Chapter 6

Biasing for a Fair Return

Folkmar Bornemann

> *It was often claimed that [direct and "exact" numerical solution of the equations of physics] would make the special functions redundant. ... The persistence of special functions is puzzling as well as surprising. What are they, other than just names for mathematical objects that are useful only in situations of contrived simplicity? Why are we so pleased when a complicated calculation "comes out" as a Bessel function, or a Laguerre polynomial? What determines which functions are "special"?*
>
> — Sir Michael Berry [Ber01]

> *People who like this sort of thing will find this the sort of thing they like.*
>
> — Barry Hughes, quoting Abraham Lincoln at the beginning of an appendix on "Special Functions for Random Walk Problems" [Hug95, p. 569]

Problem 6

> *A flea starts at* $(0, 0)$ *on the infinite two-dimensional integer lattice and executes a biased random walk: At each step it hops north or south with probability* $1/4$*, east with probability* $1/4 + \epsilon$*, and west with probability* $1/4 - \epsilon$*. The probability that the flea returns to* $(0, 0)$ *sometime during its wanderings is* $1/2$*. What is* ϵ*?*

Asking for the ϵ that gives a certain probability p of return yields a problem hardly any more difficult than calculating the probability for a given ϵ: it just adds the need to use a numerical root-finder. But the problem looks more interesting the way it is stated. In §6.1 we give a short argument, why the problem is solvable.

We will discuss several methods for calculating the probability of return. In §6.2, using virtually no probability theory, we transform the problem to one of linear algebra. Solving a sparse linear system of dimension 25 920 gives us 15 correct digits. The main story, told in §§6.3–6.5, is based on the relation between the probability p of return and the expected number E of visits to the starting site, namely $E = 1/(1-p)$. We represent E as an infinite series and, by stepwise decreasing the computational effort while increasing the level of analytic sophistication, go from a brute-force numerical approximation to a symbolic evaluation using special functions. The latter results in a closed formula involving the arithmetic-geometric mean M,

$$p = 1 - M\left(\sqrt{1-(1+\eta)^2/4}, \sqrt{1-(1-\eta)^2/4}\right), \qquad \eta = \sqrt{1-16\epsilon^2}. \tag{6.1}$$

To answer Berry's question from the quote at the beginning of this chapter: *we* are so pleased that it "comes out" as such an expression because there is an exceedingly fast algorithm known for its evaluation. It allows us to solve Problem 6 to 10 000 digits in less than a second. And even more, we can validate these digits using interval arithmetic. Finally, in §6.6 we use the technique of lattice Green functions and Fourier analysis to establish a further expression for E, a double integral. Using adaptive numerical quadrature we can thus solve Problem 6 with just three lines of Matlab code.

6.1 A First Look: Is it Solvable?

Before we start developing methods for calculating the probability of return, let us convince ourselves that root-finding for Problem 6 will yield a result. To this end, we look at the two extreme cases of the bias $0 \leqslant \epsilon \leqslant 1/4$.[1] In one of the earliest papers on random walks [Pól21], dating back to 1921, Pólya established the classic result that for the unbiased random walk in two dimensions the probability of return is one, that is $p|_{\epsilon=0} = 1$. We will come back to this point with a proof in §6.5. On the other hand, in the case of maximum bias, $\epsilon = 1/4$, at each step the walker (aka the flea) hops to the east with probability $1/2$ but never to the west. In going a step to the east he would thus prevent himself from ever returning. Hence, the probability of return is at most that of not going east in the first step, $p|_{\epsilon=1/4} \leqslant 1/2$. By continuity we can conclude that there is a bias $0 < \epsilon_* \leqslant 1/4$ such that $p_{\epsilon=\epsilon_*} = 1/2$.

Notation and Terminology. Throughout the chapter we adopt the following notation. The (nonnegative) probabilities for a step to the east, west, north, or south, are denoted by p_{E}, p_{W}, p_{N}, or p_{S}, respectively. This way we are more general than the problem demands, but we will benefit from much cleaner formulas. We also do not assume that the possibilities of a transition to either direction are exhaustive, that is we do not assume that $p_{\mathrm{E}} + p_{\mathrm{W}} + p_{\mathrm{N}} + p_{\mathrm{S}} = 1$, but only that

$$p_{\mathrm{E}} + p_{\mathrm{W}} + p_{\mathrm{N}} + p_{\mathrm{S}} \leqslant 1.$$

The excess probability $p_{\mathrm{kill}} = 1 - p_{\mathrm{E}} - p_{\mathrm{W}} - p_{\mathrm{N}} - p_{\mathrm{S}}$ will be interpreted as the probability that the walker vanishes altogether, that is, the walk stops at the current

[1] By the symmetries of the problem we can restrict ourselves to nonnegative ϵ.

lattice point. Sometimes p_{kill} is called the *killing rate* [Hug95, §3.2.4, p. 123], and the walker is said to be *mortal* if $p_{\text{kill}} > 0$.

We introduce some further useful terminology, common in the literature on random walks [Hug95, p. 122]. If the eventual return to the starting site is certain, that is, if $p = 1$, the random walk is called *recurrent*. Otherwise, when $p < 1$, the random walk is called *transient*.

6.2 Using Numerical Linear Algebra

Sometimes in mathematics a problem becomes much easier if we try to solve for more. Instead of just asking for the probability p that the walker starting at $(0,0)$ reaches $(0,0)$ *again*, we consider the probability $q(x,y)$ that the walker, starting at a lattice point (x,y), *ever* reaches $(0,0)$. Of course, we have

$$q(0,0) = 1. \tag{6.2}$$

So, how can $q : \mathbb{Z}^2 \to [0,1]$ help us in calculating p? The point is that a returning walker has left $(0,0)$ in his first step to a nearest neighbor and has reached $(0,0)$ from there. Thus, we obtain

$$p = p_{\text{E}}\, q(1,0) + p_{\text{W}}\, q(-1,0) + p_{\text{N}}\, q(0,1) + p_{\text{S}}\, q(0,-1). \tag{6.3}$$

A similar argument links the values of q themselves: the probability that the walker reaches $(0,0)$ from the lattice point $(x,y) \neq (0,0)$ can be expressed in terms of the probability that he moves to any of its nearest neighbors and reaches $(0,0)$ from there. In this way we obtain the *partial difference equation*

$$q(x,y) = p_{\text{E}}\, q(x+1,y) + p_{\text{W}}\, q(x-1,y) + p_{\text{N}}\, q(x,y+1) + p_{\text{S}}\, q(x,y-1), \tag{6.4}$$

with $(x,y) \neq (0,0)$, subject to the boundary condition (6.2). We will encounter a similar argument in §10.2, where we will approximate a Brownian motion by a random walk.

The partial difference equation (6.4) constitutes an infinite-dimensional linear system of equations for the unknowns $q(x,y)$, $(x,y) \in \mathbb{Z}^2 \setminus (0,0)$. If we confine ourselves to a finite spatial region of the lattice, say

$$\Omega_n = \{(x,y) \in \mathbb{Z}^2 : |x|, |y| \leqslant n\},$$

we have to supply the neighboring values of q as boundary values. Now, it is reasonable to expect that reaching $(0,0)$ becomes increasingly unlikely for a walker starting at points further and further away from the origin:

$$\lim_{n\to\infty} \sup_{(x,y)\notin\Omega_n} q(x,y) = 0.$$

Mimicking this, we approximate q by the solution q_n of the linear system

$$q_n(x,y) = p_{\text{E}}\, q_n(x+1,y) + p_{\text{W}}\, q_n(x-1,y) + p_{\text{N}}\, q_n(x,y+1) + p_{\text{S}}\, q_n(x,y-1), \tag{6.5}$$

with $(x, y) \in \Omega_n \setminus (0, 0)$, subject to the boundary values

$$q_n(0,0) = 1, \qquad q_n(x,y) = 0 \quad \text{for} \quad (x,y) \notin \Omega_n.$$

One can prove the convergence $q_n(x, y) \to q(x, y)$, uniformly in (x, y). In fact, the convergence is exponentially fast in n. A proof can be based on the properties of the lattice Green function that we will determine in §6.6. However, we will not go into details here because the result obtained will be validated by other methods later on, which are easier to analyze.

The difference equation (6.5) forms a linear system of N equations in the unknowns $q_n(x, y)$, $(x, y) \in \Omega_n \setminus (0, 0)$, with the dimension $N = (2n + 1)^2 - 1$. In analogy to the five-point stencil discretization of Poisson's equation, cf. §10.3, this linear system can be brought to the matrix-vector form

$$A_N x_N = b_N,$$

with a given vector $b_N \in \mathbb{R}^N$ and a sparse matrix $A_N \in \mathbb{R}^{N \times N}$ with just five nonzero diagonals. In Appendix C.3.1 the reader will find the very short Matlab function `ReturnProbability`, which generates A_N using ideas from discrete Poisson's equations (such as the Kronecker tensor product, cf. [Dem97, §6.3.3]), solves the linear system using Matlab's built-in sparse linear solver, and outputs an approximation to the return probability p using (6.3). A *Mathematica* version is in Appendix C.5.2.

Having this in hand, we can solve the equation $p|_{\epsilon=\epsilon_*} = 1/2$ for ϵ_* using Matlab's root-finder. Figure 6.1, obtained with small n for plotting accuracy only, shows that there is just one positive solution $\epsilon_* \approx 0.06$.

A Matlab session.

```
>> f = inline(’ReturnProbability(epsilon,n)-0.5’,’epsilon’,’n’);
>> for n=10*2.^(0:3)
>>    out=sprintf(’n = %3i\t N = %6i\t\t epsilon* = %18.16f’,n,...,
>>       (2*n+1)^2-1,fzero(f,[0.06,0.07],optimset(’TolX’,1e-16),n));
>>    disp(out);
>> end
n =  10  N =    440     epsilon* = 0.0614027632354456
n =  20  N =   1680     epsilon* = 0.0619113908202284
n =  40  N =   6560     epsilon* = 0.0619139543917373
n =  80  N =  25920     epsilon* = 0.0619139544739909
```

The run with $n = 80$ takes 11 seconds on a 2 GHz PC.

How do we assess the accuracy of these four approximations? First, we have to be sure that the sparse linear solver itself does not distort the result. This can be checked using techniques from a posteriori error control that we will discuss in §7.4.1. Given that the linear solver is accurate to at least 15 digits in IEEE arithmetic, we observe that the first two approximations agree to 2 digits, the second and third to 4 digits, the third and fourth to 8 digits. It appears that doubling n doubles the number of correct digits, which is experimental evidence for the claimed

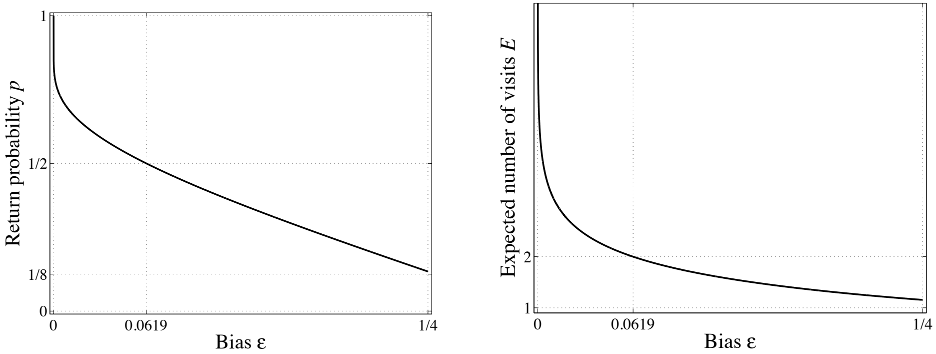

Figure 6.1. *Return probability p and expectation E as a function of the bias ϵ.*

exponentially fast convergence rate. Assuming this we would expect about 15 correct digits at $n = 80$ (which means solving a linear system of $N = 25\,920$ equations), that is

$$\epsilon_* \doteq 0.0619139544739909.$$

The following sections will show that these digits are indeed correct.

6.3 Expectations

From now on we utilize the relation of the probability p of return to the expected number E of visits to the starting site, including the initial visit to this spot. To establish this relation we observe that the probability of exactly k visits is given by $p^{k-1}(1-p)$: the walker has to return $k-1$ times in succession and then to hop away for ever. Thus, the expected number of visits is simply [Fel50, Thm. 2, §12.3]

$$E = \sum_{k=1}^{\infty} k\, p^{k-1}(1-p) = \frac{1}{1-p}, \tag{6.6}$$

with the understanding that in the case of a *recurrent* walk ($p = 1$) we have $E = \infty$. Alternatively, we could derive (6.6) by arguing that after the second visit to the starting site the walker's future is, in probability, just the same as initially. That is, the expected number of visits satisfies the simple equation $E = 1 + pE$.

The point is that the quantity E can be expressed in various other ways that are computationally accessible. In fact, we will give several such expressions in the course of this chapter. A first useful expression is the series

$$E = \sum_{k=0}^{\infty} p_k,$$

where p_k denotes the probability of occupying the starting site at step $2k$. If the series diverges, the walk is recurrent ($E = \infty$) by [Fel50, Thm. 2, §12.3]. We will approximate E by the partial sums

$$E_K = \sum_{k=0}^{K-1} p_k$$

for K sufficiently large. As it turns out there will be a trade-off between the complexity of the algorithm for calculating the terms $p_0, \ldots, p_{K-1}$ and the mathematical sophistication of its derivation: the faster the algorithm, the more theory we have to invest.

6.3.1 Using Brute Force

Here we will start with a straightforward simple algorithm of complexity $O(K^3)$. Convergence acceleration will help us to keep K reasonably small. Later, in §6.4, we will increase the efficiency of calculating E_K in two steps. Some combinatorics reduces the complexity to $O(K^2)$. Zeilberger's creative telescoping algorithm yields a three-term recurrence for p_k that allows us to calculate $p_0, \ldots, p_{K-1}$ with optimal complexity $O(K)$. Then, calculating partial sums for sufficiently large K is so affordable that acceleration of the convergence is no longer necessary. Finally, in §6.5, we will use special functions to evaluate the series symbolically.

We consider the probabilities $P_k(0,0)$ of visiting, at step k, the starting site $(0,0) \in \mathbb{Z}^2$. Then the expected number of visits to $(0,0)$ is [Fel50, Thm. 2, §12.3]

$$E = \sum_{k=0}^{\infty} P_k(0,0). \tag{6.7}$$

Now, we can calculate $P_k(0,0)$ directly from the rules of the random walk, if we know the probabilities of occupying the neighbors of the starting site at the step $k-1$. For those neighbors we do the same, recursively up to the initial step $k=0$ where all is known. Thus, if we introduce the lattice function $P_k : \mathbb{Z}^2 \to [0,1]$ that assigns to each lattice point the probability of being occupied at step k, we obtain the partial difference equation

$$\begin{multline} P_k(x,y) = p_{\mathrm{E}}\, P_{k-1}(x-1,y) + p_{\mathrm{W}}\, P_{k-1}(x+1,y) \\ + p_{\mathrm{N}}\, P_{k-1}(x,y-1) + p_{\mathrm{S}}\, P_{k-1}(x,y+1), \end{multline} \tag{6.8}$$

subject to the initial condition

$$P_0(0,0) = 1, \qquad P_0(x,y) = 0 \qquad \text{for } (x,y) \neq (0,0). \tag{6.9}$$

With a little thought we can arrange the difference equation in a form that avoids the handling and storage of zero probabilities. Because in step k the only sites that can be occupied are those no farther than k steps away from the starting site, we observe that

$$P_k(x,y) = 0 \qquad \text{for} \qquad |x| + |y| > k.$$

In fact, there are further lattice points that cannot be occupied at step k. Taking a checkerboard coloring of the lattice, the walker has to switch colors, that is the parity of $x+y$, at each step:

$$P_{2k}(x,y) = 0 \quad \text{for } x+y \text{ odd}, \qquad P_{2k+1}(x,y) = 0 \quad \text{for } x+y \text{ even}.$$

Now, we arrange all the nontrivial probabilities at step k into the $(k+1)\times(k+1)$ matrix Π_k defined by

$$\Pi_k = \begin{pmatrix} P_k(-k,0) & P_k(-k+1,1) & \cdots & P_k(0,k) \\ P_k(-k+1,-1) & P_k(-k+2,0) & \cdots & P_k(1,k-1) \\ \vdots & \vdots & \ddots & \vdots \\ P_k(0,-k) & P_k(1,-k+1) & \cdots & P_k(k,0) \end{pmatrix}.$$

The partial difference equation (6.8) can be rewritten as the matrix recurrence

$$\begin{aligned} \Pi_{k+1} = p_{\mathrm{E}} &\left(\begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \Pi_k & \\ 0 & & & \end{array}\right) + p_{\mathrm{W}} \left(\begin{array}{ccc|c} & & & 0 \\ & \Pi_k & & \vdots \\ & & & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right) + \\ & p_{\mathrm{N}} \left(\begin{array}{c|ccc} 0 & & & \\ \vdots & & \Pi_k & \\ 0 & & & \\ \hline 0 & 0 & \cdots & 0 \end{array}\right) + p_{\mathrm{S}} \left(\begin{array}{ccc|c} 0 & \cdots & 0 & 0 \\ \hline & & & 0 \\ & \Pi_k & & \vdots \\ & & & 0 \end{array}\right), \end{aligned} \tag{6.10}$$

subject to the initial condition $\Pi_0 = \big(1\big)$. The probabilities for the starting site are obtained as

$$P_{2k+1}(0,0) = 0, \qquad P_{2k}(0,0) = \text{center entry of } \Pi_{2k}.$$

Since the starting site can be occupied only at every second step, we simplify our notation and use

$$p_k = P_{2k}(0,0), \qquad E = \sum_{k=0}^{\infty} p_k. \tag{6.11}$$

The matrix recurrence (6.10) can easily be coded; the reader will find the quite compact Matlab function `OccupationProbability` in Appendix C.3.1. Using this algorithm, the cost of calculating $p_0,\ldots,p_{K-1}$, and hence that of calculating the partial sum

$$E_K = \sum_{k=0}^{K-1} p_k,$$

grows as $O(K^3)$. Let us have faith and solve Problem 6. That is with

$$p_{\mathrm{E}} = 1/4 + \epsilon, \quad p_{\mathrm{W}} = 1/4 - \epsilon, \quad p_{\mathrm{N}} = 1/4, \quad p_{\mathrm{S}} = 1/4,$$

we look for the bias ϵ_* that solves the equation $E|_{\epsilon=\epsilon_*} = 2$, which by (6.6) is equivalent to biasing for a fair return, $p|_{\epsilon=\epsilon_*} = 1/2$.

A Matlab session.

```
>> f=inline('sum(OccupationProbability(epsilon,K))-2','epsilon','K');
>> for K=125*2.^(0:3)
>>    out=sprintf('K = %4i \t\t epsilon* = %17.15f',K,...
>>      fzero(f,[0.06,0.07],optimset('TolX',1e-14),K));
>>    disp(out);
>> end

K =  125        epsilon* = 0.061778241155115
K =  250        epsilon* = 0.061912524106289
K =  500        epsilon* = 0.061913954180807
K = 1000        epsilon* = 0.061913954473991
```

The run with $K = 125$ takes 9 seconds, that one with $K = 1000$ about 2.5 hours on a 2 GHz PC.

How do we assess the accuracy of these four approximations? We observe that the first two agree to 2 digits, the second and third to 4 digits, the third and fourth to 8 digits. It appears that doubling K doubles the number of correct digits, that is, the convergence of the series is roughly exponential.[2] Taking this for granted we would expect 16 correct digits at $K = 1000$. However, the absolute error of the root finder was put to 10^{-14}, which restricts the accuracy of the fourth approximation to about 12 digits. Because the 11th and 12th digit read `39`, we have to be careful with them: they could be `40` as well. In any case, we have no reason to question the correctness of the first ten digits

$$\epsilon_* \doteq 0.06191395447.$$

In fact, we will see later that the $K = 1000$ run is correct to 13 digits.

6.3.2 Using Convergence Acceleration

The run time of 2.5 hours is a good reason for trying convergence acceleration. We use E_K and estimate the tail $E - E_K$ by Wynn's epsilon algorithm (see Appendix A, p. 247), which is particularly well suited for the near-exponential convergence of the series. A good choice of parameters requires some experimentation. A reasonable compromise between run time and accuracy turns out to be $K \approx 400$, for which E_K itself gives just 6 correct digits. To determine the number of extrapolation steps, and hence the number of extra terms, we follow the recommendations of Appendix A (see p. 255) and look, for the particular choice $\epsilon = 0.06$, at the differences of the first row of the extrapolation table, cf. Figure 6.2. We observe that roundoff error becomes significant at about $j \geqslant 3$, where the magnitude of the differences settles at about 10^{-12}. This corresponds to 12 correct digits of E, a gain of 6 digits over just using E_K with $K = 400$. The reader will find the Matlab function `ExpectedVisitsExtrapolated` in Appendix C.3.1. We fix $K = 393$ and $j = 3$, using 7 extra terms of the series for extrapolation.

[2] Indeed, we will prove this later, cf. Lemma 6.1.

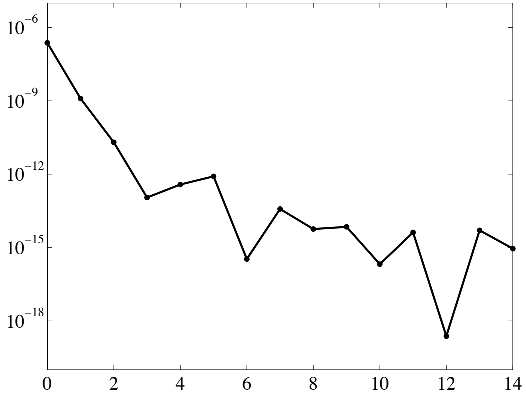

Figure 6.2. *Differences* $|s_{1,2j+2} - s_{1,2j}|$ *of the first row of the extrapolation table vs.* j*, Wynn's epsilon algorithm* ($K = 393, \epsilon = 0.06$)*.*

A Matlab session.

```
>> f=inline('ExpectedVisitsExtrapolated(epsilon,K,extraTerms)-2',...
>>      'epsilon','K','extraTerms');
>> epsilon=fzero(f,[0.06,0.07],optimset('TolX',1e-14),393,7)

epsilon = 6.191395447397203e-002
```

The run takes less than 8 minutes on a 2 GHz PC, a considerable speed-up. The absolute tolerance of the root-finder was set to 10^{-14}, which matches the accuracy of the extrapolation given by our preparatory experiments. As in the $K = 1000$ run, there remains some uncertainty about the 11th and 12th digits. In any case, we have now collected enough evidence to be confident about the correctness of 10 digits.

6.4 Reducing the Complexity by Increasing the Level of Sophistication

In this section we show that simple combinatorics and symbolic calculations reduce the complexity of generating $p_0, \ldots, p_{K-1}$ from $O(K^3)$ to the optimal $O(K)$. This enables us to efficiently use the partial sum E_K to solve the problem.

6.4.1 Using Combinatorics

To begin with, we observe that calculating all the probabilities stored in Π_{2k} is overkill if we are only interested in $p_k = P_{2k}(0,0)$. This is because a walker visiting $(0,0)$ at step $2k$ has followed a very particular type of walk: a combination of two one-dimensional walks. In fact, he has spent $2j$ steps in the east-west direction and $2k - 2j$ steps in the north-south direction. The number of eastbound steps has to be equal to the number of westbound ones, likewise for north-south. Simple combinatorics yields

$$p_k = \sum_{j=0}^{k} \binom{2k}{2j} \cdot \underbrace{\binom{2j}{j} p_{\mathrm{E}}^{j} p_{\mathrm{W}}^{j}}_{\text{east-west}} \cdot \underbrace{\binom{2k-2j}{k-j} p_{\mathrm{N}}^{k-j} p_{\mathrm{S}}^{k-j}}_{\text{north-south}}.$$

The binomial coefficients reflect the possibilities that the walker has: $2j$ steps in the east-west direction out of $2k$ steps altogether, j eastbound steps out of $2j$ steps in the east-west direction, $k-j$ northbound steps out of $2k-2j$ steps in the north-south direction.

The expression for p_k can be simplified further by a little fiddling with binomial coefficients,

$$p_k = \sum_{j=0}^{k} \binom{2k}{k} \binom{k}{j}^2 (p_{\mathrm{E}} p_{\mathrm{W}})^j (p_{\mathrm{N}} p_{\mathrm{S}})^{k-j},$$

a formula that can be found, without much explanation however, in the work of Barnett [Bar63, form. (7)]. We observe that p_k depends on p_{E}, p_{W} only via their product, and the same for p_{N}, p_{S}. Therefore it makes sense to consider their geometric means

$$p_{\mathrm{EW}} = \sqrt{p_{\mathrm{E}} p_{\mathrm{W}}}, \qquad p_{\mathrm{NS}} = \sqrt{p_{\mathrm{N}} p_{\mathrm{S}}}.$$

We obtain[3]

$$p_k = \sum_{j=0}^{k} \binom{2k}{k} \binom{k}{j}^2 p_{\mathrm{EW}}^{2j} p_{\mathrm{NS}}^{2k-2j}. \tag{6.12}$$

Because of the size of the binomial coefficients, using this expression requires some care with machine arithmetic. For instance, at $k = 1000$ the term $\binom{2k}{k}$ would become as large as 10^{600}, which far exceeds the range of IEEE double precision numbers. We have to balance this enormous size with the powers of p_{EW} and p_{NS}. For $p_{\mathrm{EW}} \leqslant p_{\mathrm{NS}}$,[4] which is true for Problem 6, we therefore rewrite (6.12) as

$$p_k = \underbrace{\binom{2k}{k} p_{\mathrm{NS}}^{2k}}_{=a_k} \sum_{j=0}^{k} \underbrace{\binom{k}{j}^2 \left(\frac{p_{\mathrm{EW}}}{p_{\mathrm{NS}}}\right)^{2j}}_{=b_j}.$$

[3] If $p_{\mathrm{EW}} = 0$, this expression simplifies to $p_k = \binom{2k}{k} p_{\mathrm{NS}}^{2k}$ yielding the return probability

$$p = 1 - 1/E = 1 - \sqrt{1 - 4p_{\mathrm{NS}}^2}.$$

Here, the random walk is effectively a biased one-dimensional one in the north-south direction, cf. [Hug95, p. 123]: the nonzero transition probability out of p_{E} or p_{W} just adds to the killing rate. In the setting of Problem 6 we get $p_{\mathrm{EW}} = 0$ for the maximum bias $\epsilon = 1/4$. Hence, because of $p_{\mathrm{NS}} = 1/4$, we obtain $p|_{\epsilon=1/4} = 1 - \sqrt{3}/2 \doteq 0.1339745962$, which is slightly larger than 1/8, cf. Figure 6.1.

[4] Without loss of generality we can generally assume this to be the case. By the symmetry of the problem we would otherwise just change the roles of p_{NS} and p_{EW}.

To be efficient we calculate the coefficients a_k and b_j recursively,

$$a_k = \frac{2k(2k-1)}{k^2} p_{\mathrm{NS}}^2 \cdot a_{k-1}, \quad a_0 = 1;$$

$$b_j = \left(\frac{k-j+1}{j}\right)^2 \frac{p_{\mathrm{EW}}^2}{p_{\mathrm{NS}}^2} \cdot b_{j-1}, \quad b_0 = 1.$$

The resulting algorithm for calculating $p_0, \ldots, p_{K-1}$ is of complexity $O(K^2)$.

A Mathematica session.

```
ExpectedVisits[ϵ_Real, K_] :=
 (pE = 0.25 + ϵ; pW = 0.25 - ϵ; pN = pS = 0.25; w = pE pW; z = pN pS;
   (a = 1) +
    Sum[(a* = (2k (2k - 1))/k^2 z)
      ((b = 1) + Sum[b* = ((k - j + 1)/j)^2 w/z, {j, 1, k}]), {k, 1, K - 1}])
```

```
{K == #,
     ϵ* ==
      (ϵ/.FindRoot[ExpectedVisits[ϵ, #] == 2, {ϵ, 0.06, 0.07},
           WorkingPrecision → MachinePrecision,
           PrecisionGoal → 12])}&/@{125, 250, 500, 1000}//TableForm
```

```
K == 125    ϵ* == 0.061778241155115 28
K == 250    ϵ* == 0.06191252410628889
K == 500    ϵ* == 0.06191395418080741
K == 1000   ϵ* == 0.06191395447399104
```

Mathematica's output confirms the result of the corresponding run using the $O(K^3)$ algorithm in §6.3.1. However, for $K = 1000$ we now need, using a 2 GHz PC, just 45 seconds instead of 2.5 hours. We recall that $K = 1000$ gives us at least 10 correct digits.

6.4.2 Using Symbolic Computation

For the initiated, the form of the expression (6.12) shows that there is a holonomic[5] M-term recurrence for p_k and that its polynomial coefficients can be calculated by Zeilberger's *creative telescoping algorithm*. This algorithm falls into the realm of symbolic hypergeometric summation [Koe98] and allows for the computerized proofs [PWZ96] of many if not most identities of sums involving binomial coefficients. For decades, the derivation and verification of such identities was considered to be

[5] A linear, homogeneous recurrence with polynomial coefficients is called *holonomic*.

a challenging problem. Indeed, a paper coauthored by Zeilberger and presenting his algorithm was entitled “How to do *Monthly* problems with your computer” [NPWZ97].

Let us briefly explain the framework of Zeilberger's algorithm. It applies to sums of the form

$$f_k = \sum_{j\in\mathbb{Z}} F(k,j),$$

where $F(k,j)$ is a *proper hypergeometric term* (cf. [PWZ96, p. 64] or [Koe98, p. 110]), that is essentially a term of the form

$$\begin{aligned} F(k,j) = {} & (\text{polynomial in } k,\ j) \times \\ & \quad (\text{product of binomial coefficients of arguments integer-linearly in } k,\ j) \times \\ & \hfill w^k\, z^j . \end{aligned} \tag{6.13}$$

Then a theorem of Zeilberger [PWZ96, Thm. 6.2.1] states that F satisfies a recurrence of the form

$$\sum_{m=0}^{M} a_m(k) F(k+m,j) = G(k,j+1) - G(k,j),$$

where $a_0, \ldots, a_M$ are polynomials and $G(k,j)/F(k,j)$ is rational in k, j. Moreover, there is an algorithm, cf. [PWZ96, §6.3] or [Koe98, Chap. 7], to construct all these functions. Now, we can sum over all integer values of j and obtain by telescoping the recurrence

$$\sum_{m=0}^{M} a_m(k) f_{k+m} = 0.$$

In the problem at hand, since $\binom{k}{j} = 0$ for $j > k$ or $j < 0$, the probability (6.12) can be written as a sum over the integers $j \in \mathbb{Z}$ of terms that have the required proper hypergeometric form (6.13). Thus, we should be able to calculate such an M-term recurrence for p_k. In fact, Maple ships with the package `sumtools`, which includes the command `sumrecursion` that performs exactly this task. Details and source code of a similar command can be found in the book by the package's author Koepf [Koe98, p. 100].

A Maple session. We use the abbreviations $w = p_{\mathrm{EW}}^2$, $z = p_{\mathrm{NS}}^2$.

```
> with(sumtools):
> sumrecursion(binomial(2*k,k)*binomial(k,j)^2*w^j*z^(k-j),j,p(k));
```

$$4\,(w-z)^2\,(2\,k-1)\,(2\,k-3)\,p\,(k-2) - 2\,(2\,k-1)^2\,(w+z)\,p\,(k-1) + p\,(k)\,k^2$$

Written in full, the Maple output shows that p_k satisfies the three-term recurrence

$$k^2\, p_k \;=\; 2(2k-1)^2\,(p_{\mathrm{EW}}^2 + p_{\mathrm{NS}}^2)\,p_{k-1} \;-\; 4(2k-1)(2k-3)\,(p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2)^2\,p_{k-2}, \tag{6.14}$$

subject to the initial conditions

$$p_0 = 1, \qquad p_1 = 2(p_{\mathrm{EW}}^2 + p_{\mathrm{NS}}^2).$$

Hence, we have obtained an algorithm that allows us to calculate $p_0, \ldots, p_{K-1}$ in optimal complexity $O(K)$.

Now, we are ready to solve the equation $E|_{\epsilon=\epsilon_*} = 2$ for ϵ_* using a software package root-finder. From §6.3.1 we know that the convergence of the series for E is essentially exponential. Therefore, it is reasonable to sum the terms of the series until the value of the sum no longer changes in finite precision arithmetic.

A Mathematica session.

```
ExpectedVisits[ϵ_Real] :=
 (pE = 1/4 + ϵ; pW = 1/4 - ϵ; pN = pS = 1/4; w = pE pW; z = pN pS;
   s = pold = 1; pnew = 2 (w + z); k = 2;
   While[s =! = (s+ = pnew),
    pnew = (-4 (2 k - 1) (2 k - 3) (w - z)^2 pold + 2 (2 k - 1)^2 (w + z) (pold = pnew))/(k++^2)];
   s);
```

```
{Precision == #,
     ϵ* ==
      (ϵ/.FindRoot[ExpectedVisits[ϵ] == 2, {ϵ, 0.06, 0.07},
          WorkingPrecision → #, AccuracyGoal → #])}&/@
  {13, MachinePrecision, 19, 22, 25}//TableForm
```

```
Precision == 13                  ϵ* == 0.06191395447253
Precision == MachinePrecision    ϵ* == 0.0619139544739901
Precision == 19                  ϵ* == 0.06191395447399094135
Precision == 22                  ϵ* == 0.06191395447399094284664
Precision == 25                  ϵ* == 0.06191395447399094284817374
```

We observe that we lose about the last three digits. In machine precision the run takes under a second on a 2 GHz PC and uses 884 terms of the series. With a working precision of 103 digits, and using 7184 terms of the series, it takes less than a minute to get

$$\epsilon_* \doteq 0.061913\,95447\,39909\,42848\,17521\,64732\,12176\,99963\,87749\,98362$$
$$07606\,14672\,58859\,93101\,02975\,96158\,45907\,10564\,57520\,87861,$$

which is correct to the printed 100 digits.

6.5 The Joy of Special Functions

In the last section we have learned that the occupation probabilities p_k are homogeneous polynomials of degree k in p_{EW}^2 and p_{NS}^2 satisfying a three-term recurrence. The odds are good that they are related to one of the canonized families of orthogonal polynomials. Let us see what happens if we ask Maple to evaluate (6.12) symbolically.

A Maple session. We use the abbreviations $w = p_{\mathrm{EW}}^2$, $z = p_{\mathrm{NS}}^2$.

```
> simplify(sum(binomial(2*k,k)*binomial(k,j)^2*z^j*w^(k-j),j=0..k))
    assuming k::integer;
```

$$\binom{2k}{k} \mathit{LegendreP}\left(k, \frac{w+z}{w-z}\right)(w-z)^k$$

We encounter the *Legendre polynomials* P_k. Written in full, the Maple output reads as

$$p_k = \binom{2k}{k} P_k\left(\frac{p_{\mathrm{EW}}^2 + p_{\mathrm{NS}}^2}{p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2}\right)(p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2)^k.$$

Certainly, using this expression instead of (6.14) is computationally of no improvement. We already had arrived at the optimal complexity of $O(K)$ for calculating the probabilities $p_0, \ldots, p_{K-1}$.

However, the Legendre polynomials are useful in deriving precise asymptotic formulas for p_k and, further below, in deriving a closed expression for the expected value E. We recall our observation that doubling K doubles the number of digits to which E_K correctly approximates E; the convergence appears to be roughly exponential. In fact, we are now able to prove this to be true.

Lemma 6.1.

(a) If $\sigma = 2(p_{\mathrm{EW}} + p_{\mathrm{NS}})$, then $\sigma \leqslant 1$, with equality if and only if

$$p_{\mathrm{E}} = p_{\mathrm{W}}, \qquad p_{\mathrm{N}} = p_{\mathrm{S}}, \qquad p_{\mathrm{E}} + p_{\mathrm{W}} + p_{\mathrm{N}} + p_{\mathrm{S}} = 1.$$

(b) If $p_{\mathrm{EW}} \cdot p_{\mathrm{NS}} \neq 0$, there is the asymptotic formula

$$p_k \simeq \frac{\sigma^{2k+1}}{4\pi k\sqrt{p_{\mathrm{EW}}\, p_{\mathrm{NS}}}}; \quad \textit{otherwise} \quad p_k \simeq \frac{\sigma^{2k}}{\sqrt{\pi k}}.$$

(c) The random walk is recurrent, that is $p = 1$, if and only if $\sigma = 1$.

Proof. (a) Using the inequality between geometric and arithmetic means we obtain

$$\sigma = 2p_{\mathrm{EW}} + 2p_{\mathrm{NS}} \leqslant (p_{\mathrm{E}} + p_{\mathrm{W}}) + (p_{\mathrm{N}} + p_{\mathrm{S}}) \leqslant 1.$$

Since equality of the arithmetic and geometric means of two quantities holds only if these quantities are equal, $\sigma = 1$ is characterized as asserted.

(b) Suppose that $p_{\mathrm{EW}} \cdot p_{\mathrm{NS}} \neq 0$. The Laplace–Heine asymptotic formula for the Legendre polynomials [Sze75, Thm. 8.21.1] yields after a short calculation

$$P_k\left(\frac{p_{\mathrm{EW}}^2 + p_{\mathrm{NS}}^2}{p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2}\right)(p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2)^k \simeq \frac{(p_{\mathrm{EW}} + p_{\mathrm{NS}})^{2k+1}}{2\sqrt{\pi k}\,\sqrt{p_{\mathrm{EW}}\, p_{\mathrm{NS}}}}.$$

A multiplication with Stirling's formula $\binom{2k}{k} \simeq 4^k/\sqrt{\pi k}$ proves the assertion.

On the other hand, if $p_{\mathrm{EW}} \cdot p_{\mathrm{NS}} = 0$, we may assume without loss of generality that $p_{\mathrm{EW}} = 0$. By (6.12) we obtain, using Stirling's formula again,

$$p_k = \binom{2k}{k} p_{\mathrm{NS}}^{2k} \simeq \frac{4^k p_{\mathrm{NS}}^{2k}}{\sqrt{\pi k}} = \frac{\sigma^{2k}}{\sqrt{\pi k}}.$$

(c) By d'Alembert's ratio test for the convergence of series with positive terms, the asymptotic formulas in (b) prove that $E = \sum_{k=0}^{\infty} p_k < \infty$ if $\sigma < 1$. On the other hand, if $\sigma = 1$, by the comparison test [Kno56, p. 56] the series $\sum_{k=0}^{\infty} p_k$ inherits the divergence of $\sum_{k=1}^{\infty} 1/k$ or $\sum_{k=1}^{\infty} 1/\sqrt{k}$, respectively.

Summarizing, the series converges, that is the walk is transient, if and only if $\sigma < 1$. Therefore, by (a), recurrence is equivalent to $\sigma = 1$. □

For the particular biasing of Problem 6 we have $\sigma = (1 + \sqrt{1 - 16\epsilon^2})/2$. Here, the walk is recurrent if and only if it is unbiased ($\epsilon = 0$). As promised at the beginning of this chapter, we have thus proven the recurrence of the unbiased random walk in two dimensions, a classic result of Pólya [Pól21].

The Legendre polynomials are also useful if we aim directly at the expected value

$$E = \sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} \binom{2k}{k} P_k\left(\frac{p_{\mathrm{EW}}^2 + p_{\mathrm{NS}}^2}{p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2}\right)(p_{\mathrm{EW}}^2 - p_{\mathrm{NS}}^2)^k.$$

The point is that there is some hope of finding a closed expression for E in terms of special functions, which might be computationally advantageous. For instance, in many compilations of formulas, such as [EMOT53, Vol. 2, §10.10] or [AS84, Table 22.9], one finds the generating functions

$$\sum_{k=0}^{\infty} P_k(x)\, z^k = \frac{1}{\sqrt{1 - 2xz + z^2}}, \qquad \sum_{k=0}^{\infty} \frac{1}{k!} P_k(x)\, z^k = e^{xz} J_0(z\sqrt{1 - x^2}) \qquad (6.15)$$

with J_0 being the Bessel function of the first kind of zero order. A systematic way to derive such expressions was communicated to us by Herbert Wilf: one simply plugs Laplace's first integral for the Legendre polynomials, cf. [WW96, §15.23] or [Rai60, Chap. 10, §97],

$$P_k(x) = \frac{1}{\pi}\int_0^{\pi} (x + \sqrt{x^2 - 1}\cos\theta)^k \, d\theta, \qquad x \in \mathbb{C},$$

into the series at hand and changes the order of summation and integration. If the power series $\sum_{k=0}^{\infty} a_k z^k$ is a known function $f(z)$, one gets

$$\sum_{k=0}^{\infty} a_k P_k(x) z^k = \frac{1}{\pi}\int_0^{\pi} \sum_{k=0}^{\infty} a_k (x + \sqrt{x^2-1}\cos\theta)^k z^k \, d\theta$$
$$= \frac{1}{\pi}\int_0^{\pi} f\left((x+\sqrt{x^2-1}\cos\theta)z\right) d\theta.$$

The integral form might be advantegeous because there are more methods known for the closed evaluation of integrals than of sums. We leave the actual calculations to a computer algebra system such as *Mathematica* and just assist the symbolic integration by restricting the possible values of p_{EW} and p_{NS}:

$$0 \leqslant p_{\mathrm{EW}} \leqslant p_{\mathrm{NS}}, \qquad \sigma = 2(p_{\mathrm{EW}} + p_{\mathrm{NS}}) < 1.$$

Because of the symmetry of the problem, the first inequality is no loss of generality. By Lemma 6.1 we know that the second inequality is equivalent to $E < \infty$.

A Mathematica session.

$$\mathtt{LegendreKernel[x\_,\ k\_]\ :=\ \frac{1}{\pi}\left(x + \sqrt{x^2-1}\,Cos[\theta]\right)^{k};}$$

$$\mathtt{Integrate\Big[}$$
$$\mathtt{\quad Sum\Big[Binomial[2k,\ k]\,LegendreKernel\Big[\frac{p_{EW}{}^2 + p_{NS}{}^2}{p_{EW}{}^2 - p_{NS}{}^2},\ k\Big]\left(p_{EW}{}^2 - p_{NS}{}^2\right)^{k},}$$
$$\mathtt{\quad\ \{k,\ 0,\ \infty\}\Big],\ \{\theta,\ 0,\ \pi\},\ Assumptions \to \{0 \le p_{EW} \le p_{NS},\ 2\,(p_{EW} + p_{NS})\ <\ 1\}\Big]}$$

$$\frac{\mathtt{2\,EllipticK\Big[-\frac{16\,p_{EW}\,p_{NS}}{-1+4\,(p_{EW}-p_{NS})^{2}}\Big]}}{\mathtt{\pi\sqrt{1-4\,(p_{EW}-p_{NS})^{2}}}}$$

The reader must be alert to the fact that *Mathematica* uses the complete elliptic integral of the first kind

$$K(k) = \int_0^1 \frac{dt}{\sqrt{1-t^2}\,\sqrt{1-k^2t^2}}$$

not as a function of the modulus k, but of the parameter $m = k^2$ instead. Summarizing, we have obtained the closed-form expression

$$E = \frac{2}{\pi\sqrt{1-4(p_{\mathrm{EW}}-p_{\mathrm{NS}})^2}}\, K\left(\frac{4\sqrt{p_{\mathrm{EW}}p_{\mathrm{NS}}}}{\sqrt{1-4(p_{\mathrm{EW}}-p_{\mathrm{NS}})^2}}\right), \tag{6.16}$$

a result that was discovered in the early 1960s independently by Henze [Hen61, form. (3.3)] and Barnett [Bar63, form. (43)].[6] The advantage of K being involved is that it can be evaluated [BB87, Alg. 1.2(a)] exceedingly fast with the help of the *arithmetic-geometric mean* $M(a, b)$ of Gauss,

$$K\left(\sqrt{1-k^2}\right) = \frac{\pi}{2\,M(1,k)}. \tag{6.17}$$

The reader will find more details on M in §6.5.1. From (6.17) we infer the final formula of this section

$$E = 1/M\left(\sqrt{1-4(p_{\mathrm{EW}}+p_{\mathrm{NS}})^2}, \sqrt{1-4(p_{\mathrm{EW}}-p_{\mathrm{NS}})^2}\right), \tag{6.18}$$

which we now use to solve Problem 6.

A Mathematica session.

```
ExpectedVisits[ϵ_] :=
  1/ArithmeticGeometricMean[√(1 - 4 (p_EW + p_NS)^2), √(1 - 4 (p_EW - p_NS)^2)]/.
   {p_EW → √((1/4 + ϵ) (1/4 - ϵ)), p_NS → 1/4}
```

```
{Precision == #,
       ϵ_* ==
         (ϵ/.FindRoot[ExpectedVisits[ϵ] == 2, {ϵ, 0.06, 0.07},
              WorkingPrecision → #, AccuracyGoal → #])}&/@
    {13, MachinePrecision, 19, 22, 25}//TableForm
```

```
Precision == 13                  ϵ_* == 0.06191395447402
Precision == MachinePrecision    ϵ_* == 0.06191395447399095
Precision == 19                  ϵ_* == 0.0619139544739909428 7
Precision == 22                  ϵ_* == 0.0619139544739909428482 0
Precision == 25                  ϵ_* == 0.0619139544739909428481751 9
```

All but about the last two digits are correct. The code is so fast that using it with a working precision of 10 010 digits takes less than a second (on a 2 GHz PC) to get 10 000 correct digits.

[6] One motivation of Barnett was to characterize recurrent biased random walks, that is $E = \infty$. From (6.16) he could infer that $E = \infty$ if and only if the argument of K equals 1,

$$\frac{4\sqrt{p_{\mathrm{EW}}p_{\mathrm{NS}}}}{\sqrt{1-4(p_{\mathrm{EW}}-p_{\mathrm{NS}})^2}} = \frac{4\sqrt{p_{\mathrm{EW}}p_{\mathrm{NS}}}}{\sqrt{1-\sigma^2+16p_{\mathrm{EW}}p_{\mathrm{NS}}}} = 1.$$

This is the case if and only if $\sigma = 1$, which is consistent with Lemma 6.1.

6.5.1 Using Interval Arithmetic

The arithmetic-geometric mean $M(a, b)$ of nonnegative numbers a and b is defined as the common limit of the two quadratically convergent sequences that are built by taking successively the geometric and arithmetic means,

$$M(a,b) = \lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n; \quad a_0 = a,\ b_0 = b,\ a_{n+1} = \sqrt{a_n b_n},\ b_{n+1} = \frac{a_n + b_n}{2}.$$

It enjoys many monotonicity properties that make it particularly well suited for interval calculations. First, it is easy to check for $0 \leqslant a_* \leqslant a \leqslant a^*$, $0 \leqslant b_* \leqslant b \leqslant b^*$ that

$$M(a_*, b_*) \leqslant M(a, b) \leqslant M(a^*, b^*).$$

It follows that

$$M([a,b],[c,d]) = [M(a,c), M(b,d)], \qquad 0 \leqslant a \leqslant b,\ 0 \leqslant c \leqslant d.$$

Next, the above iteration is always enclosing [BB87, §1.1],

$$a_n \leqslant a_{n+1} \leqslant M(a,b) \leqslant b_{n+1} \leqslant b_n, \qquad n \geqslant 1.$$

Thus, using downward rounding for a_n and upward rounding for b_n yields a straightforward, tight, and rigorous implementation using an interval arithmetic package such as Intlab. The reader will find the Intlab function `AGM` in Appendix C.4.3.

For Problem 6, appropriate interval root-finding can be based on the interval-bisection method. The Intlab procedure `IntervalBisection` of Appendix C.4.3 is coded essentially in analogy to the interval-minimization algorithm of §4.3.

A Matlab/Intlab session.

```
>> E=inline('1/AGM(sqrt(1-4*(pEW+pNS)^2),sqrt(1-4*(pEW-pNS)^2))',...
>>     'pEW','pNS');
>> f=inline('E(sqrt((0.25+epsilon)*(0.25-epsilon)),0.25)-2',...
>>     'epsilon','E');
>> epsilon=IntervalBisection(f,infsup(0.06,0.07),1e-15,E)

intval epsilon = [  6.191395447399049e-002,  6.191395447399144e-002]
```

Just using IEEE machine arithmetic, the run takes less than 4 seconds on a 2 GHz PC and validates the correctness of 13 digits,

$$\epsilon_* \doteq 0.06191395447399.$$

For higher precision we use *Mathematica*. Here, no directive for directional rounding is accessible to the user and a rigorous implementation of the arithmetic-geometric mean has to use interval arithmetic for the iteration itself. In machine arithmetic the resulting code is slightly less efficient and less tight than the Intlab implementation. The reader will find an overload[7] of the command `ArithmeticGeometricMean` in Appendix C.5.3 and the code `IntervalBisection` in Appendix C.5.3.

As discussed in §4.5, there is no need to use the interval method from scratch. Instead we start using our favorite method to obtain an approximation of ϵ_* that is very likely correct to the requested number of digits. After inflation of this approximation to a small interval, we apply the interval bisection-method and obtain a validated enclosure.

A Mathematica session. *(cont. of session on p. 139)*[8]

```
prec = 10010;
ϵTry = ϵ/.FindRoot[ExpectedVisits[ϵ] - 2, {ϵ, 0.06, 0.07},
     WorkingPrecision → prec, AccuracyGoal → prec];
ϵInterval = IntervalBisection[ExpectedVisits[#] - 2&, ϵTry + {-1, 1}10^(-prec+5),
    10^(-prec)];
DigitsAgreeCount[ϵInterval]
10006
```

This technique of approximation first, validation last shows the correctness of 10 006 digits in less than a minute computing time on a 2 GHz PC.

6.6 Using Fourier Analysis

In this final section we follow a different approach to calculating E. Instead of looking at the expected number of visits to just the starting site, we consider it for *all* sites at once. That is, generalizing (6.7) we introduce the expected number $E(x, y)$ of visits to the site (x, y),

$$E(x,y) = \sum_{k=0}^{\infty} P_k(x,y), \qquad E = E(0,0).$$

Now, summing the partial difference equation (6.8) over all k yields a partial difference equation for $E(\cdot,\cdot)$,

$$\begin{aligned} E(x,y) = P_0(x,y) \;+\; p_{\mathrm{E}}\, E(x-1,y) + p_{\mathrm{W}}\, E(x+1,y) \\ + p_{\mathrm{N}}\, E(x,y-1) + p_{\mathrm{S}}\, E(x,y+1). \end{aligned}$$

Because of the initial condition (6.9), written with the Kronecker δ symbol as $P_0(x,y) = \delta_{x,0}\delta_{y,0}$, we obtain without any further reference to probabilities

$$\begin{aligned} E(x,y) = \delta_{x,0}\delta_{y,0} \;+\; p_{\mathrm{E}}\, E(x-1,y) + p_{\mathrm{W}}\, E(x+1,y) \\ + p_{\mathrm{N}}\, E(x,y-1) + p_{\mathrm{S}}\, E(x,y+1). \qquad (6.19) \end{aligned}$$

[7] This way we can reuse the function `ExpectedVisits` from page 139 for interval arguments.

[8] The command `DigitsAgreeCount` can be found in Appendix C.5.1.

In analogy with partial differential equations, the solution of this difference equation is called the *lattice Green function* of the random walk [Hug95, p. 132].

Concerning the existence and construction of the lattice Green function one encounters some technical difficulties of convergence for an immortal walker, that is, if $p_{\text{kill}} = 1 - p_{\text{E}} - p_{\text{W}} - p_{\text{N}} - p_{\text{S}} = 0$. We will avoid them by temporarily assuming $p_{\text{kill}} > 0$. As soon as it is safe to take the limit $p_{\text{kill}} \to 0$, we will drop this assumption. By continuity, formulas will then be correct for $p_{\text{kill}} = 0$ also.

A convenient way to solve the linear partial difference equation (6.19) is to use Fourier series,

$$\hat{E}(\phi, \theta) = \sum_{(x,y)\in\mathbb{Z}^2} E(x, y) e^{ix\phi} e^{iy\theta}. \tag{6.20}$$

Multiplying (6.19) by $e^{ix\phi}e^{iy\theta}$ and summing over all integer values of x and y yields

$$\hat{E}(\phi, \theta) = 1 + \underbrace{\left(p_{\text{E}}e^{i\phi} + p_{\text{W}}e^{-i\phi} + p_{\text{N}}e^{i\theta} + p_{\text{S}}e^{-i\theta}\right)}_{=\lambda(\phi,\theta)} \hat{E}(\phi, \theta),$$

where $\lambda(\phi, \theta)$ is called the *structure function* or *symbol* of the random walk [Hug95, form. (3.116)]. This equation for $\hat{E}$ is readily solved by

$$\hat{E}(\phi, \theta) = \frac{1}{1 - \lambda(\phi, \theta)} = \frac{1}{1 - p_{\text{E}}e^{i\phi} - p_{\text{W}}e^{-i\phi} - p_{\text{N}}e^{i\theta} - p_{\text{S}}e^{-i\theta}}.$$

Up to here, our arguments were purely formal. Now, by our temporary assumption $p_{\text{kill}} > 0$ we see that

$$\sum_{(x,y)\in\mathbb{Z}^2} |E(x, y)| = \sum_{(x,y)\in\mathbb{Z}^2} E(x, y) = \hat{E}(0, 0) = p_{\text{kill}}^{-1} < \infty.$$

Therefore, the Fourier series (6.20) converges absolutely and the coefficients $E(x, y)$ are established as the solution of the difference equation.

The value of interest, $E = E(0, 0)$, is obtained by an inverse Fourier transformation,

$$E = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{d\phi \, d\theta}{1 - p_{\text{E}}e^{i\phi} - p_{\text{W}}e^{-i\phi} - p_{\text{N}}e^{i\theta} - p_{\text{S}}e^{-i\theta}}. \tag{6.21}$$

Numerical quadrature for this double integral becomes less involved if we avoid the use of complex numbers. To this end we make use of a trick that is provided by the following lemma.

Lemma 6.2. *Let $f(z)$ be a function that is analytic in the disk $|z| < R$. For $a, b > 0$ with $a + b < R$ the integral*

$$I(a, b) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(ae^{ix} + be^{-ix}) \, dx$$

is well defined. Moreover, the following transformation is valid:

$$I(a, b) = I(\sqrt{ab}, \sqrt{ab}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(2\sqrt{ab} \cos x) \, dx.$$

Proof. Note that $|ae^{ix} + be^{-ix}| \leqslant a + b < R$ for $x \in [-\pi, \pi]$ and that $2\sqrt{ab} \leqslant a + b < R$. Let $\mathcal{C}$ be the positively oriented unit circle. We have

$$I(a,b) = \frac{1}{2\pi i} \int_{\mathcal{C}} f(az + b/z)\, \frac{dz}{z}.$$

The substitution $z = \sqrt{b/a} \cdot w$ and Cauchy's integral theorem show that

$$\begin{aligned} I(a,b) &= \frac{1}{2\pi i} \int_{\sqrt{a/b}\,\mathcal{C}} f(\sqrt{ab}\, w + \sqrt{ab}/w)\, \frac{dw}{w} \\ &= \frac{1}{2\pi i} \int_{\mathcal{C}} f(\sqrt{ab}\, w + \sqrt{ab}/w)\, \frac{dw}{w} = I(\sqrt{ab}, \sqrt{ab}). \end{aligned}$$

The observation $\sqrt{ab}\, e^{ix} + \sqrt{ab}\, e^{-ix} = 2\sqrt{ab} \cos x$ finishes the proof. □

If we apply this Lemma to (6.21) twice, which is possible by our temporary assumption $p_{\text{kill}} > 0$, we obtain by using the symmetry of the cosine

$$\begin{aligned} E &= \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{d\phi\, d\theta}{1 - 2p_{\text{EW}} \cos\phi - 2p_{\text{NS}} \cos\theta} \\ &= \frac{1}{\pi^2} \int_{0}^{\pi} \int_{0}^{\pi} \frac{d\phi\, d\theta}{1 - 2p_{\text{EW}} \cos\phi - 2p_{\text{NS}} \cos\theta}. \end{aligned} \tag{6.22}$$

Now, since

$$|2p_{\text{EW}} \cos\phi + 2p_{\text{NS}} \cos\theta| \leqslant 2p_{\text{EW}} + 2p_{\text{NS}} = \sigma$$

the integral expression (6.22) is well defined as long as $\sigma < 1$. Therefore from now on we may drop the assumption $p_{\text{kill}} > 0$ and replace it by the transience condition $\sigma < 1$, cf. Lemma 6.1. As we have already noted following that lemma, the specific transition probabilities of Problem 6 yield that $\sigma < 1$ is equivalent to the condition $0 < \epsilon \leqslant 1/4$ on the bias.

Using adaptive numerical quadrature, expression (6.22) can readily be used to solve Problem 6, that is, to solve the equation $E|_{\epsilon=\epsilon_*} = 2$ for ϵ_*.

A Matlab session.

```
>> g=inline(...
    ’1./(1-2*sqrt((1/4+ep)*(1/4-ep))*cos(phi)-cos(theta)/2)/pi^2’,...
    ’phi’,’theta’,’ep’);
>> f=inline(’dblquad(g,0,pi,0,pi,1e-13,@quadl,ep)-2’,’ep’,’g’);
>> epsilon=fzero(f,[0.06,0.07],optimset(’TolX’,1e-16),g)

epsilon = 6.191395447399090e-002
```

The parameters are chosen so that the code should deliver at least 13 correct digits. In fact, 15 digits are correct. The run time on a 2 GHz PC is 8 minutes.

On comparing the two expressions (6.16) and (6.22) for E, we have obtained

$$\frac{1}{\pi^2}\int_0^\pi\int_0^\pi \frac{d\phi\,d\theta}{1-a\cos\phi-b\cos\theta} = \frac{2}{\pi\sqrt{1-(a-b)^2}}\,K\left(\frac{2\sqrt{ab}}{\sqrt{1-(a-b)^2}}\right), \tag{6.23}$$

assuming $a \geqslant 0$, $b \geqslant 0$, and $a+b<1$. Given all the success of symbolic computation in this chapter so far, it is amusing to note that neither *Mathematica* nor Maple is currently able to deliver this formula. We therefore cannot refrain from showing the reader how to get it directly by hand. From the straightforward fact [PBM86, form. (1.5.9.15)]

$$\frac{1}{\pi}\int_0^\pi \frac{d\theta}{c+d\cos\theta} = \frac{1}{\sqrt{c^2-d^2}}, \qquad c>|d|,$$

which is certainly no problem for any of the computer algebra systems, we infer that

$$\begin{aligned}\frac{1}{\pi^2}\int_0^\pi\int_0^\pi \frac{d\phi\,d\theta}{1-a\cos\phi-b\cos\theta} &= \frac{1}{\pi}\int_0^\pi \frac{d\phi}{\sqrt{(1-a\cos\phi)^2-b^2}}\\ &= \frac{1}{\pi}\int_{-1}^1 \frac{ds}{\sqrt{(1-s^2)((1-as)^2-b^2)}} = \int_{-1}^1 \frac{ds}{\sqrt{\text{quartic polynomial in } s}}.\end{aligned}$$

We recognize the last expression as an elliptic integral of the first kind. It can be brought to canonical form by the substitution

$$t=\sqrt{\frac{(1+b-a)(1+s)}{2(1+b-a\,s)}},$$

cf. [BF71, form. (252.00)], which finally yields the desired result

$$\begin{aligned}\frac{1}{\pi}\int_{-1}^1 &\frac{ds}{\sqrt{(1-s^2)((1-as)^2-b^2)}}\\ &= \frac{2}{\pi\sqrt{1-(a-b)^2}}\int_0^1 \frac{dt}{\sqrt{1-t^2}\sqrt{1-\frac{4ab}{1-(a-b)^2}\,t^2}}\\ &= \frac{2}{\pi\sqrt{1-(a-b)^2}}\,K\left(\frac{2\sqrt{ab}}{\sqrt{1-(a-b)^2}}\right).\end{aligned}$$

6.7 Harder Problems

For two-dimensional random walks on the square lattice, E is expressible in terms of the arithmetic-geometric mean (6.18), and so there is no conceivable question about return probabilities that would increase the computational difficulty beyond that of Problem 6. This changes if we consider other lattices, say triangular two-dimensional, cubic three-dimensional, or hypercubic d-dimensional ones.

Cubic Lattices. For cubic lattices the techniques of §§6.2–6.4 and §6.6 are still applicable, though the computational effort will increase considerably. However, we leave it as a challenge to the reader to obtain for the general biased random walk on cubic lattices such nice formulas using special functions as in §6.5—if there are any at all.

To highlight the difficulty that will be encountered, let us review what is known for the random walk on the simple cubic lattice. In the unbiased case, that is, if all the next-neighbor transition probabilities are equal to 1/6, one immediately generalizes (6.22) to the triple integral

$$E = \frac{1}{\pi^3}\int_0^\pi\int_0^\pi\int_0^\pi \frac{d\phi_1\,d\phi_2\,d\phi_3}{1-(\cos\phi_1+\cos\phi_2+\cos\phi_3)/3}.$$

Watson proved in 1939, in a famous *tour de force* [Wat39], the result[9]

$$E = \frac{12}{\pi^2}(18+12\sqrt{2}-10\sqrt{3}-7\sqrt{6})\,K^2(k_6), \qquad k_6 = (2-\sqrt{3})(\sqrt{3}-\sqrt{2}).$$

In 1977 Glasser and Zucker [GZ77] gave the formula[10]

$$E = \frac{\sqrt{6}}{32\pi^3}\,\Gamma(\tfrac{1}{24})\Gamma(\tfrac{5}{24})\Gamma(\tfrac{7}{24})\Gamma(\tfrac{11}{24}),$$

which by [BZ92, Table 3(vii)] can be simplified to

$$E = \frac{\sqrt{3}-1}{32\pi^3}\,\Gamma^2(\tfrac{1}{24})\Gamma^2(\tfrac{11}{24}).$$

The two formulas involving the gamma function may look more pleasing and elementary than Watson's. However, from a computational perspective the latter is still the best. For instance, to get 1000 digits the formulas involving the gamma function require about 16 seconds of CPU time, while Watson's does it in under a tenth of a second; this is because of the cost of evaluating the gamma function to high precision (see §5.7). In fact, Borwein and Zucker suggested in 1992 to use in turn such relations between the gamma function and the complete elliptic integral of the first kind for a fast evaluation of the gamma function at particular rational arguments, cf. [BZ92].

In the general biased case, let us—in addition to the two-dimensional transition probabilities p_E, p_W, p_N, and p_S—denote the probabilities for a step upwards or downwards in the third dimension by p_U and p_D, respectively. Analogously to p_EW and p_NS we use the abbreviation $p_\mathrm{UD} = \sqrt{p_\mathrm{U}p_\mathrm{D}}$. Now, formula (6.22) is readily generalized to

$$E = \frac{1}{\pi^3}\int_0^\pi\int_0^\pi\int_0^\pi \frac{d\phi_1\,d\phi_2\,d\phi_3}{1-2p_\mathrm{EW}\cos\phi_1-2p_\mathrm{NS}\cos\phi_2-2p_\mathrm{UD}\cos\phi_3}.$$

[9] In particular we observe that $E<\infty$. Therefore, unlike in one or two dimensions, the unbiased random walk in three dimensions is transient ($p<1$), a classic fact already known to Pólya in 1921 [Pól21]. Watson evaluated his expression as $E/3 \doteq 0.5054620197$, cf. [Wat39, p. 267], which gives the return probability $p = 1-1/E \doteq 0.3405373295$ — a value correct to the 10 digits given.

[10] Glasser communicated to Hughes [Hug95, p. 614] that "the expression reported in this paper is spoiled by the accidental omission in transcription of a factor of 384π".

Using (6.23) we obtain[11]

$$E = \frac{2}{\pi^2} \int_0^\pi \frac{K(k)\, d\phi}{\sqrt{(1-2p_{\mathrm{UD}}\cos\phi)^2 - 4(p_{\mathrm{EW}} - p_{\mathrm{NS}})^2}},$$

$$k = \frac{4\sqrt{p_{\mathrm{EW}}\, p_{\mathrm{NS}}}}{\sqrt{(1-2p_{\mathrm{UD}}\cos\phi)^2 - 4(p_{\mathrm{EW}} - p_{\mathrm{NS}})^2}},$$

a formula that can be straightforwardly evaluated numerically. Evaluating it symbolically is, however, still an open problem. Recently, the particular case $p_{\mathrm{EW}} = p_{\mathrm{NS}}$ was solved in another *tour de force* by Delves and Joyce [DJ01], later simplified in collaboration with Zucker [JDZ03]: their result [DJ01, form. (5.13)], resp. [JDZ03, form. (4.26)], reads in the notation used so far as

$$E|_{p_{\mathrm{EW}}=p_{\mathrm{NS}}} = \frac{8\, K(k_+)\, K(k_-)}{\pi^2\left(\sqrt{1-4(p_{\mathrm{UD}}-2p_{\mathrm{EW}})^2} + \sqrt{1-4(p_{\mathrm{UD}}+2p_{\mathrm{EW}})^2}\,\right)},$$

with the moduli

$$k_\pm^2 = \frac{1}{2}\left(1 - \frac{\sqrt{(1-2p_{\mathrm{UD}})^2 - 16p_{\mathrm{EW}}^2} + \sqrt{(1+2p_{\mathrm{UD}})^2 - 16p_{\mathrm{EW}}^2}}{\left(\sqrt{1-4(p_{\mathrm{UD}}-2p_{\mathrm{EW}})^2} + \sqrt{1-4(p_{\mathrm{UD}}+2p_{\mathrm{EW}})^2}\,\right)^3}\right.$$
$$\left.\cdot\left(\sqrt{1-4p_{\mathrm{UD}}^2}\left(\sqrt{(1-4p_{\mathrm{EW}})^2 - 4p_{\mathrm{UD}}^2} + \sqrt{(1+4p_{\mathrm{EW}})^2 - 4p_{\mathrm{UD}}^2}\right)^2 \pm 64p_{\mathrm{EW}}^2\right)\right).$$

What a triumph of dedicated men; for such problems current computer algebra systems are of little help.

Hypercubic Lattices. The extension to higher dimensions requires further thinking. None of the methods presented so far directly generalizes to a really efficient numerical method for calculating E in dimensions $d \gg 3$. However, the result of §6.6 can be transformed appropriately to address biased random walks on the *hypercubic lattice* in d dimensions, that is, on the integer lattice $\mathbb{Z}^d$.

We denote the transition probabilities for a step forward or backward in the j-th dimension by p_j^+ and p_j^-, respectively, and their geometric mean by

$$p_j^* = \sqrt{p_j^+ p_j^-}.$$

Just like in Lemma 6.1 we can prove that $0 \leqslant \sigma = 2(p_1^* + \ldots + p_d^*) \leqslant 1$. Formula (6.22) is readily generalized to

$$E = \frac{1}{\pi^d} \int_0^\pi \cdots \int_0^\pi \frac{d\phi_1 \ldots d\phi_d}{1 - 2p_1^*\cos\phi_1 - \ldots - 2p_d^*\cos\phi_d}. \tag{6.24}$$

[11] This result and the one that follows were brought to our attention by John Boersma, who learned them from John Zucker.

In higher dimensions numerical integration of this formula is prohibitively expensive. However, there is now a neat trick (see [Mon56, §2]) that transforms the d-dimensional integral to a single integral of a product of modified Bessel functions: we insert the integral

$$\frac{1}{1 - 2p_1^* \cos\phi_1 - \ldots - 2p_d^* \cos\phi_d} = \int_0^\infty e^{-t(1-2p_1^* \cos\phi_1 - \ldots - 2p_d^* \cos\phi_d)}\, dt$$

into (6.24), interchange orders of integration, use the representation [Olv74, p. 82]

$$I_0(x) = \frac{1}{\pi} \int_0^\pi e^{x\cos\phi}\, d\phi$$

of the modified Bessel function I_0 of the first kind of order zero, and deduce that[12]

$$E = \int_0^\infty e^{-t}\, I_0(2p_1^*\, t) \cdot \ldots \cdot I_0(2p_d^*\, t)\, dt. \tag{6.25}$$

Since $I_0(x)$ grows like (see [Olv74, p. 83])

$$I_0(x) \sim \frac{e^x}{\sqrt{2\pi x}} \qquad \text{as} \quad x \to \infty,$$

the integrand in (6.25) decays like $O(e^{-(1-\sigma)t} t^{-d/2})$ as $t \to \infty$. In particular, we observe that $E < \infty$ for $d \geqslant 3$,[13] that is, the biased random walk is always *transient* in dimensions higher than two.

The efficient numerical evaluation of the integral (6.25) can be based on a double-exponential quadrature formula (see §§3.6.1 and 9.4). However, because of the exponential growth of $I_0(x)$ we have to exercise some care to avoid overflow. It is advisable to perform the calculation using the scaled function $\tilde{I}_0(x) = e^{-x} I_0(x)$, $x \geqslant 0$, and the expression

$$E = \int_0^\infty e^{-(1-\sigma)t}\, \tilde{I}_0(2p_1^*\, t) \cdot \ldots \cdot \tilde{I}_0(2p_d^*\, t)\, dt.$$

In Matlab the scaled function $\tilde{I}_0(x)$ can be evaluated with the built-in command `BesselI(0,x,1)`. A *Mathematica* implementation `BesselITilde[x]` can be found at the web page for this book. It is based on an asymptotic expansion of $I_0(x)$ with rigorous bounds [Olv74, p. 269]: for $x > 0$

$$I_0(x) = \frac{e^x}{\sqrt{2\pi x}} \left(\sum_{k=0}^{n-1} \frac{(2k-1)!!^2}{k!\,(8x)^k} + R_n \right), \qquad |R_n| \leqslant 2e^{1/4x} \frac{(2n-1)!!^2}{n!\,(8x)^n}.$$

We can use this approach to solve Problem 6 once more:

[12] This generalizes the result [Mon56, form. (2.11)] that Montroll had obtained in 1956 for the particular case $p_j^+ = p_j^-$, $j = 1, \ldots, d$.

[13] And for $d \leqslant 2$, if and only if $\sigma < 1$; in agreement with Lemma 6.1.

Table 6.1. *Probability p of return for the d-dimensional hypercubic lattice.*

d	p	d	p	d	p
3	0.340537329550999	10	0.056197535974268	17	0.031352140397027
4	0.193201673224984	11	0.050455159821331	18	0.029496289133281
5	0.135178609820655	12	0.045789120900621	19	0.027848522338807
6	0.104715495628822	13	0.041919897078975	20	0.026375598694496
7	0.085844934113379	14	0.038657877090674	30	0.017257643569441
8	0.072912649959384	15	0.035869623125357	40	0.012827098305686
9	0.063447749652725	16	0.033458364465789	100	0.005050897571251

A Mathematica session.

```
ExpectedVisits[ϵ_Real] := (pE = 1/4 + ϵ; pW = 1/4 - ϵ;
   pN = pS = 1/4; pEW = √(pE pW); pNS = √(pN pS); σ = 2 (pEW + pNS);
   NIntegrate[Exp[-(1 - σ) t] BesselITilde[2 pEW t] BesselITilde[2 pNS t],
    {t, 0, ∞}, PrecisionGoal → 15, Method → DoubleExponential]);
FindRoot[ExpectedVisits[ϵ] == 2, {ϵ, 0.06, 0.07}, AccuracyGoal → 15]

{ϵ → 0.0619139544739909 7}
```

The run, in IEEE arithmetic, takes about one second on a 2 GHz PC and yields 15 correct digits.

Finally, we apply the approach to higher-dimensional problems and calculate, for various dimensions d, the probability of return, $p = 1 - 1/E$, of the *unbiased* random walk, that is, for the specific transition probabilities $p_j^* = 1/2d$, $j = 1, \ldots, d$.

A Mathematica session. *(Generates Table 6.1.)*

```
ReturnProbability[d_Integer] :=
 1 - 1 / NIntegrate[BesselITilde[t / d]^d, {t, 0, ∞}, WorkingPrecision → 20,
    PrecisionGoal → 17, Method → DoubleExponential]
{#, NumberForm[ReturnProbability[#], {15, 15}]} & /@
  Join[Range[3, 20], {30, 40, 100}] // TableForm
```

The data of Table 6.1 have previously been calculated by Griffin [Gri90, Table 4] (differing in the last digit for $d = 11$ and $d = 12$, however). He uses a dimensional recursion that generalizes the combinatorial method of §6.4.1, and accelerates the convergence of the underlying series by Aitken's Δ^2-method.