(a) Write a program to implement this algorithm in one dimension. One way to implement step (iii) is given in the following code, where `totalMass` is the sum of the lengths of all the objects.

```
int i = 0;   // label of object
// length of first object, all lengths are integers
int sum = length[0];
// choose object to fragment so that choice is proportional to
// length
int x = (int)(Math.random()*totalMass);
while(sum < x) {
   i++;
   sum += length[i];
}
// if object big enough to fragment, choose random fraction for
// each part
if(length[i] > 1) {
   int partA = 1 + (int)(Math.random()*(length[i]-1));
   int partB = length[i] - partA;
   length[i] = partA;
   length[numberOfObjects] = partB;  // new object
   numberOfObjects++;
}
```

The main quantity of interest is the distribution of lengths $P(L)$. Explore a variety of initial length distributions with a total mass of 5000 for which the distribution is peaked at about 20 mass units. Is the long time behavior of $P(L)$ similar in shape for any initial distribution? Compute the total mass (sum of the lengths) and output this value periodically. Although the total mass will fluctuate, it should remain approximately constant. Why?

(b) Collect data for three different initial distributions with the same number of objects N, and scale $P(L)$ and L so that the three distributions roughly fall on the same curve. For example, you can scale $P(L)$ so that the maximum of the three distributions has the same value. Then multiply each value of L by a factor so that the distributions overlap.

(c) The analytical results suggest that the universal behavior can be obtained by scaling L by the total mass raised to the $1/3$ power. Is this prediction consistent with your results? Test this hypothesis by adjusting the initial distributions so that they all have the same total mass. Your results for the long time behavior of $P(L)$ should fall on a universal curve. Why is this universality interesting? How can this result be used to analyze different systems? Would you need to do a new simulation for each value of L?

(d) What happens if step (iii) is done more or less often than each random change of length. Does the scaling change? ■

Project 7.41 Application of the pivot algorithm to self-avoiding walks

The algorithms that we have discussed for generating self-avoiding random walks are all based on making *local* deformations of the walk (polymer chain) for a given value of N, the number of bonds. As discussed in Problem 7.31, the time τ between statistically independent configurations is nonzero. The problem is that τ increases with N as some power, for example, $\tau \sim N^3$. This power law dependence of τ on N is called *critical*

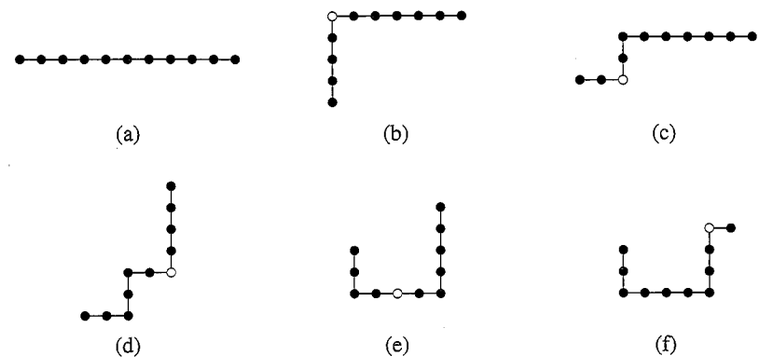

Figure 7.14 Examples of the first several changes generated by the pivot algorithm for a self-avoiding walk of $N = 10$ bonds (11 sites). The open circle denotes the pivot point. This figure is adopted from the article by MacDonald et al.

slowing down and implies that it becomes increasingly more time consuming to generate long walks. We now discuss an example of a *global* algorithm that reduces the dependence of τ on N. Another example of a global algorithm that reduces critical slowing down is discussed in Project 15.32.

(a) Consider the walk shown in Figure 7.14a. Select a site at random and one of the four possible directions. The shorter portion of the walk is rotated (pivoted) to this new direction by treating the walk as a rigid structure. The new walk is accepted only if the new walk is self-avoiding; otherwise, the old walk is retained. (The shorter portion of the walk is chosen to save computer time.) Some typical moves are shown in Figure 7.14. Note that if an end point is chosen, the previous walk is retained. Write a program to implement this algorithm and compute the dependence of the mean square end-to-end distance R^2 on N. Consider values of N in the range $10 \leq N \leq 80$. A discussion of the results and the implementation of the algorithm can be found in MacDonald et al. and Madras and Sokal, respectively.

(b) Compute the correlation time τ for different values of N using the approach discussed in Problem 7.31b. ■

Project 7.42 Pattern formation

In Problem 7.34 we saw that simple patterns can develop as a result of random behavior. The phenomenon of pattern formation is of much interest in a variety of contexts ranging from the large scale structure of the universe to the roll patterns seen in convection (for example, smoke rings). In the following, we explore the patterns that can develop in a simple reaction diffusion model based on the reactions, $A + 2B \rightarrow 3B$ and $B \rightarrow C$, where C is inert. Such a reaction is called *autocatalytic*.

In Problem 7.34 we considered chemical reactions in a closed system where the reactions can proceed to equilibrium. In contrast, open systems allow a continuous supply of fresh reactants and a removal of products. These two processes allow steady states to be realized and oscillatory conditions to be maintained indefinitely. In this problem we assume that A is added at a constant rate, and that both A and B are removed by the feed pro-