been developed. We first discuss a relatively simple algorithm proposed by Rosenbluth and Rosenbluth in which each walk of N steps is associated with a weighting function $w(N)$. Because the first step to the north is always possible, we have $w(1) = 1$. In order that all allowed configurations of a given N are counted equally, the weights $w(N)$ for $N > 1$ are determined according to the following possibilities:

1. All three possible steps violate the self-intersection constraint (see Figure 7.7b). The walk is terminated with a weight $w(N) = 0$, and a new walk is generated at the origin.
2. All three steps are possible and $w(N) = w(N-1)$.
3. Only m steps are possible with $1 \leq m < 3$ (see Figure 7.7c). In this case $w(N) = (m/3)w(N-1)$, and one of the m possible steps is chosen at random.

The desired unbiased value of $\langle R^2 \rangle$ is obtained by weighting R_i^2, the value of R^2 obtained in the ith trial, by the value of $w_i(N)$, the weight found for this trial. Hence, we write

$$\langle R^2 \rangle = \frac{\sum_i w_i(N) R_i^2}{\sum_i w_i(N)}, \tag{7.47}$$

where the sum is over all trials.

Problem 7.29 Rosenbluth and Rosenbluth enrichment method
Incorporate the Rosenbluth method into your Monte Carlo program and compute R^2 for $N = 4, 8, 16$, and 32. Estimate the exponent ν from a log-log plot of R^2 versus N. Can you distinguish your estimate for ν from its random walk value $\nu = 1/2$? ■

The Rosenbluth and Rosenbluth procedure is not particularly efficient because many walks still terminate, and thus we do not obtain many walkers for large N. Grassberger improved this algorithm by increasing the population of walkers with high weights and reducing the population of walkers with low weights. The idea is that if $w(N)$ for a given trial is above a certain threshold, we add a new walker and give the new and old walker half of the original weight. If $w(N)$ is below a certain threshold, then we eliminate half of the walkers with weights below this threshold (for example, every second walker) and double the weights of the remaining half. It is a good idea to adjust the thresholds as the simulation runs in order to maintain a relatively constant number of walkers.

More recently Prellberg and Krawczyk further improved the Rosenbluth and Rosenbluth enrichment method so that there is no need to provide a threshold value. After each step the average weight of the walkers $\langle w(N) \rangle$ is computed for a given trial, and the ratio $r = w(N)/\langle w(N) \rangle$ is used to determine whether to add walkers (enrichment) or eliminate walkers (pruning). If $r > 1$, then $c = \min(r, m)$ copies of the walker are made each with weight $w(N)/c$. If $r < 1$, then remove this walker with probability $1 - r$. This algorithm leads to an approximately constant number of walkers and is related to the Wang–Landau method which we will discuss in Problem 15.30.

Another enrichment algorithm is the "reptation" method (see Wall and Mandel). For simplicity, consider a model polymer chain in which all bond angles are ±90°. As an example of this model, the five independent $N = 5$ polymer chains are shown in Figure 7.8. (Other chains differ only by a rotation or a reflection.) The reptation method can be stated as follows:

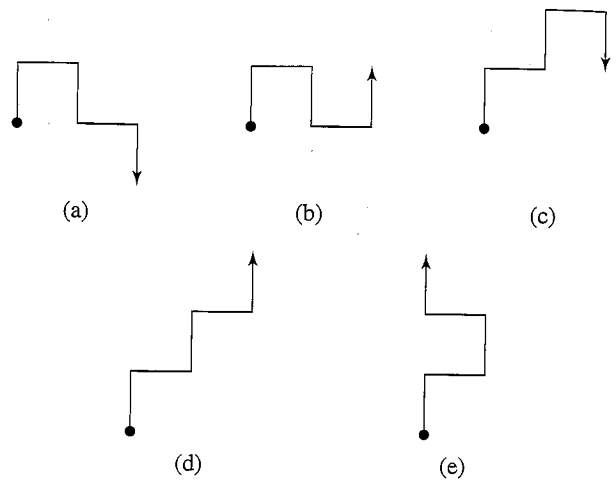

Figure 7.8 The five independent possible walks of $N = 5$ steps on a square lattice with ±90° bond angles. The tail and head of each walk are denoted by a circle and arrow, respectively.

1. Choose a chain at random and remove the tail link.
2. Attempt to add a link to the head of the chain. There is a maximum of two directions in which the new head link can be added.
3. If the attempt violates the self-intersection constraint, return to the original chain and interchange the head and tail. Include the chain in the statistical sample.

The above steps are repeated many times to obtain an estimate of R^2.

As an example of the reptation method, consider chain a of Figure 7.8. A new link can be added in two directions (see Figure 7.9a), so that on the average we find $a \to \frac{1}{2}c + \frac{1}{2}d$. In contrast, a link can be added to chain b in only one direction, and we obtain $b \to \frac{1}{2}e + \frac{1}{2}b$, where the tail and head of chain b have been interchanged (see Figure 7.9b). Confirm that $c \to \frac{1}{2}e + \frac{1}{2}a$, $d \to \frac{1}{2}c + \frac{1}{2}d$, and $e \to \frac{1}{2}a + \frac{1}{2}b$, and that all five chains are equally probable. That is, the transformations in the reptation method preserve the proper statistical weights of the chains without attrition. There is just one problem: unless we begin with a double-ended "cul-de-sac" configuration, such as shown in Figure 7.10, we will never obtain such a configuration using the above transformation. Hence, the reptation method introduces a small statistical bias, and the calculated mean end-to-end distance will be slightly larger than if all configurations were considered. However, the probability of such trapped configurations is very small, and the bias can be neglected for most purposes.

***Problem 7.30 The reptation method**

(a) Adopt the ±90° bond angle restriction and calculate by hand the exact value of $\langle R^2 \rangle$ for $N = 5$. Then write a Monte Carlo program that implements the reptation method. Generate one walk of $N = 5$ and use the reptation method to generate a statistical sample of chains. As a check on your program, compute $\langle R^2 \rangle$ for $N = 5$ and compare your result with the exact result. Then extend your Monte Carlo computations of $\langle R^2 \rangle$ to larger N.