28
Quantum Paths for Functional Integration

This chapter deals with Feynman's path integral formulation of quantum mechanics [83]. It is hardest material in this book. In recent times this path integral formulation has been applied to field theory calculations (quantum chromodynamics) and, in the process, has become a major consumer of the world's high-performance computer time. The calculations we present are based on those of other authors [84–86]. Different approaches and further references can be found in an article in Computers in Physics *[87].*

Problem: As we have seen in this book, and as is known from elementary physics, a classical particle attached to linear spring ($F \propto x$), undergoes simple harmonic motion with position $x(t) = A\sin(\omega_0 t + \phi)$. Your **problem** is to take this classical solution or trajectory $x(t)$, and make a direct connection between it and the quantum wave function $\psi(x,t)$ for a particle bound within a harmonic oscillator potential.

28.1
Feynman's Space–Time Propagation (Theory)

Feynman was looking for a formulation of quantum mechanics that had a more direct connection to classical mechanics than does Schrödinger theory, and which made its statistical aspects apparent from the start. He followed a suggestion by Dirac that Hamilton's principle may lead to classical mechanics occurring as a special case of quantum mechanics for vanishingly small values of $\hbar$. Seeing that Hamilton's principle deals with the paths of particles through space time, Feynman observed that the quantum wave function describing the propagation of a free particle from the space–time point $a = (x_a,t_a)$ to the point $b = (x_b,t_b)$ can be expressed as [83]

$$\psi(x_b,t_b) = \int dx_a\, G(x_b,t_b;x_a,t_a)\psi(x_a,t_a) \tag{28.1}$$

Computationyal Physics. Problem Solving with Computers (2nd edn).
Rubin H. Landau, Manuel José Páez, Cristian C. Bordeianu

ISBN: 978-3-527-40626-5

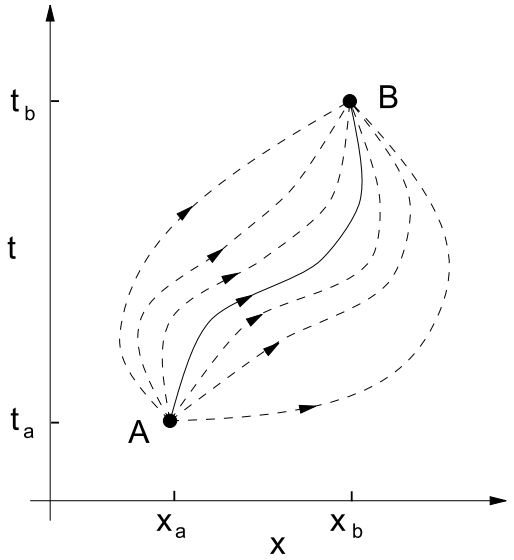

Fig. 28.1 A collection of paths connecting the initial space–time point A to the final point B. The solid line is the trajectory followed by a classical particle while the dashed lines are additional paths sampled by a quantum particle. A classical particle somehow "knows" ahead of time that travel along the classical trajectory minimizes the action S.

where G is the space and time dependent *Green's function* or *propagator*

$$G(x_b, t_b; x_a, t_a) \equiv G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}}\, e^{i\frac{m(x_b - x_a)^2}{2(t_b - t_a)}} \tag{28.2}$$

Equation (28.1) is a form of Huygens's wavelet principle in which each point on the wavefront $\psi(x_a, t_a)$ emits a spherical wavelet $G(b, a)$ that propagates forward in space and time. It states that by summation and interference with all the other wavelets, a new wavefront $\psi(x_b, t_b)$ is created.

Feynman imagined another way of interpreting (28.1) as a form of Hamilton's principle. It envisions the probability amplitude (wave function ψ) for a particle to be at b as equal to the sum over all *paths* through space–time originating at time t_a and ending at b (Fig. 28.1). This view incorporates the statistical nature of quantum mechanics by having different probabilities for travel along the different paths. All paths are possible, but some are more likely than others. [When you realize that Schrödinger theory solves for wave functions and considers paths a classical concept, you appreciate how different a view is Feynman's.] The values for the probabilities of the paths derives from *Hamilton's principle of least action* in classical mechanics:

> *The most general motion of a physical particle moving along the classical trajectory $\bar{x}(t)$ from time t_a to t_b is along a path such that the action*

$S[\bar{x}(t)]$ is an extremum:

$$\delta S[\bar{x}(t)] \;=\; S[\bar{x}(t)+\delta x(t)] - S[\bar{x}(t)] \;=\; 0 \tag{28.3}$$

with the paths constrained to pass through the endpoints:

$$\delta(x_a) = \delta(x_b) = 0$$

This formulation of classical mechanics, which is based on the calculus of variations, is equivalent to Newton's differential equations if the action S is taken as the line integral of the Lagrangian along the path:

$$S[\bar{x}(t)] = \int_{t_a}^{t_b} dt\, L\,[x(t), \dot{x}(t)] \qquad\qquad L = T\,[x, \dot{x}] - V[x] \tag{28.4}$$

Here T is the kinetic energy, V is the potential energy, $\dot{x} = dx/dt$, and a square brackets indicate a *functional*[1] of the function $x(t)$ and $\dot{x}(t)$.

Feynman's insight begins with the observation that the classical action for a free particle ($V = 0$),

$$S[b,a] \;=\; \frac{m}{2}\,(\dot{x})^2\,(t_b - t_a) \;=\; \frac{m}{2}\frac{(x_b - x_a)^2}{t_b - t_a} \tag{28.5}$$

is related to the free-particle propagator (28.2) by

$$G(b,a) \;=\; \sqrt{\frac{m}{2\pi i(t_b - t_a)}}\; e^{iS[b,a]/\hbar} \tag{28.6}$$

This is the much-sought a connection between quantum mechanics and Hamilton's principle. Feynman then postulated a reformulation of quantum mechanics that incorporated its statistical aspects by expressing $G(b,a)$ as the weighted sum over all *paths* connecting a to b,

$$G(b,a) \;=\; \sum_{\text{paths}} e^{iS[b,a]/\hbar} \qquad\qquad \text{(Path integral)} \tag{28.7}$$

Here the classical action S (28.4) is evaluated along different paths (Fig. 28.1), and the exponential of the action is summed over paths. The sum (28.7) is called a *path integral* because it sums over actions $S[b,a]$, each of which is an integral (on the computer an integral and sum are the same anyway). The essential connection between classical and quantum mechanics is the realization

[1] A *functional* is a number whose value depends on the complete behavior of some function and not just on its behavior at one point. For example, the derivative $f'(x)$ depends on the value of f at x, yet the integral $I[f] = \int_a^b dx\, f(x)$ depends on the entire function and is therefore a functional of f.

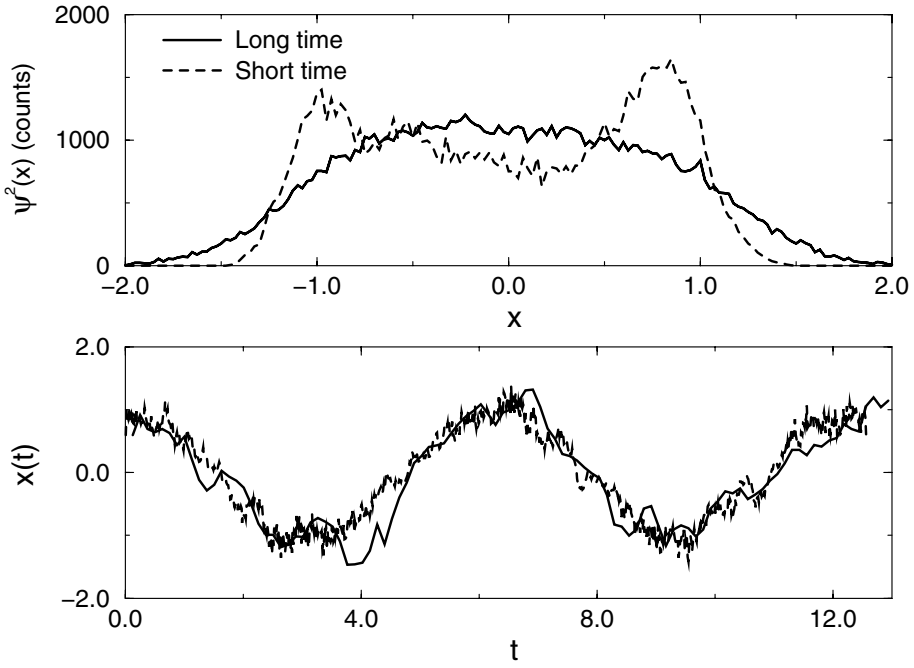

Fig. 28.2 The ground-state wave function of the harmonic oscillator as determined with a path-integral calculation. *Upper*: The dashed curve is the wave function for a short time $t_b - t_a$ (twice the classical period) and the solid curve for a long time (20 times the classical period). The long time yields a wave function closer to the Gaussian form expected for the ground state. *Lower*: The long- and short-time trajectories in space–time used last in the solutions for the wave function. The oscillator has initial and final amplitudes of $x = 1$, $m = k = 1$, and, consequently, a period of $T = 2\pi$.

that in units of $\hbar \simeq 10^{-34}$ Js, the action is a very large number, $S/\hbar \succeq 10^{20}$, and so even though all paths enter into the sum (28.7), the main contributions come from those paths adjacent to the classical trajectory $\bar{x}$. In fact, because S is an extremum for the classical trajectory, it is a constant to first order in variation of paths, and so nearby paths have phases that vary smoothly and relatively slowly. In contrast, those paths far from the classical trajectory are weighted by a rapidly oscillating $\exp(iS/\hbar)$, and when many are included they tend to cancel each other out. In the classical limit, $\hbar \to 0$, only the classical trajectory contributes and (28.7) becomes Hamilton's principle of least action! In Fig. 28.2 we show concrete examples of the trajectories used in actual path-integral calculations.

28.1.1 Bound-State Wave Function (Theory)

Although you may be thinking that you have already seen enough expressions for Green's function, there is yet another one we need for our computation. Let us assume that the Hamiltonian $\tilde{H}$ supports a spectrum of eigenfunctions, $\tilde{H}\psi_n = E_n\psi_n$, labeled by the index n. Because $\tilde{H}$ is hermitian, the solutions form a complete orthonormal set in which we may expand a general solution:

$$\psi(x,t) = \sum_{n=0}^{\infty} c_n\, e^{-iE_n t}\, \psi_n(x) \qquad c_n = \int_{-\infty}^{+\infty} dx\, \psi_n^*(x)\psi(x,t=0) \tag{28.8}$$

where the value for the expansion coefficients c_n follows from orthonormality of ψ_n's. If we substitute this c_n back into the wave function expansion (28.8), we obtain the identity:

$$\psi(x,t) = \int_{-\infty}^{+\infty} dx_0 \sum_n \psi_n^*(x_0)\psi_n(x)e^{-iE_n t}\psi(x_0,t=0) \tag{28.9}$$

Comparison with (28.1) yields the eigenfunction expansion for G:

$$G(x,t;x_0,t_0=0) = \sum_n \psi_n^*(x_0)\psi_n(x)e^{-iE_n t} \tag{28.10}$$

We relate this to the bound-state wave function (recall our **problem** is to calculate that) by (1) requiring all paths to start and end at the space position $x_0 = x$, (2) by taking $t_0 = 0$, and (3) by making an analytic continuation of (28.10) to negative imaginary time (permissable for analytic functions):

$$G(x,-i\tau;x,0) = \sum_n |\psi_n(x)|^2 e^{-E_n\tau} = |\psi_0|^2 e^{-E_0\tau} + |\psi_1|^2 e^{-E_1\tau} + \cdots$$

$$\Rightarrow \quad |\psi_0(x)|^2 = \lim_{\tau\to\infty} e^{E_0\tau} G(x,-i\tau;x,0) \tag{28.11}$$

The limit here corresponds to long imaginary times τ, after which the parts of ψ with higher energies decay away more quickly, leaving only the ground state ψ_0.

Equation (28.11) provides a closed-form solution for the ground-state wave function directly in terms of the propagator G. Although we will soon describe how to compute this equation, look now at Fig. 28.2 showing some results of a computation. We see in the top of the figure that if we wait for only a short imaginary time, then the wave function resembles the classical solution; namely, it has peaks near the classical turning points at the edges of the well. However, if we wait for longer imaginary times, then the wave function resembles the expected Gaussian. The bottom of the figure shows two of the variations on the classical trajectory used in the calculation. Observe that the

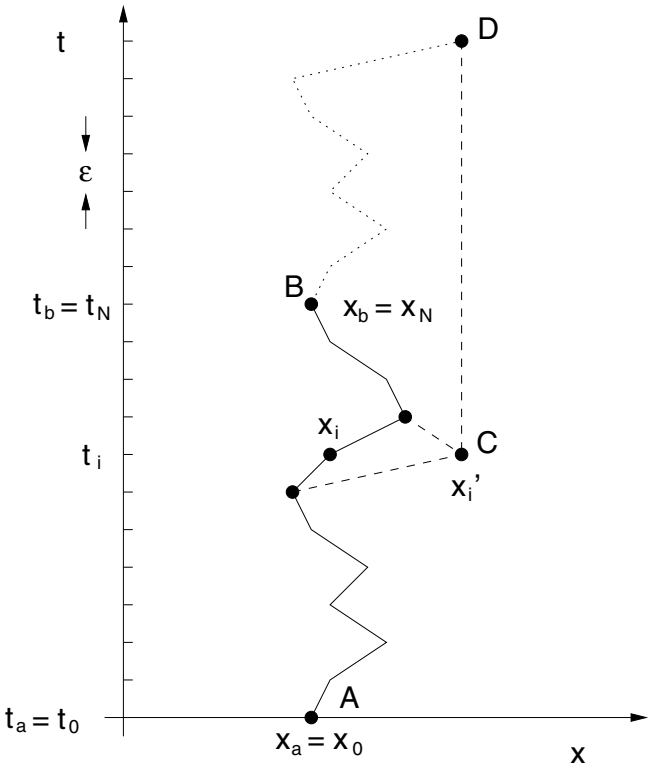

Fig. 28.3 A path through the space–time lattice that starts and ends at $x_a = x_b$. The action is an integral over this path, while the *path integral* is a sum of integrals over all paths. The dotted path BD is a transposed replica of the path AC.

quantum paths are clearly statistical variations about the classical trajectory $x(t) = A\sin(\omega_0 t + \phi)$.

28.1.2 Lattice Path Integration (Algorithm)

Because both time and space get integrated over when evaluating a path integral, we set up a lattice of discrete points in space–time, and visualize a particle's trajectory as a series of straight lines connecting one time to the next (Fig. 28.3). We divide the time between A and B into N equal steps of size ε, and label them with the index j:

$$\varepsilon \stackrel{\text{def}}{=} \frac{t_b - t_a}{N} \qquad \Rightarrow \qquad t_j = t_a + j\varepsilon \qquad (j = 0, N) \tag{28.12}$$

Although more precise to use the actual positions $x(t_j)$ of the trajectory at the times t_j to determine the x_j's (as in Fig. 28.3), in practice we discretize space

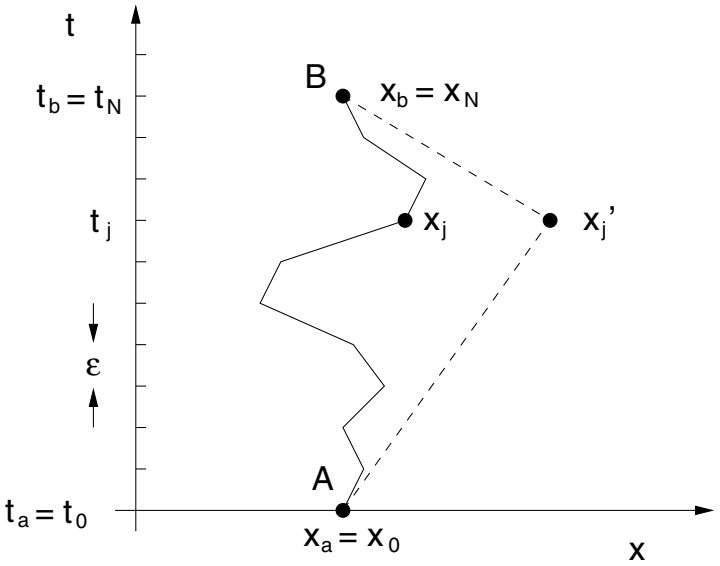

Fig. 28.4 The dashed path joins the initial and final times in two equal time steps, the solid curve uses N steps each of size ε. The position of the curve at time t_j defines the position x_j.

uniformly and have the links end at the nearest regular points. Once we have a lattice, it is easy to evaluate derivatives or integrals on a link [2]:

$$\frac{dx_j}{dt} \simeq \frac{x_j - x_{j-1}}{t_j - t_{j-1}} = \frac{x_j - x_{j-1}}{\varepsilon} \tag{28.13}$$

$$S_j \simeq L_j \Delta t \simeq \frac{1}{2} m \frac{(x_j - x_{j-1})^2}{\varepsilon} - V(x_j)\varepsilon \tag{28.14}$$

where we have assumed that the Lagrangian is constant over each link.

Path integration on a lattice is based on the *composition theorem* for propagators:

$$G(b,a) = \int dx_j \, G(x_b, t_b; x_j, t_j) G(x_j, t_j; x_a, t_a) \qquad (t_a < t_j \, t_j < t_b). \tag{28.15}$$

[2] Even though Euler's rule has a large error, it is often use in lattice calculations because of its simplicity. However, if the Lagrangian contains second derivatives, you should use the more precise central difference method to avoid singularities.

For a free particle, this yields

$$G(b,a) = \sqrt{\frac{m}{2\pi i(t_b - t_j)}}\sqrt{\frac{m}{2\pi i(t_j - t_a)}}\int dx_j\, e^{i(S[b,j]+S[j,a])}$$

$$= \sqrt{\frac{m}{2\pi i(t_b - t_a)}}\int dx_j\, e^{iS[b,a]} \tag{28.16}$$

where we have added the actions since line integrals combine as $S[b,j] + S[j,a] = S[b,a]$. For the N-linked path of Fig. 28.3, Eq. (28.15) becomes

$$G(b,a) = \int dx_1 \cdots dx_{N-1}\, e^{iS[b,a]} \qquad S[b,a] = \sum_{j=1}^{N} S_j \tag{28.17}$$

where S_j is the value of the action for link j. At this point the integral over the *single* path shown in Fig. 28.3 has become an N-term sum that becomes an infinite sum as the time step ε approaches zero.

To summarize, Feynman's path-integral postulate (28.7) means that we sum over all paths connecting a to b to obtain the Green's function $G(b,a)$. This means that we must sum not only over the links in one path but *also* over all different paths in order to produce the variation in paths that is required by Hamilton's principle. The sum is constrained such that paths must pass through a and b and cannot double back on themselves (causality requires that particles move only forward in time). This is the essence of *path integration*. Because we are integrating over functions as well as along paths, the technique is also known as *functional integration*.

The propagator (28.7) is the sum over all paths connecting a to b, with each path weighted by the exponential of the action along that path, explicitly:

$$G(x,t;x_0,t_0) \;=\; \sum\!\!\!\!\!\!\int dx_1 dx_2 \cdots dx_{N-1} e^{iS[x,x_0]} \tag{28.18}$$

$$S[x,x_0] \;=\; \sum_{j=1}^{N-1} S[x_{j+1},x_j] \;\simeq\; \sum_{j=1}^{N-1} L\left(x_j,\dot{x}_j\right)\varepsilon \tag{28.19}$$

where $L(x_j,\dot{x}_j)$ is the average value of the Lagrangian on link j corresponding to time $t = j\varepsilon$. To keep the computation simple, we assume that the potential $V(x)$ is independent of velocity and does not depend on other x values (local potential). Next we observe that in the expression (28.11) for the ground-state wave function, G is evaluated with a negative imaginary time. Accordingly,

we evaluate the Lagrangian with $t = -i\tau$:

$$L(x, \dot{x}) = T - V(x) = +\tfrac{1}{2}m\left(\frac{dx}{dt}\right)^2 - V(x) \tag{28.20}$$

$$\Rightarrow \quad L\left(x, \frac{dx}{-id\tau}\right) = -\tfrac{1}{2}m\left(\frac{dx}{d\tau}\right)^2 - V(x) \tag{28.21}$$

We see that the reversal of the sign of the kinetic energy in L means that L now equals the negative of the Hamiltonian evaluated at a real positive time $t = \tau$:

$$H\left(x, \frac{dx}{d\tau}\right) = \tfrac{1}{2}m\left(\frac{dx}{d\tau}\right)^2 + V(x) = E \tag{28.22}$$

$$\Rightarrow \qquad L\left(x, \frac{dx}{-id\tau}\right) = -H\left(x, \frac{dx}{d\tau}\right) \tag{28.23}$$

In this way we rewrite the t-path integral of L as a τ-path integral of H, and so express the action and Green's function in terms of the Hamiltonian:

$$S[j+1, j] \;=\; \int_{t_j}^{t_{j+1}} L(x,t)\,dt \;=\; -i\int_{\tau_j}^{\tau_{j+1}} H(x,\tau)d\tau \tag{28.24}$$

$$\Rightarrow \quad G(x, -i\tau; x_0, 0) \;=\; \int dx_1 \cdots dx_{N-1}\, e^{-\int_0^\tau H(\tau')d\tau'} \tag{28.25}$$

where the line integral of H is over an entire trajectory. Next we express the path integral in terms of an average energy of the particle on each link $E_j = T_j + V_j$, and then sum over links[3] to obtain the summed energy $\mathcal{E}$:

$$\int H(\tau)d\tau \;\simeq\; \sum_j \varepsilon E_j = \varepsilon\mathcal{E}(\{x_j\}) \tag{28.26}$$

$$\mathcal{E}(\{x_j\}) \quad \stackrel{\text{def}}{=} \quad \sum_{j=1}^{N}\left[\frac{m}{2}\left(\frac{x_j - x_{j-1}}{\varepsilon}\right)^2 + V\left(\frac{x_j + x_{j-1}}{2}\right)\right] \tag{28.27}$$

In (28.27) we have approximated each path link as a *straight line*, used Euler's derivative rule for the velocity, and evaluated the potential at the midpoint of each link. We now substitute this expression for G into our solution (28.11)

[3] In some cases, such as for an infinite square well, this can cause problems if the trial link cause the energy to be infinite. In that case, one can modify the algorithm to use the potential at the beginning of a link.

for the ground-state wave function, which requires that the initial and final points in space be the same:

$$\lim_{\tau\to\infty}\frac{G(x,-i\tau,x_0=x,0)}{\int dx\,G(x,-i\tau,x_0=x,0)}=\frac{\int dx_1\cdots dx_{N-1}\exp\left[-\int_0^\tau H d\tau'\right]}{\int dx dx_1\cdots dx_{N-1}\exp\left[-\int_0^\tau H d\tau'\right]}$$

$$\Rightarrow\qquad |\psi_0(x)|^2=\frac{1}{Z}\lim_{\tau\to\infty}\int dx_1\cdots dx_{N-1}\,e^{-\varepsilon\mathcal{E}} \tag{28.28}$$

$$Z=\lim_{\tau\to\infty}\int dx\,dx_1\cdots dx_{N-1}e^{-\varepsilon\mathcal{E}} \tag{28.29}$$

The similarity of these expressions to thermodynamics, even with a partition function Z, is no accident; by making the time parameter of quantum mechanics imaginary, we have converted the time-dependent Schrödinger equation into the heat-diffusion equation:

$$i\frac{\partial\psi}{\partial(-i\tau)}=\frac{-\nabla^2}{2m}\psi\qquad\rightarrow\quad\frac{\partial\psi}{\partial\tau}=\frac{\nabla^2}{2m}\psi. \tag{28.30}$$

It is not a surprise then that the sum over paths in Green's function has each path weighted by the Boltzmann factor $\mathcal{P}=e^{-\varepsilon\mathcal{E}}$ that is usually associating with thermodynamics. We make the connection complete by identifying the temperature with the inverse time step:

$$\mathcal{P}=e^{-\varepsilon\mathcal{E}}=e^{-\mathcal{E}/k_BT},\qquad\Rightarrow\qquad k_BT=\frac{1}{\varepsilon}\equiv\frac{\hbar}{\varepsilon}. \tag{28.31}$$

Consequently, the $\varepsilon\to 0$ limit, which makes time continuous, is a "high-temperature" limit. The $\tau\to\infty$ limit, which is required to project out the ground-state wave function, means that we must integrate over a path that is long in imaginary time, that is, long compared to a typical time $\hbar/\Delta E$. Just as our simulation of the Ising model in Chapter 12 required us to wait a long time while the system equilibrated, so the present simulation requires us to wait around a long time so that all but the ground state wave function has decayed away. Alas, the solution to our **problem**.

To summarize, we have expressed Green's function as an evaluation of the path integral (28.28), which requires integration of the Hamiltonian along paths and a summation over all paths. We evaluate this path integral as the sum over all trajectories in our space–time lattice. The links on each path, and correspondingly each trial path, occur with a probability based on its action. We use the Metropolis algorithm to fluctuate the links as if they are in thermal equilibrium and obeying a Boltzmann distribution of energy. This is similar to our work with the Ising model in Chapter 12, however, now, rather than reject or accept a flip in spin based on the change in energy, we reject or accept

a change in a link based on the change in energy. The more iterations we let the algorithm run for, the more the determined wave function equilibrates to the ground state wave function.

In general, Monte Carlo Green's function techniques work best if we start off with a good guess at the correct answer and have the algorithm calculate variations to our guess. For the present problem this means that if we start off with a path in space–time close to the classical trajectory, the algorithm may be expected to do a good job at simulating the quantum fluctuations about the classical trajectory. However, it does not appear to be good at finding the classical trajectory from arbitrary locations in space time. We suspect that the latter arises from $\delta S/\hbar$ being so large, that $\exp(\delta S/\hbar)$ fluctuates wildly (essentially averaging out to zero) and so loses its sensitivity.

28.1.2.1 **A Time Saving Trick**

As we have formulated the computation, we pick a value of x and perform a rather lengthy computation of line integrals over all space and time to obtain $|\psi_0(x)|^2$ at one x. To obtain the wave function at another x, the entire simulation would have to be repeated from scratch. Rather than go through all that trouble again and again, we will compute the entire x dependence of the wave function in one fell swoop. The trick is to insert a delta function into the probability integral (28.28), thereby fixing the initial position to be x_0, and then to integrate over all values for x_0:

$$|\psi_0(x)|^2 = \int dx_1 \cdots dx_N \, e^{-\varepsilon \mathcal{E}(x,x_1,\ldots)} = \int dx_0 \cdots dx_N \delta(x - x_0) \, e^{-\varepsilon \mathcal{E}(x,x_1,\ldots)}. \tag{28.32}$$

This equation expresses the wave function as an average of a delta function over all paths, a procedure that might appear totally inappropriate for numerical computation because there is tremendous error in representing a singular function on a finite-word-length computer. Yet when we simulate the sum over all paths with (28.32), there will always be some x value for which the integral is nonzero, and we need to only accumulate the solution for various (discrete) x values to determine $|\psi_0(x)|^2$ for all x.

To understand how this works in practise, consider path AB in Figs. 28.3 and 28.4 for which we have just calculated the summed energy. We next form a new path by having one point on the chain jump to point C (which changes two links). If we replicate section AC and use it as the extension AD to form the top path, we see that the path CBD has the same summed energy (action) as path ACB and in this way can be used to determine $|\psi(x'_j)|^2$. That being the case, once the system is equilibrated, we determine new values of the wave function at new locations x'_j by flipping links to new values and calculating

new actions. The more frequently some x_j is accepted, the greater is the wave function at that point.

Listing 28.1: `QMC.java` solves for the ground state probability distribution via a Feynman path integration using the Metropolis algorithm to simulate variations about the classical trajectory.

```
// QMC.java: Quantum MonteCarlo Feynman path integration

import java.io.*;                                  // Location of PrintWriter
import java.util.*;                                  // Location of Random
import java.lang.*;                                    // Location of Math

public class QMC {

  public static void main(String[] argv)
                        throws IOException, FileNotFoundException {

    PrintWriter q = new PrintWriter(                        // File output
                          new FileOutputStream("QMC.DAT"), true);
    int N = 100, M = 101, Trials = 25000, seedTrials = 200;
    double path[] = new double[N], xscale = 10.;
    long prop[]   = new long[M],   seed = 10199435;
                                                          // Begin Trials
    for ( int count = 0; count < seedTrials*10; count +=  10) {
      Random randnum = new Random(seed + count);
      double change = 0., newE = 0., oldE = 0. ;
                                                          // Initial path
      for ( int i=0; i < N; i++ )  path[i] = 0. ;
      oldE = energy(path);                              // Find E of path
                                                  // Pick random element
      for ( int i=0; i < Trials; i++ )  {
        int element = randnum.nextInt(N);
        change = 1.8*(0.5 - randnum.nextDouble());
        path[element] += change;                          // Change path
        newE = energy(path);                               // Find new E
                  // Metroplis algorithm
        if ( newE > oldE && Math.exp(-newE + oldE)
                                                                // Reject
          <= randnum.nextDouble() ) path[element]-=change;
                                                  // Add probabilities
        for ( int j=0; j < N; j++ )  {
          element = (int)Math.round( (M-1)*(path[j]/xscale + .5) );
          if (element < M && element>=0)  prop[element]++ ;
       }
       oldE = newE;
      }                                                       // t loop
    }                                                       // Seed loop
    for ( int i=0; i < M; i++ )  q.println(xscale*(i-(M-1)/2)
        + " " + (double)prop[i]/((double)Trials*(double)seedTrials));
    System.out.println(" ");
    System.out.println("QMC Program Complete.");
    System.out.println("Data stored in QMC.DAT");
    System.out.println(" ");
  }

  public static double energy(double path[])  {
```

```
    int i = 0;
    double sum = 0. ;

    for ( i=0; i < path.length-2; i++ )
      { sum +=  (path[i+1] - path[i])*(path[i+1] - path[i]) ;  }
    sum +=  path[i+1]*path[i+1];
    return sum;
} }                                                    // End class
```

28.1.3 Lattice Implementation

The program `QMC.java` in List 28.1 evaluates the integral (28.7) by finding the average of the integrand $\delta(x_0 - x)$ with paths distributed according to the weighting function $\exp[-\varepsilon\mathcal{E}(x_0, x_1, \ldots, x_N)]$. The physics enters via (28.34), the calculation of the summed energy $\mathcal{E}(x_0, x_1, \ldots, x_N)$. We evaluate the action integral for the harmonic oscillator potential

$$V(x) = \frac{1}{2}x^2 \tag{28.33}$$

and for a particle of mass $m = 1$. A convenient set of natural units is to measure lengths in $\sqrt{1/m\omega} \equiv \sqrt{\hbar/m\omega} = 1$, and times in $1/\omega = 1$. Correspondingly, the oscillator has a period $T = 2\pi$. Figure 28.2 shows results from an application of the Metropolis algorithm. In this computation we started off with an initial path close to the classical trajectory and then examined one-half million variations about this path. All paths are constrained to begin and end at $x = 1$ (which turns out to be somewhat less than the amplitude of the classical oscillation).

When the time difference $t_b - t_a$ equals a short time like $2T$, the system has not had enough time to equilibrate to its ground state and, as we see in the top of Fig. 28.2, the wave function looks like the probability distribution of an excited state (nearly classical with the probability highest for the particle to be near its turning points where its velocity vanishes). However, when the time difference $t_b - t_a$ equals the longer time $20T$, the system has enough time to decay to its ground state and the wave function looks like the expected Gaussian distribution. In either case, we see in the bottom part of Fig. 28.2 that the trajectory through space–time fluctuates about the classical trajectory. This fluctuation is a consequence of the Metropolis algorithm occasionally going uphill in its search; if you modify the program so that searches go only downhill, the space–time trajectory would be a very smooth trigonometric function (the classical trajectory), but the wave function, which is a measure of the fluctuations about the classical trajectory, would vanish! The explicit steps of the calculation are

1. Construct a time grid of N time steps of length ε (Fig. 28.3). Start at $t = 0$ and extend to time $\tau = N\varepsilon$ [this means N time intervals and $(N + 1)$ lattice points in time]. Note that time always increases monotonically along a path.

2. Construct a space grid of M points separated by steps of size δ. Use a range of x values several time larger than the characteristic size or range of the potential being used, and start with $M \simeq N$.

3. When calculating the wave function, any x or t value falling between lattice points should be assigned to the closest lattice point.

4. Associate a position x_j with each time τ_j, subject to the boundary conditions that the initial and final positions always remain the same, $x_N = x_0 = x$.

5. Choose an *arbitrary path* of straight-line links connecting the lattice points. For the most realistic simulation it may be best to start with something close to the classical trajectory, as otherwise the simple numerical procedures may not converge. Note that the x values for the links of the path may have values that increase, decrease, or remain unchanged (in contrast to time, which always increases).

6. Evaluate the energy $\mathcal{E}$ by summing the kinetic and potential energies for each link of the path starting at $j = 0$:

$$\mathcal{E}(x_0, x_1, \ldots, x_N) \simeq \sum_{j=1}^{N} \left[\frac{m}{2} \left(\frac{x_j - x_{j-1}}{\varepsilon} \right)^2 + V \left(\frac{x_j + x_{j-1}}{2} \right) \right] \quad (28.34)$$

7. Begin the first of a sequence of repetitive steps in which a random position x_j associated with time t_j is changed to the position x_j' (point C in Fig. 28.4). This changes *two* links in the path.

8. For the coordinate that gets changed, weigh the change with the Boltzmann factor (28.31) by using the Metropolis algorithm.

9. For each lattice point, establish a running sum to represent the value of the wave function squared at that point.

10. After each single-link change (or decision not to change), increase the running sum for the new x value by 1. After a sufficiently long running time, the sum divided by the number of steps is the simulated value for $|\psi(x_j)|^2$ at each lattice point x_j.

11. Repeat the entire link-changing simulation using a different seed for the Metropolis algorithm. The average wave function from a number of intermediate-length runs should be better than that from one very long run.

28.1.4 Assessment and Exploration

1. Examine some of the actual space–time paths used in the simulation. Compare those paths to the classical trajectory.

2. For a more continuous picture of the wave function, make the x lattice spacing smaller; for a more precise value of the wave function at any particular lattice site, sample more points (run longer) and use a smaller time step ε.

3. Because there are no sign changes in a ground-state wave function, you can ignore the phase and assume $\psi(x) = \sqrt{\psi^2(x)}$, and then estimate the energy via

$$E = \langle\psi| H |\psi\rangle = \frac{\omega}{2}\int_{-\infty}^{+\infty} \psi^*(x)\left(-\frac{d^2}{dx^2} + x^2\right)\psi(x)dx \qquad (28.35)$$

   where the space derivative is evaluated numerically.

4. Explore the effect of making $\hbar$ larger, and thus permitting greater fluctuations around the classical trajectory. Do this by decreasing the value of the exponent in the Boltzman factor. Determine if this makes the calculation more or less robust in its ability to find the classical trajectory.

5. Test the wave function computation for the gravitational potential

$$V(x) = mg|x|, \qquad x(t) = x_0 + v_0 t + \tfrac{1}{2}gt^2 \qquad (28.36)$$

   You may want to set the initial positions to be close to the classical trajectory to ensure convergence.