculate $p_{h_1}$ all one has to do is multiply $T$ by itself $n - 1$ times and take the ratio of one of the diagonal elements of the resulting matrix, and divide this by the sum of the diagonal elements.
B. Problems

1. Compute \( p_{h_1} \) by Monte Carlo. To do this pick a random site \( i \) and propose a new value of the height variable \( h_i \), one of the two different values equal probability. Accept the proposed new value of \( h_i \) using the Metropolis-Hastings algorithm.

2. Study what happens to the estimate of \( p_{h_1} \) and the error estimate as \( \kappa \) increases.

3. What happens if once every so often you do a global move that changes the values of all \( h_i \) to \(-h_i\).

Note: \( p_{h_1} \) is the probability of the height variable of site 1. This quantity can be estimated from a histogram for \( h_1 \). Clearly, all sites are equivalent. The statistics can be improved dramatically by averaging the histogram over all sites.

To be continued

VI. APPENDIX

When I use something of the form \(<a \text{ function}>\), the angular brackets mean that the bracketed construct as a whole should be replaced by a valid object of the sort described in brackets. Similarly, \(<\text{return}>\) and \(<\text{esc}>\) indicate the return and escape keys on your keyboard. I am recycling this appendix and you do not need everything in it for this project.

Matrix tutorial To learn to deal with matrices in Mathematica read the tutorial *Getting and Setting Pieces of Matrices*.

Updating cells in arbitrary order I recently switched to using Mathematica notebooks and my style probably is fairly primitive, so beware! I tend to change cells and execute them in more or less arbitrary order. This is convenient, but it can give unexpected results. For instance, if you change something in the first cell that uses a result of the last cell, there is no problem as long as the last cell has already been executed. That, however, is not the case if you quit Mathematica and restart, as cells are executed in the order in which they appear. If you work in this style, it is convenient to start your program with \texttt{Clear["Global\"\*"]}. Then you can select all cells – I use \texttt{(command) a},
but that is may be different if you are using a different operating system – after which you execute all cells – \(\langle\text{shift}\rangle\langle\text{return}\rangle\) for me. The select-all/execute-all sequence has the same effect as restarting and it guarantees that a program that seemed to work one day still works the next day.

**Help on a subset of commands** Execute \(?\text{FindRoot}\) to obtain help for the command \texttt{FindRoot}; \(?\text{Find}\ast\) gives the same for all commands starting with \texttt{Find}. The asterisk is a wild card character; it stands for any number of arbitrary characters, including none, and can be put anywhere you like. \(?\ast\text{Fi}\ast\), for example, will give you a list of all commands that contain \texttt{Fi} anywhere in their name.

**Subroutine works only in isolation** This is one of the more obscure problems that you may run into when you use Mathematica utilities \texttt{FindMinimum}, \texttt{FindMaximum}, and \texttt{FindRoot}. Read tutorial/UnconstrainedOptimizationSymbolicEvaluation You may not fully understand the tutorial; look at the error messages shown in the tutorial for a case that goes wrong. If you see anything like that, make sure that your function works if you call it directly, and try what happens you attach \_\texttt{NumericQ} to one or more of the function arguments. That is likely to prevent the symbolic evaluation of the function, which causes the problem.

**Summing the elements of a list** Of course, you can do this by using a loop. Mathematica can do text manipulation that makes the code much shorter, although not necessarily faster. If you do \((a+b+c)\//\text{FullForm}\) you get \texttt{Plus\[a,b,c\]}, which shows how Mathematica represents this expression. Similarly the list \{a,b,c\} is represented as \texttt{List\[a,b,c\]}. In other words, if you substitute the word \texttt{Plus} for the word \texttt{List}, you have the desired sum; if the list contains numbers rather than variables, Mathematica realizes that it can actually do the summation. The command \texttt{Head} can be used to find out what the head of a full form expression is, as you can see e.g. with \texttt{Head\[\{a,b,c\}\]. To change the head of an expression you can use \texttt{Apply}: \texttt{Apply\[\{a,b,c\},\texttt{Plus}\]} does the summation trick; an equivalent alternative is \texttt{Plus \@\@ \{a,b,c\}}.

**Summing the squared elements of a list** Many of the built-in operations of Mathematica are called listable, e.g., a one-number operation operation applied to a list is applied to each element of the list. \{a,b,c\}^2 produces \{a^2,b^2,c^2\} and \texttt{Cos\[\{a,b\}\]} gives
\{\text{Cos}[a],\text{Cos}[b]\} \text{ Similarly Plus} @@ (\{a,b,c\}^2) \text{ (Note the parentheses!)} \text{ yields } a^2+b^2+c^2.

**Applying a function to each element of a list** If \( A=\{a_1,a_2,\ldots\} \) is a list and \( f \) is a function with one argument \((f[#] &)/@ A\) applies \( f \) to each element the list, i.e., it produces \( \{f[a_1],f[a_2],\ldots\} \). Here \# stands for whatever the actual argument will be, while & indicates that the preceding construct is function. (See documentation for \text{Map}).