A SIMPLE MARKOV MODEL
FOR THE CONSTRUCTION OF BROWNIAN MOTION
ON SIERPIŃSKI'S HEXAGONAL GASKET

by

Richard F. Green

Technical Report 89-03

Department of Mathematics and Statistics
University of Minnesota
Duluth, Minnesota 55812

March 1989

A SIMPLE MARKOV MODEL FOR THE CONSTRUCTION OF BROWNIAN MOTION ON SIERPIŃSKI'S HEXAGONAL GASKET

Richard F. Green
Department of Mathematics and Statistics
University of Minnesota
Duluth, Minnesota 55812

INTRODUCTION

Sierpiński's hexagonal gasket is a fractal generated in an infinite series of steps by beginning with the area inside a regular hexagon, removing 2/9 of the area, leading to 7 smaller regular hexagons, each of whose area is 1/9 of that of the original, then, in the second step, removing 2/9 of the area of each hexagon produced in the first step, resulting in 49 still smaller regular hexagons, each of whose area is 1/81 of that of the original, and so on. An illustration of the first two steps is given in Figure 1.

If we write X_0 for the original hexagon, X_1 for the set of seven hexagons resulting from the first step, illustrated in Fig. 1b, X_2 for the 49 hexagons illustrated in Fig. 1c, and X_n for the set of 7^n hexagons, each of area 9^{-n}, that are left after n steps, then the area of X_n will be $(7/9)^n$. If we write

$$X = \bigcap_{i=1}^{\infty} X_i, \tag{1}$$

then the measure of X will be 0. That is,

$$\mu(X) = \lim_{n\to\infty} (7/9)^n = 0. \tag{2}$$

The set X is a fractal having Hausdorff dimension

$$d = \frac{\log 7}{\log 3} \doteq 1.77124375.$$

The problem of interest is to construct Brownian motion on X. The aspect that makes this problem interesting is that, in a sense, X contains mostly open space and a random walk on X would be highly restricted by the sparseness of X.

Unrestricted Brownian motion in the plane may be generated by a random walk on a two-dimensional grid of points corresponding to pairs of integers. At each instant the process takes a step of unit length at random in one of the four directions, North, East, South or West. Such a process, viewed from a great distance, with time slowed down appropriately, would yield two-dimensional Brownian motion with North-South and East-West displacements from the origin having independent, normal distributions. One could generate an equivalent Brownian motion by considering a process that takes steps of length two (Manhattan distance), with particular steps parallel to an axis having probability $1/12$ each, while displacements diagonal to the axes have probability $1/6$ each, and the steps occurring after time intervals of length $8/3$. Two-dimensional Brownian motion may be generated in many ways, but the important thing is that no matter how the process is generated, it will look the same from a great distance after a long time, and, except for scale, it will look the same if viewed after a long time, or after a longer time. This property of looking alike at different scales, which here is due to the stability of the normal distribution, has been referred to by Mandelbrot as "self-similarity."

We would like to generate Brownian motion on Sierpiński's gasket, and we observe that any continuous process that is at some time within a hexagon at the nth level (whose area is $(1/9)^n$ times the area of the original hexagon) and which leaves this hexagon, must do so through one of its vertices. It seems sensible to model the movement within the gasket by considering steps from one vertex of a hexagon at level n to another vertex of the same hexagon. It is not obvious how to do this, but if the process is self-similar (at some high level— for example, for hexagons of finite area) then it must be true that (i) the probabilities of similar transitions must be the same at different levels. That is, the probability that a particle which enters a hexagon at level n at a given vertex will reach a particular adjacent vertex of that hexagon before reaching one of the four other vertices (I ignore revisits to the point of entry) should be the same as the probability that the particle will hit the corresponding adjacent vertex in the hexagon at level $n-1$ before one of the four other vertices in this larger hexagon, and (ii) the ratio of the expected times to hit particular vertices in hexagons of one size will be the same as the ratios of times to hit corresponding vertices in hexagons of different sizes.

What I will do is calculate the probabilities of particular transitions from one vertex of a hexagon to another *under the assumption that these probabilities are the same at different levels* (that is, are invariant), and I will give evidence that these probabilities are, in fact, invariant by showing that other choices of transition probabilities at a low level lead to the invariant probabilties at a higher level. I will also find approximate values for the ratios of the expected times to make particular transitions within a hexagon.

METHODS

Finding the invariant transition probabilities

I think of a process consisting of the movement of a particle within a hexagon. I assume that the particle enters the hexagon at a vertex labelled (1) in Fig. 2b, and will remain within the hexagon at least until it reaches one of the other vertices, which may be labelled (2), (3), (4), (5) and (6) in counter clockwise order starting from point (1). The probability that the particle reaches vertex (2) first is $p(1)$ [the probaility of reaching vertex (6) first is also $p(1)$], the probability of reaching vertex (3) first is $p(2)$ [the probability of reaching vertex (5) first is also $p(2)$], and the probability of reaching vertex (4) first is $p(3) = 1 - 2p(1) - 2p(2)$. These transitions and the corresponding probabilities are illustrated in Fig. 2a.

What I want to do is find values of $p(1), p(2)$ and $p(3)$ such that the probabilities of reaching vertices (2), 3), (4), (5), and (6) first from vertex (1) will be $p(1), p(2), p(3), p(2)$ and $p(1)$, respectively, given that the probabilities of the corresponding transitions at the next lower level are the same. That is, in Fig. 2b, I set the probabilities of reaching vertices (7), (8), (9), (10) and (11), first from vertex (1) equal to $p(1), p(2), p(3), p(2)$, and $p(1)$, respectively. I begin by choosing candidate values for $p(1)$ and $p(2)$ [and, consequently, for $p(3)$], the transition probabilities from vertex to vertex within the smaller hexagons in Fig. 2b, and then calculate the resulting probabilities of reaching of the vertices (2), (3), (4), (5) and (6) before reaching any of the others. I do this by treating the process as an absorbing discrete Markov chain with 30 states, one corresponding to each vertex of one (or more) of the small hexagons. Time units correspond to transitions from one vertex to another. In any realistic model of diffusion these times would not be equal, but I may treat them as equal here because the only issue is which of the vertices (2), (3), (4), (5) or (6) is reached first.

In order to determine the transition probabilities for the 30-state Markov chain that I have in mind, it is useful to break the states into four groups:

(i) Vertex (1) is the "entry point." I assume that the process begins in state (1), corresponding to vertex (1) [I will identify states with vertices with the same numbers], and that the process must reach one of the other vertices [(2), (3), (4), (5) or (6)] of the large hexagon before leaving the large hexagon. From state (1) the process must go to one of the states (7), (8), (9), (10) or (11). The process may return to state (1), but whenever it does it must return to one of the states (7), (8), (9), (10) or (11).

(ii) The other vertices of the larger hexagon are treated as absorbing states (2), (3), (4), (5) and (6). The problem is to calculate the probability of being absorbed in each of these states.

(iii) Vertices (7), (12), (13), (16), (17), (20), (21), (24), (25), (29), (30) and (11) lie on the edges of the large hexagon and transitions from these vertices must be to other vertices within the same small hexagon.

(iv) The other vertices (14), (18), (19), (15), (8), (23), (9), (22), (27), (28), (10) and (26) lie inside the large hexagon and each of these points is a vertex of two small hexagons. Transitions from such vertices may be to any one of the other ten vertices in these two small hexagons. For example, the probability of a transition from vertex (8) to vertex (7) is $.5p(1)$.

If we write the probability of being in state (k) at time t as $P(k,t)$, then we can write a series of thirty equations, for $k = 1, 2, \ldots, 30$. For any time t the sum of these probabilities will be one. That is,

$$\sum_{k=1}^{30} P(k,t) = 1, \qquad \text{for} \quad t = 0, 1, 2, \ldots \tag{3}$$

We start with $P(1,0) = 1$. Eventually, we have

$$\lim_{t \to \infty} \sum_{k=2}^{6} P(k,t) = 1. \tag{4}$$

A BASIC program that does the calculations is given in Appendix 1. I will not write all thirty equations here, but I will give examples for three of the four types of vertices listed above. For vertex (1) we have

$$\begin{aligned} P(1,t) =& p(1)P(7,t-1) + .5p(2)P(8,t-1) + .5p(3)P(9,t-1) \\ & + .5p(2)P(10,t-1) + p(1)P(11,t-1). \end{aligned} \tag{5}$$

For vertices (2), (3), (4), (5) and (6) the equations are similar. For vertex (7), an edge point, we have

$$\begin{aligned} P(7,t) =& p(1)P(1,t-1) + .5p(1)P(8,t-1) + .5p(2)P(9,t-1) \\ & + .5p(3)P(10,t-1) + p(2)P(11,t-1). \end{aligned} \tag{6}$$

The equations are similar for other edge points, except that for other edge points [with the exception of vertex (11) which is in the same small hexagon as the starting point, vertex (1)], it is impossible to reach the point from the vertex of the larger hexagon which is also a vertex of the small hexagon containing the point of interest. Interior vertices can be reached in one step from eight, nine or ten other vertices, with the number depending on how many of the vertices of the two small hexagons that the vertex is in are stopping points of the process. Vertex (9) is the only one which can be reached from ten other vertices. We have

$$\begin{aligned} P(9,t) =& p(3)P(1,t-1) + p(2)P(7,t-1) + .5p(1)P(8,t-1) \\ & + .5p(1)P(10,t-1) + p(2)P(11,t-1) \\ & + .5p(1)P(15,t-1) + .5p(2)P(19,t-1) \\ & + .5p(3)P(23,t-1) + .5p(2)P(27,t-1) \\ & + .5p(1)P(28,t-1). \end{aligned} \tag{7}$$

In the program in the Appendix I do not use a separate argument for time, but let the number of iterations take care of that. It is necessary to distinguish the present values, $P(k,t)$ from the previous values, so I use $Q(k)$ for the present values, which are calculated from the previous values, denoted by $P(k)$. After all the new values, $Q(k)$, have been calculated, they are substituted for the values of $P(k)$ for the next iteration.

The goal is to find the probabilities of being absorbed in states (2), (3), (4), (5) and (6). If we denote these probabilities by $P(k,\infty)$ for state k, we will have $P(2,\infty) = P(6,\infty)$, and $P(3,\infty) = P(5,\infty)$ by symmetry. The sum of all five probabilities will be one. For a given choice of $p(1)$ and $p(2)$, we can find the probabilities of being absorbed in states (2) and (3). What I want is to find values, $p(1)$ and $p(2)$ which will equal the values of the absorbing probabilities, $P(2,\infty)$ and $P(3,\infty)$, respectively.

The desired values of $p(1)$ and $p(2)$ may be found by choosing one set of values, say $p(1) = p(2) = 0.2[= p(3)]$, finding the resulting values of $P(2,\infty)$ and $P(3,\infty)$, using these values for $p(1)$ and $p(2)$ for the next iteration, and repeating the process until we have $P(2,\infty) = p(1)$ and $P(3,\infty) = p(2)$. Results are given in Table 1.

It is not obvious that the process described above will produce a unique pair of values for $p(1)$ and $p(2)$, but numerical calculations suggest that this is so. Table 2 gives values of $P(2,\infty)$ and $P(3,\infty)$ that result from beginning at a variety of values for $p(1)$ and $p(2)$. I believe that the values that I find for invariant probabilities $p(1)$ and $p(2)$ are unique.

It is possible to calculate the expected number of steps to reach an specified absorbing state, k, by using the following formula:

$$\sum_{t=1}^{\infty} t[P(k,t) - P(k,t-1)]/P(k,\infty). \tag{8}$$

The expected numbers of steps to reach specified absorbing states calculated in this way provide a good first approximation for the relative expected times to reach the various absorbing states.

Finding the relative expected times to make various transitions

If Brownian motion on Sierpiński's gasket is a self-similar process then the ratio of expected times to go from one vertex of a hexagon to particular vertices in that hexagon should be the same for hexagons of different sizes. In particular, these ratios should be the same for hexagons of one size and the next size larger. For example, the ratio of expected times to go from vertex (1) to vertex (8) to the expected time to go from vertex (1) to vertex (7) should be the same as the ratio of the expected time to go from vertex (1) to vertex (3) to the expected time to go from vertex (1) to vertex (2).

If we choose expected times to make transitions between vertices of a small hexagon, and if the ratios of these expected times are rational, then, in principle, we can calculate the expected times to make transitions between vertices of the next larger size of hexagon. This calculation can be done by using a semi-Markov model for the process. A semi-Markov process differs from a Markov process in that for a Markov process the probability of a transition to a particular state depends only on the present state, while for a semi-Markov process these transtions also depend on the time that has been spent in the present state.

I will let $t(1), t(2)$ and $t(3)$ be the expected times to make transitions of the type from vertex (1) to vertices (7) [or (11)], (8) [or (10)], and (9), respectively, In Fig. 2b. In order to do the calculations I require that these times be integers. Preliminary calculations indicate that $t(1) < t(2) < t(3)$. I assume that the process, beginning in state (1), will move to vertex (7) or vertex (11), each with probability $p(1)$, after exactly time $t(1)$, or will move to vertex (8) or vertex (10), each with probability $p(2)$, after exactly time $t(2)$, or will move to vertex (9) with probability $p(3)$ after exactly time $t(3)$. If the process has been in a state for some time not equal to $t(1), t(2)$ or $t(3)$ it must remain in that state at least until such a time is reached. After a stay of length $t(1)$ in a state [other that one of the absorbing states (2), (3), (4), (5) or (6)] the process will leave with probability equal to $2p(1)$, and then will move to an adjacent vertex in the same small hexagon. After a stay of length $t(2)$ in a state, the process will leave with probability $2p(2)/[1 - 2p(1)]$. After a stay of length $t(3)$ in a state the process must leave the state and move to the opposite vertex in the same small hexagon.

What I want to do is find the expected times $E1, E2$, and $E3$ that a process, beginning in state (1) at time 0, takes to reach absorbing states (2), (3) or (4), respectively, given expected times $t(1), t(2)$ and $t(3)$, respectively, to make the corresponding transitions within the small hexagons. I want to find values of $t(1)$, $t(2)$ and $t(3)$ such that $E2 : E1 = t(2) : t(1)$ and $E3 : E1 = t(3) : t(1)$.

My object is to find the expected time at which a process, beginning in state (1) at time 0, is absorbed in state (k), where $k = 2, 3, 4, 5$ or 6, given that the process is absorbed in state (k), the transition probabilities $p(1)$ and p(2) are the ones, and the expected times of transitions within a small hexagon are specified values, $t(1), t(2)$ and $t(3)$. What I will calculate is an array of probabilities, $P(j, t, i)$, where (j) is a state [$j = 1$ or $j = 7, 8, \ldots, 30$], t is the length of time that the process has spent in that state [$t = 1, 2, \ldots, t(3)$], and i is the time that has elapsed since the beginning of the process [that is, the time at which state (1) is first entered]. I will suppress the argument i, which appears in the BASIC program given in Appendix 2 as an index for the number of iterations. States (2), (3),(4), (5) and (6) are absorbing, and the time that the process has spent in them is irrelevant. The probabilities of being in one of these states at time i may be denoted as $S(j, i)$, or, suppressing the dependence on time i, as $S(j)$.

In doing the calculations, the probabilities of entering a state are of greatest interest. For example, if we write $q(1) = p(1)$, $q(2) = p(2)/[1 - 2p(1)]$, $q(3) = 1$, $q(4) = .5q(1)$,

$q(5) = .5q(2)$, and $q(6) = .5p(3)$, we have

$$\begin{aligned} P(9,1,i) = \quad & q(3)P(1,t(3),i-1) + q(2)P(7,t(2),i-1) \\ & + q(4)P(8,t(1),i-1) + q(4)P(10,t(1),i-1) \\ & + q(2)P(11,t(2),i-1) + q(4)P(15,t(1),i-1) \\ & + q(5)P(19,t(2),i-1) + q(6)P(23,t(3),i-1) \\ & + q(5)P(27,t(2),i-1) + q(4)P(28,t(1),i-1). \end{aligned} \tag{9}$$

For an absorbing state, for example, state (2), we have

$$\begin{aligned} S(2,i) = \quad & q(1)P(13,t(1),i-1) + q(5)P(14,t(2),i-1) \\ & + q(6)P(15,t(3),i-1) + q(5)P(8,t(2),i-1) \\ & + q(1)P(12,t(1),i-1) + S(2,i-1). \end{aligned} \tag{10}$$

In the program in Appendix 2, instead of using a three-dimensional array, I use $Q(j,t)$ for $P(j,t,i)$, $P(j,t)$ for $P(j,t,i-1)$, $S(j)$ for $S(j,i)$, and $Z(j)$ for $S(j,i-1)$.

The expected time to reach absorbing state (2), given that it is reached, is given by

$$E1 = \sum_{i=1}^{\infty} i[S(2,i) - S(2,i-1)]/P(2,\infty). \tag{11}$$

The values of greatest interest are the ratios, $E2 : E1$ and $E3 : E1$.

Finding the probabilities of escaping from the gasket through particular vertices

One question that arises in treating Brownian motion on Sierpiński's gasket—perhaps the main question—is what happens to a process that starts at a particular point within a gasket. That is, if a particle starts at some vertex of a hexagon at level n in a gasket, eventually the particle must reach one of the vertices of the outermost hexagon (the hexagon of level 0). The question is, which of these vertices will be reached first? To be more precise, we ask, what is the probability that vertex i (for $i = 1, 2, \ldots, 6$) of the outermost hexagon will be reached first? We can denote this probability as $P(w(n), w(0))$, where $w(n)$ denotes the starting vertex of a hexagon at level n and $w(0)$ denotes the stopping vertex of the hexagon at level 0.

In order to find the values $P(w(n), w(0))$ we need a bit of notation, an idea, and a routine calculation. First, we need a way of writing $w(n)$ so that its location is explicit. We have that $w(n)$ is a vertex of a hexagon at level n, which is itself contained within some hexagon at level $n-1$, which is, in turn, contained within some hexagon at level $n-2$, and so on until we reach the hexagon at level 1. I will write $w(n)$ as

$$w(n) = .h(1)h(2)\ldots h(n)v(n) \tag{12}$$

where $h(i)$ denotes a hexagon of level i that contains the point, and $v(n)$ denotes the vertex in the hexagon $h(1)h(2)\dots h(n)$. The vertices are numbered in the same way as illustrated in Figure 2. That is, $v(n) = 1$ is the right-most vertex in the level-n hexagon, and vertices $2, 3, \dots, 6$ are those encountered by moving counter-clockwise around the hexagon. The hexagons at a given level are numbered in a similar way, such that at the first level we have $h(1) = i$ (for $i = 1, 2, \dots, 6$) for the hexagon containing vertex i of the hexagon at level 0. There are seven smaller hexagons within each larger hexagon, and I denote the inner small hexagon as number 0. The labels given by (12) are not unique.

The idea that we need is that if a particle is at a vertex of a hexagon at level n, eventually this particle must hit one of the vertices of the hexagon at level $n-1$ which contains the hexagon at level n (if $v(n) = h(n)$ then the vertex of the hexagon at level n is already a vertex of the hexagon at level $n-1$, and the probability that this will be the first such vertex reached is one). I will calculate the probability that each of the six vertices of the hexagon of level $n-1$ will be the first one reached. Once we have the probability of first reaching each of the vertices of the hexagon of level $n-1$ which contains our starting point we will use these probabilities to calculate the probabilities of first reaching each vertex of the hexagon of level $n-2$ containing our starting point. We continue in this way until we have the probabilities of first hitting each of the vertices of the hexagon at level 0, the outermost hexagon.

We want to find $P_0(i) = P(w(n), w(0) = i)$, the probability that vertex i of the outer hexagon is reached first, for $i = 1, 2, \dots, 6$. Suppressing the dependence of $w(n)$ in the notation we may write $P_t(i)$ for the probability that vertex i of the hexagon at level t which contains our starting point is hit first. We have

$$\mathbf{P}_t \mathbf{M}_t = \mathbf{P}_{t-1} \tag{13}$$

where $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$ are row vectors whose entries are the probabilities of first reaching the various vertices of hexagons at levels t and $t-1$, respectively, and $\mathbf{M}_t$ is a 6×6 matrix whose entries are $M(i, j|k)$, the probabilities that a particle at vertex i in a hexagon k at level t will reach vertex j of the hexagon at level $t-1$ before reaching any other vertex of that hexagon. If we denote the starting point by $w(n) = .h(1)h(2)\dots h(n)v(n)$, we have

$$\begin{aligned} P_n(i) &= 1 \quad \text{if} \quad v(n) = i, && (14) \\ &= 0 \quad \text{otherwise, and} && \\ \mathbf{M}_n &= M(i, j|h(n)), \qquad \text{and, in general,} && (15) \\ \mathbf{M}_t &= M(i, j|h(t)). && \end{aligned}$$

We find $\mathbf{P}_0$ by beginning with these values and iterating.

Since there are seven possible values of $h(t)$, there are seven 6×6 matrices $\mathbf{M}(k)$. We may think of $M(i, j|k)$ as a three-dimensional array. The values of M may be calculated by slightly modifying the program given in Appendix 1. Because there is a great deal of symmetry, it is only necessary to calculate a few entries. It suffices to consider only one (of seven possible) starting hexagons (I choose $h(1) = 1$), and within this hexagon it is not necessary to consider all six vertices. The program in Appendix 1 was modified to make state (1) absorbing and to start at one of the points labelled (7), (8) or (9) in Fig. 2. Using the notation $w(n) = .h(1)h(2)\ldots h(n)v(n)$, we have that

$$\begin{aligned} &\text{Point (7) is} \quad w(1) = .12, \\ &\text{Point (8) is} \quad w(1) = .13, \quad \text{and} \\ &\text{Point (9) is} \quad w(1) = .14. \end{aligned}$$

Table 4 gives probabilities $P(i,j) = M(i,j|1)$ which are required to fill out the array $M(i,j|k)$, except for the cases in which a vertex of a hexagon at one level is also a vertex of a hexagon at the next level as well. Of course, such a point will "reach" itself first and we will have $P(1,1) = 1$ and $P(1,i) = 0$ for $i \neq 1$.

RESULTS

Invariant transition probabilities

The invariant transition probabilities are

$$\begin{aligned} p(1) &= 0.297369802570, \\ p(2) &= 0.143897908641, \quad \text{and} \\ p(3) &= 0.117464577578. \end{aligned} \tag{16}$$

It suffices to find $p(1)$ and $p(2)$ since $p(3) = 1 - 2p(1) - 2p(2)$. If we start by guessing that $p(1) = p(2) = 0.2$, and iterate, we find, successively, the pairs of values for $p(2,\infty)$ and $p(3,\infty)$ given in Table 1. That is, we have convergence to invariant probabilities if we start with $p(1) = p(2) = 0.2$. Notice that the convergence is quite fast. What this sugggests is that if we are interested in modelling Brownian motion on Sierpiński's gasket it is not very important what transition probabilities are assumed at the smallest scale. However, these invariant probabilities will help describe what happens at a large scale, no matter what happens at a small scale (unless, perhaps, some transitions are impossible).

Table 2 gives values for $P(2,\infty)$ and $P(3,\infty)$ resulting from various transition probabilities $p(1)$ and $p(2)$. While these results are only suggestive, they do suggest that almost any choice of probabilities $p(1)$ and $p(2)$ at a small scale will lead to the invariant probabilities given in (16) at a large scale.

Expected time to make transitions

Calculations of the sort described by equation (11), beginning with the invariant probabilities given in (16), yield values for the expected number of steps to reach absorbing states (2), (3), and (4) respectively from state (1):

$$\begin{aligned} E1 &= 11.39570637, \\ E2 &= 14.77299158, \quad \text{and} \\ E3 &= 15.84468003. \end{aligned} \tag{17}$$

Most interestingly, we have

$$\begin{aligned} E2 : E1 &= 1.2964, \quad \text{and} \\ E3 : E1 &= 1.3904. \end{aligned} \tag{18}$$

Notice that these ratios are quite close to $13 : 10$ and $14 : 10$.

Using the semi-Markov model with the invariant transition probabilities and $t(1) = 10$, $t(2) = 13$ and $t(3) = 14$, we obtain

$$\begin{aligned} E1 &= 129.03876196, \\ E2 &= 167.55556859, \quad \text{and} \\ E3 &= 179.81900463, \end{aligned} \tag{19}$$

and

$$\begin{aligned} E2 : E1 &= 1.2984902059, \quad \text{and} \\ E3 : E1 &= 1.3935270449. \end{aligned} \tag{20}$$

Table 2 gives the pair of ratios $E2 : E1$ and $E3 : E1$ for part of a small grid of pairs of values for $t(2)$ and $t(3)$, with $t(1) = 20$ fixed. I use this grid first to test the accuracy of linear interpolation within the grid and then to interpolate to find ratios $E2 : E1$ and $E3 : E1$ such that $E2 : E1 = t(2) : t(1)$ and $E3 : E1 = t(3) : t(1)$.

i) Accuracy of linear interpolation.

(1) If we think of the ratios $E2 : E1$ and $E3 : E1$ as functions of the ratios $t(2) : t(1)$ and $t(3) : t(1)$, then the values, given in (20), namely, $E2 : E1 = 1.2984902059$ and $E3 : E1 = 1.3935270449$, for $t(2) : t(1) = 26 : 20$ and $t(3) : t(1) = 28 : 20$, may be compared with the values

$$\begin{aligned} E2 : E1 &= 1.2984890630, \quad \text{and} \\ E3 : E1 &= 1.3935250524, \end{aligned} \tag{21}$$

obtained by interpolating between the values determined by $t(2) : t(1) = 26 : 20$ and $t(3) : t(1) = 29 : 20$, and by $t(2) : t(1) = 26 : 20$ and $t(3) : t(1) = 27 : 20$.

Or, we obtain

$$\begin{aligned} E2 : E1 = &\ 1.2984900390, \quad \text{and} \\ E3 : E1 = &\ 1.3935278501 \end{aligned} \tag{22}$$

when we interpolate between the values determined by $t(2) : t(1) = 25 : 20$ and $t(3) : t(1) = 28 : 20$, and by $t(2) : t(1) = 27 : 20$ and $t(3) : t(1) = 28 : 20$. Notice that the interpolated values differ from the exact values given in (20) only in the 6th decimal place for (21) and in the 7th decimal place for (22).

(2) Now interpolate inside the square whose vertices are:

$$\begin{aligned} (t(2) : t(1) = &\ 25 : 20, \quad t(3) : t(1) = 28 : 20), \\ (t(2) : t(1) = &\ 26 : 20, \quad t(3) : t(1) = 28 : 20), \\ (t(2) : t(1) = &\ 25 : 20, \quad t(3) : t(1) = 27 : 20), \quad \text{and} \\ (t(2) : t(1) = &\ 26 : 20, \quad t(3) : t(1) = 27 : 20) \end{aligned} \tag{23}$$

to approximate the ratios $E2 : E1$ and $E3 : E1$ when the expected transition times are $t(1) = 21, t(2) = 27$ and $t(3) = 29$. Direct calculation for these expected transition times shows:

$$\begin{aligned} E2 : E1 = &\ 1.2984003212, \text{and} \\ E3 : E1 = &\ 1.3933951394. \end{aligned} \tag{24}$$

Interpolation, assuming that the ratios $E2 : E1$ and $E3 : E1$ are linear along cross-sections perpendicular to the $t(2) : t(1)$ and $t(3) : t(1)$ ratio axes, yields approximations:

$$\begin{aligned} E2 : E1 = &\ 1.2984000166, \quad \text{and} \\ E3 : E1 = &\ 1.3933948254. \end{aligned} \tag{25}$$

Notice that the interpolated values given in (25) differ from the exact values given in (24) only in the 7th decimal place.

ii) Final interpolation to obtain $E2 : E1$ and $E3 : E1$

Finally, I want to interpolate within the square whose vertices are given in (23), again assuming that the ratios $E2 : E1$ and $E3 : E1$ are linear along cross-sections perpendicular to the $t(2) : t(1)$ and $t(3) : t(1)$ ratio axes, in order to obtain values for the $E2 : E1$ and $E3 : E1$ ratios such that $E2 : E1 = t(2) : t(1)$ and $E3 : E1 = t(3) : t(1)$.

This interpolation, which is done by iteration in the brief BASIC program given in Appendix 3, yields the values:

$$\begin{aligned} E2 : E1 = &\quad 1.2984603125, \quad \text{and} \\ E3 : E1 = &\quad 1.3934776234. \end{aligned} \tag{26}$$

Since the ratios given in (26) are closer to one of the vertices of the square specified by (23), namely ($t(2) : t(1) = 26 : 20, t(3) : t(1) = 28 : 20$), than are the ratios $t(2) : t(1) = 27 : 21 = 1.2857143$ and $t(3) : t(1) = 29 : 21 = 1.3809524$, I expect the approximate values given in (26) to be closer to the unknown desired values than the interpolated values given in (25) are to the known values given in (24). Thus, I am quite certain that the values given in (26) are accurate to six decimal places, I think that they probably are accurate to seven decimal places, but I doubt that they are accurate to eight decimal places.

The time scaling factor

If we want to set equal to one the average time to go from one vertex at a certain level to another at that level we can use (26) for the ratios $t(2) : t(1)$ and $t(3) : t(1)$ and solve the equation

$$2p(1)t(1) + 2p(2)t(2) + p(3)t(3) = 1 \tag{27}$$

and get

$$\begin{aligned} t(1) &= 0.88330224760 \\ t(2) &= 1.14693291250 \\ t(3) &= 1.2308619168. \end{aligned} \tag{28}$$

Then we can find the "time scaling factor" (Lindstrøm, "Brownian motion on nested fractals," ms.) λ, which will be the average time to go from a vertex at one level to another vertex at the next level. The factor λ is found by interpolating a function of the form

$$f(E2 : E1, E3 : E1) = 2p(1)E1 + 2p(2)E2 + p(3)E3 \tag{29}$$

at the point specified by (26) within the square whose vertices are specified by (23). The values of $f(.,.)$ are, respectively,

$$\begin{aligned} f(1.25, 1.40) &= 12.88950188032586, \\ f(1.30, 1.40) &= 12.89027491648719, \\ f(1.25, 1.35) &= 12.88965645688207, \\ f(1.30, 1.35) &= 12.89043156562745. \end{aligned} \tag{30}$$

The resulting value is

$$\lambda = 12.89027151380. \tag{31}$$

Comparison of an interpolated value with one for which the exact value is known (to the limits of accuracy of the calculation) suggests that this value of λ is accurate to five decimal places, and probably six, and that a good estimate of the value of the time scaling factor is: $\lambda = 12.890272$.

What this number means is that for Sierpiński's hexagonal gasket it takes about 13 times as long to go three times as far. (For Brownian motion it takes $9 = 3^2$ times as long to go three times as far.) Put another way, for Sierpiński's hexagonal gasket, time scales as the $\log(\lambda)/\log 3 = 2.327002$ power of distance.

The probability of leaving the gasket through particular points, calculated as a function of location within the gasket

Table 4 gives the probabilities that a particle at vertex i in hexagon 1 at level n will visit vertex j at level $n-1$ before visiting any other vertex at level $n-1$. These probabilities, which I write as $P(i,j)$, really depend on the fact that I am starting in hexagon 1, and I should write $P(i,j) = M(i,j|1)$. While 18 probabilities are given in Table 4, only 13 of these are different because, by symmetry:

$$\begin{aligned} P(i=3, j=1) = {} & P(3,1) = P(3,2), \\ P(3,3) = {} & P(3,6), \\ P(3,4) = {} & P(3,5), \\ P(4,2) = {} & P(4,6), \quad \text{and} \\ P(4,3) = {} & P(4,5). \end{aligned}$$

Since our particle is starting in hexagon 1 we have

$$\begin{aligned} P(1,i) = {} & 1 \quad \text{if } j = 1, \text{ and} \\ = {} & 0 \quad \text{if } j \neq 1. \end{aligned} \tag{32}$$

By symmetry we have, for particles starting at vertices $i = 5$ or $i = 6$:

$$\begin{array}{ll} P(5,1) = P(3,1) & P(6,1) = P(2,1) \\ P(5,2) = P(3,6) & P(6,2) = P(2,6) \\ P(5,3) = P(3,5) & P(6,3) = P(2,5) \\ P(5,4) = P(3,4) & P(6,4) = P(2,4) \\ P(5,5) = P(3,3) & P(6,5) = P(2,3) \\ P(5,6) = P(3,2) & P(6,6) = P(2,2). \end{array} \tag{33}$$

For particles starting in hexagons $k = 2,3,4,5$ or 6 we can find the transition probabilities $M(i,j|k)$ from the probabities given in Table 4, those given by expression (32), and those found by using equations (33) by symmetry. Using a sort of modulus function:

$$\begin{aligned} s(i) = &\quad i \quad \text{for } i = 1,2,3,4,5 \text{ or } 6, \text{ and} \\ s(0) = &\quad 6, \\ s(-1) = &\quad 5, \\ s(-2) = &\quad 4, \\ s(-3) = &\quad 3, \\ s(-4) = &\quad 2, \\ s(-5) = &\quad 1, \end{aligned} \tag{34}$$

we have

$$\begin{aligned} M(i,j|2) =& M(s(i-1),s(j-1)|1) = P(s(i-1),s(j-1)), \\ M(i,j|3) =& M(s(i-2),s(j-2)|1) = P(s(i-2),s(j-2)), \\ M(i,j|4) =& M(s(i-3),s(j-3)|1) = P(s(j-3),s(j-3)), \\ M(i,j|5) =& M(s(i-4),s(j-4)|1) = P(s(i-4),s(j-4)), \\ M(i,j|6) =& M(s(i-5),s(j-5)|1) = P(s(i-5),s(j-5)). \end{aligned} \tag{35}$$

Finally, since a vertex of the inner hexagon, $w(1) = .0i$, is also a vertex of an outer hexagon, $w(1) = .is(i-3)$, we have

$$M(i,j|0) = M(s(i-3),j|i). \tag{36}$$

The values given in Table 4, those given by expression (32), and those found by using equations (33), (35), and (36), may be used in equation (13) to find the desired escape probabilities, $P(w(n), w(0))$.

ACKNOWLEDGMENTS AND DISCUSSION

The calculations that I do in this technical report were stimulated by the discussion of Tom Lindstrøm's monograph, "Brownian motion on nested fractals," in a graduate seminar course given by David Ross. Differences in notation and in language between this report and Lindstrøm's monograph are due to my having undertaken the calculations without having seen the monograph.

Originally I set out to find the invariant transition probabilities $p(1), p(2)$ and $p(3)$. Then I found the expected transition times, or rather, their ratios. Then I realized that if I used a convenient notation for points in the gasket I could find an algorithm that would permit a numerical solution of the Dirichlet problem. Finally, after looking at Lindstrøm's monograph, I realized that I could calculate the value of the "time scaling constant," and I did so. This constant, which says that the time to travel a certain distance tends to increase at a higher rate than the square of the distance, is perhaps the fact of greatest practical interest about Brownian motion on fractals. In order to give a physical interpretation to this constant it might be convenient to construct Sierpiński's hexagonal gasket from the inside out, rather than the outside in. With an inside-out construction we would begin with a regular grid of hexagons, connected at their vertices [that is, a homeotoxal tiling of type $< 3.6; 4^2 >$, see Grünbaum and Shephard (1987, pp.179-180)], and then holes of ever-increasing size would be created in the grid by eliminating hexagons and hexagon-shaped sets of hexagons in a regular way. One difficulty in interpreting the time scaling constant is that for Brownian motion on a fractal (as opposed to a homogeneous grid) the distribution of distance moved in a given time depends on the starting point.

LITERATURE CITED

Grünbaum, B., and G. L. Shephard. 1987. *Tilings and Patterns.* W. H. Freeman, New York.

Lindstrøm, T. ms. *Brownian motion on nested fractals.*

Mandelbrot, B. 1983. *The Fractal Geometry of Nature.* W. H. Freeman, New York.

TABLE 1. Calculated values of transition probabilities, $p(2,\infty)$ and $p(3,\infty)$, beginning with $p(1) = 0.2$, $p(2) = 0.2$, and $p(3) = 0.2$, and iterating. At each iteration the previous values of $p(2,\infty)$ and $p(3,\infty)$ are used as new values for $p(1)$ and $p(2)$, respectively, and the values for $p(2,\infty)$ and $p(3,\infty)$ are compared with the limiting values: $p(1) = 0.2973698025704289$ and $p(2) = 0.1438979086412864$, respectively. The values of $p(2,\infty)$ and $p(3,\infty)$ for the *i*th iteration are compared with the limiting values, and the differences are given to three significant figures. Values for $p(2,\infty)$ and $p(3,\infty)$ for successive iterations are rounded to twelve decimal places. Calculations are done in double precision.

	Values		Differences		
i	$p(2,\infty)$	$p(3,\infty)$	$p(2,\infty)$	$p(3,\infty)$	
0	0.2	0.2	-9.74	5.61	$(\times 10^{-2})$
1	0.2890625	0.1484375	-8.31	4.54	$(\times 10^{-3})$
2	0.296612956552	0.144314111578	-7.57	4.16	$(\times 10^{-4})$
3	0.297301080011	0.143935883475	-6.87	3.80	$(\times 10^{-5})$
4	0.297363585190	0.143901348081	-6.22	3.44	$(\times 10^{-6})$
5	0.297369240567	0.143898219605	-5.62	3.11	$(\times 10^{-7})$
6	0.297369751778	0.143897936746	-5.08	2.81	$(\times 10^{-8})$
7	0.297369797980	0.143897911181	-4.59	2.54	$(\times 10^{-9})$
8	0.297369802156	0.143897908871	-4.15	2.30	$(\times 10^{-10})$
9	0.297369802533	0.143897908662	-3.75	2.07	$(\times 10^{-11})$
10	0.297369802567	0.143897908643	-3.39	1.87	$(\times 10^{-12})$
11	0.297369802570	0.143897908641	-3.06	1.69	$(\times 10^{-13})$
12	0.297369802570	0.143897908641	-2.74	1.53	$(\times 10^{-14})$

TABLE 2. Values for $P(2,\infty)$ and $P(3,\infty)$ for a variety of values for $p(1)$ and $p(2)$. Most of these values seem to be fairly close to the stationary transition probabilities: $p(1) = 0.297$ and $p(2) = 0.144$.

		$p(1)$										
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
	0.00	.000 .000	.129 .083	.188 .118	.224 .135	.247 .143	.264 .146	.277 .147	.288 .146	.297 .143	.306 .139	.316 .132
	0.05	.116 .094	.218 .125	.218 .140	.243 .146	.261 .149	.274 .149	.285 .147	.294 .144	.303 .140	.312 .135	
	0.10	.173 .133	.213 .145	.239 .150	.257 .152	.271 .151	.283 .149	.292 .146	.300 .142	.309 .137		
$p(2)$	0.15	.208 .151	.235 .155	.254 .155	.269 .153	.281 .151	.290 .147	.299 .143	.306 .139			
	0.20	.233 .159	.253 .158	.268 .155	.279 .152	.289 .148	.297 .144	.305 .140				
	0.25	.252 .161	.267 .158	.279 .153	.289 .149	.297 .145	.304 .141					
	0.30	.270 .159	.281 .154	.291 .149	.299 .145	.306 .141						
	0.35	.286 .153	.295 .148	.303 .144	.309 .139							
	0.40	.305 .144	.311 .139	.316 .136								
	0.45	.329 .130	.332 .127									
	0.50	.368 .105										

TABLE 3. $E2 : E1$ and $E3 : E1$ ratios for part of a grid of $t(2) : t(1)$ and $t(3) : t(1)$ ratios. In all cases $t(1) = 20$. Values were not calculated for $t(2) = 27$ and $t(3) = 27$ because of the way I wrote the program (see Appendix 2).

	$t(2)$		
$t(3)$	25	26	27
29	1.2987034655 1.3939857984	1.2987138464 1.3939169309	1.2987239683 1.3938497820
28	1.2984768947 1.3935912518	1.2984902059 1.3935270449	1.2985031833 1.3934644484
27	1.2982479782 1.3931926203	1.2982642796 1.3931331739	

TABLE 4. P(i,j), the probability that a particle at vertex i in hexagon 1 at level n will visit vertex j at level $n-1$ before visiting any other vertex at that level.

	i = 2	i = 3	i = 4
j = 1	.5534134658	.318385988	.3080737534
j = 2	.1495690579	.318385988	.1736930029
j = 3	.0671851101	.111205627	.1188406744
j = 4	.0520785899	.070408385	.1068588920
j = 5	.0607687708	.070408385	.1188406744
j = 6	.1169850055	.111205627	.1736930029

Fig. 1. The initiator (a), the generator (b), and the next stage (c) in the generation of Sierpiński's hexagonal gasket. Sierpiński's hexagonal gasket X is generated by beginning with the hexagon, X_0 , illustrated in (a), removing twelve triangular areas, yielding the seven hexagonal areas contained in X_1, illustrated in (b). Then twelve small triangular areas are removed from each of these hexagons, resulting in the 49 small hexagons contained in X_2, illustrated in (c). This process of removing more and more, ever smaller, triangular areas continues indefinitely, and the limiting structure is the fractal X, Sierpiński's hexagonal gasket. As more and more triangles are removed, the gasket will contain more and more holes, each of which will assume the shape of a "Koch island" (see Mandelbrot 1983, pp. 42-3).

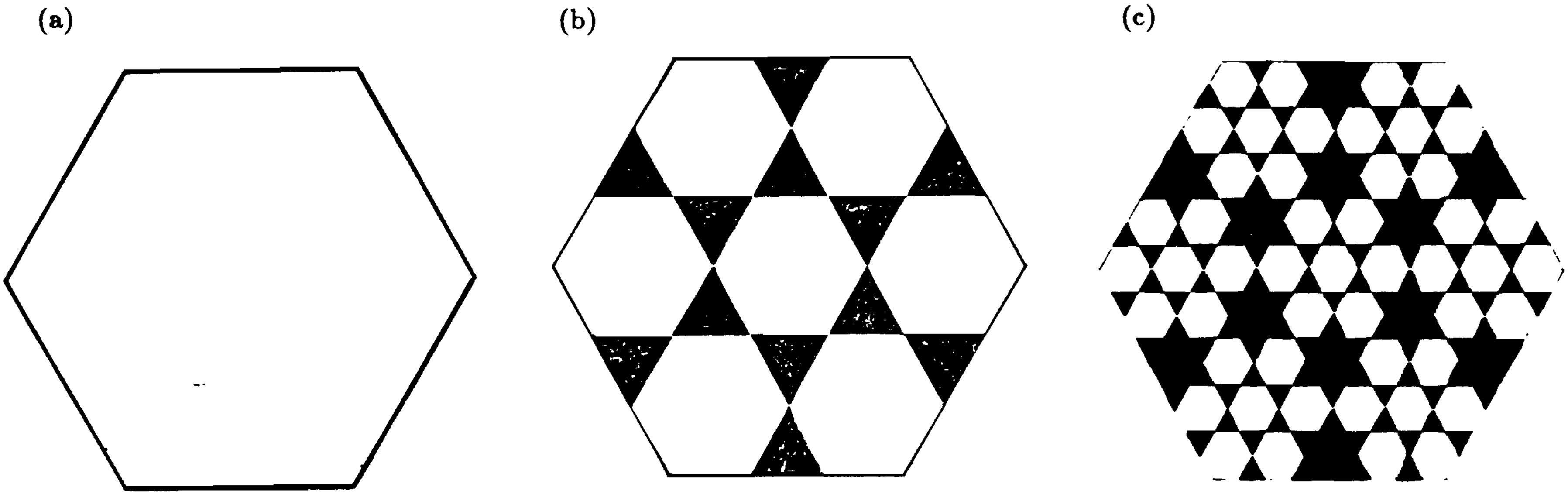
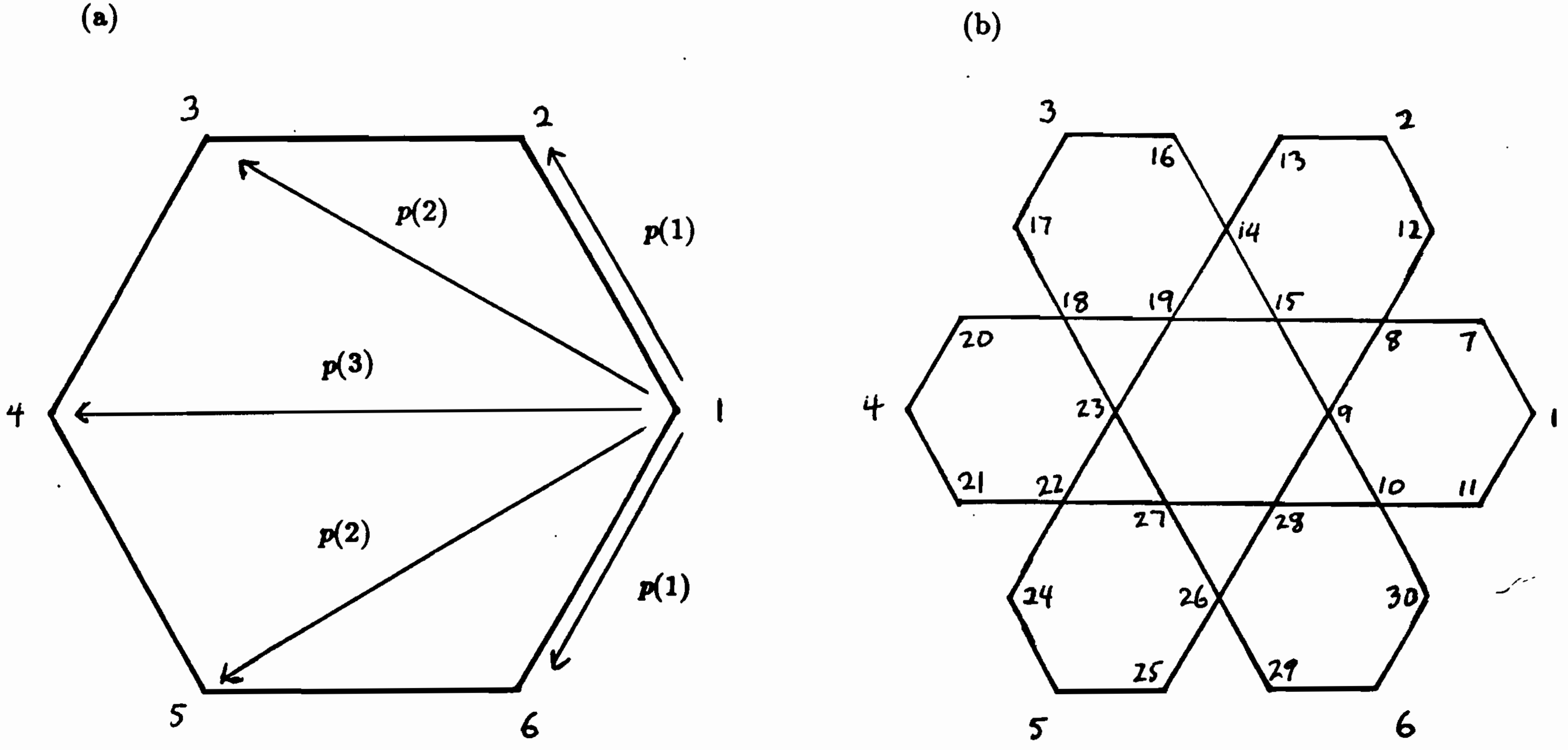


Fig. 2. Graphs on which a random walk takes place. The probabilities indicated in (a) are the probabilities of a transition of a given type, given that a transition is made from vertex (1) to one of the other vertices of X_0, namely (2), (3), (4), (5) or (6). The thirty vertices of the seven hexagons of X_1 are labelled in (b). This labelling is used in the formulas given in the text and in the programs used to calculate invariant transition probabilities and the relative expected times to make the transitions.



APPENDIX 1

This BASIC program calculates the probabilities that a random walk on the 30 vertices of Sierpiński's gasket at the first level will be absorbed at vertices (2), (3), (4), (5) or (6), given that it begins at vertex (1) [indicated by setting the probability of originally being in state (1), $P(1) = 1$, in line 40], and that the probabilities of transitions of types 1 and 2, as illustrated in Fig. 2a, namely $p(1)$ and $p(2)$, are given in lines 100 and 110, respectively. These transition probabilities, and $p(3)$, which is given in line 120, are printed in line 130.

The process is treated as an absorbing Markov chain which is followed through N steps, where N is given in line 200. Time is given as the index I in line 205. The probabilities of being in states (k) at time $t = I$, given as $P(k,t)$ in the text, are given as $P(k)$ in lines 210 to 500, and these are found in terms of the probabilities $P(k, t-1)$ in the text, or $Q(k)$ in the program, of being in state (k) at the previous time. The probabilities of first reaching the vertices (2), (3), (4), (5) or (6), respectively, which are given as the probabilities, $P(2,\infty), \ldots, P(6,\infty)$, of eventually being absorbed in the corresponding states, are approximately equal to the probabilities of being in those states after N steps. These probabilities are printed in line 710.

The expected number of steps to reach an absorbing state, given in (8) in the text, are printed in lines 800 to 820, the sums given in the numerator of (8) having been calculated in lines 510 to 530. The ratios of the expected number of steps to reach different absorbing states are printed in lines 900 and 910.

```
5 defdbl e, p, q
10 LPRINT "Sierpinski's gasket"
20 DIM P(30)
30 DIM Q(30)
40 P(1) = 1
100 P1 = .2973698025704289
110 P2 = .1438979086412864
120 P3 = 1 - 2*P1 - 2*P2
130 LPRINT "p1 =";P1;"p2 =";P2;"p3 =";P3
140 P4 = .5*P1
150 P5 = .5*P2
160 P6 = .5*P3
170 E1 = 0
180 E2 = 0
190 E3 = 0
200 N = 500
201 LPRINT "Number of time intervals =";N
205 FOR I = 1 TO N
210 Q(1) = P1*P(7) + P5*P(8) + P6*P(9) + P5*P(10) + P1*P(11)
220 Q(2) = P1*P(13) + P5*P(14) + P6*P(15) +_
 P5*P(8) + P1*P(12) + P(2)
```

```
230 Q(3) = P1*P(17) + P5*P(18) + P6*P(19) + P5*P(14)_
 + P1*P(16) + P(3)
240 Q(4) = P1*P(21) + P5*P(22) + P6*P(23) + P5*P(18)_
 + P1*P(20) + P(4)
250 Q(5) = P1*P(25) + P5*P(26) + P6*P(27) + P5*P(22)_
 + P1*P(24) + P(5)
260 Q(6) = P1*P(30) + P5*P(10) + P6*P(28) + P5*P(26)_
 + P1*P(29) + P(6)
270 Q(7) = P1*P(1) + P4*P(8) + P5*P(9) + P6*P(10) + P2*P(11)
280 Q(8) = P1*P(12) + P3*P(13) + P5*P(14) + P4*P(15)_
 + P1*P(7) + P2*P(1)+P3*P(11) + P5*P(10) + P4*P(9)
290 Q(9) = P3*P(1) + P2*P(7) + P4*P(8) + P4*P(10)_
 +P2*P(11) + P4*P(15) + P5*P(19) + P6*P(23) _
 + P5*P(27) + P4*P(28)
300 Q(10) = P1*P(11) + P2*P(1) + P3*P(7) + P5*P(8)_
 + P4*P(9) + P4*P(28) + P5*P(26) + P3*P(29) + P1*P(30)
310 Q(11) = P1*P(1) + P2*P(7) + P6*P(8) + P5*P(9) + P4*P(10)
320 Q(12) = P2*P(13) + P6*P(14) + P5*P(15) + P4*P(8)
330 Q(13) = P4*P(14) + P5*P(15) + P6*P(8) + P2*P(12)
340 Q(14) = P1*P(13) + P4*P(15) + P5*P(8) + P3*P(12)_
 +P1*P(16) + P3*P(17) + P5*P(18) + P4*P(19)
350 Q(15) = P2*P(13) + P4*P(14) + P4*P(8) + P2*P(12)_
 + P4*P(19) +P5*P(23) + P6*P(27) + P5*P(28) + P4*P(9)
360 Q(16) = P2*P(17) + P6*P(18) + P5*P(19) + P4*P(14)
370 Q(17) = P4*P(18) + P5*P(19) + P6*P(14) + P2*P(16)
380 Q(18) = P4*P(19) + P5*P(14) + P3*P(16) + P1*P(17)_
 + P1*P(20) + P3*P(21) + P5*P(22) + P4*P(23)
390 Q(19) = P4*P(14) + P2*P(16) + P2*P(17) + P4*P(18)_
 + P4*P(23) + P5*P(27) + P6*P(28) + P5*P(9) + P4*P(15)
400 Q(20) = P2*P(21) + P6*P(22) + P5*P(23) + P4*P(18)
410 Q(21) = P4*P(22) + P5*P(23)+P6*P(18) + P2*P(20)
420 Q(22) = P4*P(23) + P5*P(18) + P3*P(20) + P1*P(21)_
 +P1*P(24) +P3*P(25) + P5*P(26) + P4*P(27)
430 Q(23) = P4*P(18) + P2*P(20) + P2*P(21) + P4*P(22)_
 + P4*P(27) + P5*P(28) + P6*P(9) + P5*P(15) + P4*P(19)
440 Q(24) = P2*P(25) + P6*P(26) + P5*P(27) + P4*P(22)
450 Q(25) = P4*P(26) + P5*P(27) + P6*P(22) + P2*P(24)
460 Q(26) = P4*P(27) + P5*P(22) + P3*P(24) + P1*P(25)_
 + P1*P(29) + P3*P(30) + P5*P(10) + P4*P(28)
470 Q(27) = P4*P(26) + P2*P(25) + P2*P(24) + P4*P(22)_
 + P4*P(28) + P5*P(9) + P6*P(15) + P5*P(19) + P4*P(23)
480 Q(28) = P4*P(26) + P2*P(29) + P2*P(30) + P4*P(10)_

 + P4*P(9) + P5*P(15) + P6*P(19) + P5*P(23) + P4*P(27)
490 Q(29) = P2*P(30) + P6*P(10) + P5*P(28) + P4*P(26)
500 Q(30) = P4*P(10) + P5*P(28) + P6*P(26) + P2*P(29)
510 E1 = E1 + I*(Q(2) - P(2))
520 E2 = E2 + I*(Q(3) - P(3))
530 E3 = E3 + I*(Q(4) - P(4))
600 FOR J = 1 TO 30
610 P(J) = Q(J)
620 NEXT J
630 NEXT I
700 FOR I = 1 TO 6
710 LPRINT "i =";I;"P(i) =";P(I)
720 NEXT I
800 LPRINT E1;E1/P(2)
810 LPRINT E2;E2/P(3)
820 LPRINT E3;E3/P(4)
900 LPRINT "Time ratio 2:1";(E2/P(3))/(E1/P(2))
910 LPRINT "Time ratio 3:1";(E3/P(4))/(E1/P(2))
920 lprint p(2) - .297369802570428#, p(3) - .1438979086412864#,_
 p(4) - .1174645775765694#
```

APPENDIX 2

This BASIC program calculates the expected time that it takes for a semi-Markov process, beginning in state (1) [vertex (1)], to reach one of the absorbing states, (2), (3), (4), (5), or (6), given that the probabilities of transitions of types 1 and 2, shown in Fig. 2a, are the stationary ones, given as $P1$ and $P2$ on lines 100 and 110, respectively, and the expected times to make transitions of types 1, 2 and 3 at level 1 are $t(1)$, $t(2)$ and $t(3)$, respectively, given in lines 12 to 16. Transition probabilities are found in lines 122 to 126 and 140 to 160.

Probabilities $P(j,t,i)$ of having been in state j for length of time t at time i ("epoch" i, the time since the process started) are given as $q(i,t)$ in the program, where i in the program is j in the text and k in the program [which is implicit in $q(i,t)$] is i in the text. There are two classes of times t that the process has been in the state: (i) $1 < t \leq t(3)$, for which the probabilities are calculated in lines 162 to 192, and (ii) $t = 1$, for which the probabilities are calculated in lines 210 to 500.

Equation (9) in the text is given in line 290 of the program, and equation (1) is given in line 220. The sums required for the numerator of expectations such as given in equation (11) are calculated in lines 550 to 570, and these expectations are printed in line 670.

```
6 defint i, j, k, n, t
8 defdbl e, p, q, s, z
10 LPRINT "Sierpinski's gasket times"
12 t1 = 10
14 t2 = 13
16 t3 = 14
17 lprint "Input times:";"t1 =";t1;"t2 =";t2;"t3 =";t3
18 lprint "Input time ratios:  t2:t1 =";t2/t1;"t3:t1 =";t3/t1
20 DIM P(30,t3)
30 DIM Q(30,t3)
32 dim z(6)
34 dim s(6)
40 P(1,1) = 1
50 n = 8000
60 lprint "Number of intervals =";n
100 P1 = .297369802570
110 P2 = .143897908641
120 P3 = 1 - 2*P1 - 2*P2
122 q1 = p1
124 q2 = p2/(1 - 2*p1)
126 q3 = 1.0
130 LPRINT "p1 =";P1;"p2 =";P2;"p3 =";P3
140 q4 = .5*q1
150 q5 = .5*q2
160 q6 = .5*q3
161 for k = 1 to n
162 for t = 1 to t1 - 1
163 q(1,t+1) = p(1,t)
164 for i = 7 to 30
166 q(i,t+1) = p(i,t)
168 next i
170 next t
```

```
171 q(1,t1+1) = (1 - 2*q1)*p(1,t1)
172 for i = 7 to 30
173 q(i,t1+1) = (1 - 2*q1)*p(i,t1)
174 next i
175 if t2 = t1 + 1 then goto 182
176 for t = t1 + 1 to t2 - 1
177 q(1,t+1) = p(1,t)
178 for i = 7 to 30
179 q(i,t+1) = p(i,t)
180 next i
181 next t
182 q(1,t2+1) = (1 - 2*q2)*p(1,t2)
183 for i = 7 to 30
184 q(i,t2+1) = (1 - 2*q2)*p(i,t2)
185 next i
186 if t3 = t2 + 1 then goto 210
187 for t = t2 + 1 to t3 - 1
188 q(1,t+1) = p(1,t)
189 for i = 7 to 30
190 q(i,t+1) = p(i,t)
191 next i
192 next t
210 q(1,1) = q1*P(7,t1) + q5*P(8,t2) + q6*P(9,t3)_
 + q5*P(10,t2) + q1*P(11,t1)
220 s(2) = q1*P(13,t1) + q5*P(14,t2)_
 + q6*P(15,t3) + q5*P(8,t2) + q1*P(12,t1) + z(2)
230 s(3) = q1*P(17,t1) +_
 q5*P(18,t2) + q6*P(19,t3) + q5*P(14,t2) + q1*P(16,t1) + z(3)
240 s(4) = q1*P(21,t1) + q5*P(22,t2)_
 + q6*P(23,t3) + q5*P(18,t2) + q1*P(20,t1) + z(4)
250 s(5) = q1*P(25,t1) + q5*P(26,t2) + q6*P(27,t3)_
 + q5*P(22,t2) + q1*P(24,t1) + z(5)
260 s(6) = q1*P(30,t1) + q5*P(10,t2) + q6*P(28,t3)_
 + q5*P(26,t2) + q1*P(29,t1) + z(6)
270 Q(7,1) = q1*P(1,t1) + q4*P(8,t1) + q5*P(9,t2) + q6*P(10,t3)_
 + q2*p(11,t2)
280 Q(8,1) = q1*P(12,t1) + q3*P(13,t3) + q5*P(14,t2)_
 + q4*P(15,t1) + q1*P(7,t1) + q2*P(1,t2)_
 + q3*P(11,t3) + q5*P(10,t2) + q4*P(9,t1)
290 Q(9,1) = q3*P(1,t3) + q2*P(7,t2) + q4*P(8,t1) + q4*P(10,t1)_
 + q2*P(11,t2) + q4*P(15,t1) + q5*P(19,t2)_
 + q6*P(23,t3) + q5*P(27,t2) + q4*P(28,t1)
300 Q(10,1) = q1*P(11,t1) + q2*P(1,t2) + q3*P(7,t3) + q5*P(8,t2)_
 + q4*P(9,t1) + q4*P(28,t1) + q5*P(26,t2)_
 + q3*P(29,t3) + q1*P(30,t1)
310 Q(11,1) = q1*P(1,t1) + q2*P(7,t2) + q6*P(8,t3) + q5*P(9,t2)_
 + q4*P(10,t1)
320 Q(12,1) = q2*P(13,t2) + q6*P(14,t3) + q5*P(15,t2)_
 + q4*P(8,t1)
330 Q(13,1) = q4*P(14,t1) + q5*P(15,t2) + q6*P(8,t3)_
 + q2*P(12,t2)
340 Q(14,1) = q1*P(13,t1) + q4*P(15,t1)_
 + q5*P(8,t2) + q3*P(12,t3)_
 +q1*P(16,t1) + q3*P(17,t3) + q5*P(18,t2) + q4*P(19,t1)
350 Q(15,1) = q2*P(13,t2) + q4*P(14,t1)_
 + q4*P(8,t1) + q2*P(12,t2)_
 + q4*P(19,t1) +q5*P(23,t2) + q6*P(27,t3)_
 + q5*P(28,t2) + q4*P(9,t1)
360 Q(16,1) = q2*P(17,t2) + q6*P(18,t3) + q5*P(19,t2)_
 + q4*P(14,t1)
370 Q(17,1) = q4*P(18,t1) + q5*P(19,t2) + q6*P(14,t3)_
 + q2*P(16,t2)
```

```
380 Q(18,1) = q4*P(19,t1) + q5*P(14,t2) + q3*P(16,t3)_
 + q1*P(17,t1) + q1*P(20,t1)_
 + q3*P(21,t3) + q5*P(22,t2) + q4*P(23,t1)
390 Q(19,1) = q4*P(14,t1) + q2*P(16,t2) + q2*P(17,t2)_
 + q4*P(18,t1) + q4*P(23,t1) + q5*P(27,t2)_
 + q6*P(28,t3) + q5*P(9,t2) + q4*P(15,t1)
400 Q(20,1) = q2*P(21,t2) + q6*P(22,t3)_
 + q5*P(23,t2) + q4*P(18,t1)
410 Q(21,1) = q4*P(22,t1) + q5*P(23,t2)_
 + q6*P(18,t3) + q2*P(20,t2)
420 Q(22,1) = q4*P(23,t1) + q5*P(18,t2) + q3*P(20,t3)_
 + q1*P(21,t1) + q1*P(24,t1) + q3*P(25,t3)_
 + q5*P(26,t2) + q4*P(27,t1)
430 Q(23,1) = q4*P(18,t1) + q2*P(20,t2) + q2*P(21,t2)_
 + q4*P(22,t1) + q4*P(27,t1) + q5*P(28,t2)_
 + q6*P(9,t3) + q5*P(15,t2) + q4*P(19,t1)
440 Q(24,1) = q2*P(25,t2) + q6*P(26,t3) + q5*P(27,t2)_
 + q4*P(22,t1)
450 Q(25,1) = q4*P(26,t1) + q5*P(27,t2) + q6*P(22,t3)_
 + q2*P(24,t2)
460 Q(26,1) = q4*P(27,t1) + q5*P(22,t2) + q3*P(24,t3)_
 + q1*P(25,t1) + q1*P(29,t1) + q3*P(30,t3) + q5*P(10,t2)_
 + q4*P(28,t1)
470 Q(27,1) = q4*P(26,t1)_
 + q2*P(25,t2) + q2*P(24,t2)_
 + q4*P(22,t1) + q4*P(28,t1) + q5*P(9,t2)_
 + q6*P(15,t3) + q5*P(19,t2) + q4*P(23,t1)
480 Q(28,1) = q4*P(26,t1) + q2*P(29,t2) + q2*P(30,t2)_
 + q4*P(10,t1) + q4*P(9,t1) + q5*P(15,t2) + q6*P(19,t3)_
 + q5*P(23,t2) + q4*P(27,t1)
490 Q(29,1) = q2*P(30,t2) + q6*P(10,t3) + q5*P(28,t2)_
 + q4*P(26,t1)
500 Q(30,1) = q4*P(10,t1) + q5*P(28,t2) + q6*P(26,t3)_
 + q2*P(29,t2)
550 e2 = e2 + k*(s(2) - z(2))
555 e3 = e3 + k*(s(3) - z(3))
560 e4 = e4 + k*(s(4) - z(4))
565 e5 = e5 + k*(s(5) - z(5))
570 e6 = e6 + k*(s(6) - z(6))
575 z(2) = s(2)
580 z(3) = s(3)
585 z(4) = s(4)
590 z(5) = s(5)
595 z(6) = s(6)
600 for t = 1 to t3
605 p(1,t) = q(1,t)
610 for j = 7 to 30
615 p(j,t) = q(j,t)
620 next j
625 next t
650 next k
660 lprint e2;e3;e4;e5;e6
670 lprint e2/s(2);e3/s(3);e4/s(4);e5/s(5);e6/s(6)
680 lprint "Output time ratios:  e2:e1 =";_
 (e3/s(3))/(e2/s(2));_
 "e3:e1 =";(e4/s(4))/(e2/s(2))
690 w = 0
700 FOR I = 2 TO 6
705 w = w + s(i)
710 LPRINT "i =";I;"S(i) =";s(I)
720 NEXT I
725 lprint "Sum of probabilities =";w
730 end
```

APPENDIX 3

This BASIC program interpolates to find values of the ratio $E2 : E1$ equal to $t(2) : t(1)$ and of the ratio $E3 : E1$ equal to $t(3) : t(1)$ under the assumption that the ratios $E2 : E1$ and $E3 : E1$ are linear in the ratio $t(2) : t(1)$ for a given ratio $t(3) : t(1)$ and linear in the ratio $t(3) : t(1)$ for a given ratio $t(2) : t(1)$. We begin with values A, B, C and D for the ratio $E2 : E1$ and values E, F, G and H for the ratio $E2 : E1$

when $t(2) : t(1) = R1 = 1.25$, and $t(3) : t(1) = R3 = 1.40$,

when $t(2) : t(1) = R2 = 1.30$, and $t(3) : t(1) = R3 = 1.40$,

when $t(2) : t(1) = R1 = 1.25$, and $t(3) : t(1) = R4 = 1.35$, and

when $t(2) : t(1) = R2 = 1.30$, and $t(3) : t(1) = R4 = 1.35$, respectively.

The interpolation is done by iteration, with the iterate number, I, and the successive iterates of $E2 : E1$ and $E3 : E1$, which are denoted by $X3$ and $Y3$, respectively, being printed in line 300.

```
10 DEFDBL X, Y, R, A, B, C, D, E, F, G, H
20 R1 = 1.25#
30 R2 = 1.3#
40 R3 = 1.4#
50 R4 = 1.35#
60 X = 1.3#
70 Y = 1.4#
75 LPRINT X,Y
80 A = 1.2984768947#
90 B = 1.2984902059#
100 C = 1.2982479782#
110 D = 1.2982642796#
120 E = 1.3935912518#
130 F = 1.3935270449#
140 G = 1.3931926203#
150 H = 1.3931331739#
200 FOR I = 1 TO 10
210 X1 = A + (X - R1)*(B - A)/(R2 - R1)
220 Y1 = E + (X - R1)*(F - E)/(R2 - R1)
230 X2 = C + (X - R1)*(D - C)/(R2 - R1)
240 Y2 = G + (X - R1)*(H - G)/(R2 - R1)
250 X3 = X1 + (Y - R3)*(X2 - X1)/(R4 - R3)
260 Y3 = Y1 + (Y - R3)*(Y2 - Y1)/(R4 - R3)
300 LPRINT I,X3,Y3
310 X = X3
320 Y = Y3
330 NEXT I
```