Methodol Comput Appl Probab (2013) 15:485–509
DOI 10.1007/s11009-011-9254-7

Exact Computation and Approximation of Stochastic and Analytic Parameters of Generalized Sierpinski Gaskets

Uta Freiberg · Christoph Thäle

Received: 26 July 2010 / Revised: 4 September 2011 /
Accepted: 12 September 2011 / Published online: 4 October 2011

Abstract The interplay of fractal geometry, analysis and stochastics on the one-parameter sequence of self-similar generalized Sierpinski gaskets is studied. An improved algorithm for the exact computation of mean crossing times through the generating graphs $SG(m)$ of generalized Sierpinski gaskets $sg(m)$ for m up to 37 is presented and numerical approximations up to $m = 100$ are shown. Moreover, an alternative method for the approximation of the mean crossing times, the walk and the spectral dimensions of these fractal sets based on quasi-random so-called rotor walks is developed, confidence bounds are calculated and numerical results are shown and compared with exact values (if available) and with known asymptotic formulas.

Keywords Crossing time · Einstein relation · Fractal geometry · Hausdorff dimension · Rotor walks · Sierpinski gasket · Spectral dimension · Walk dimension

AMS 2000 Subject Classifications 28A80 · 60J10 · 65C50 · 05C81

The second author was supported by the Swiss National Science Foundation Grant SNF PP002-114715/1.

U. Freiberg
University of Siegen, Emmy-Noether-Campus, Walter-Flex-Str. 3,
57068 Siegen, Germany
e-mail: freiberg@mathematik.uni-siegen.de

C. Thäle
Department of Mathematics, University of Fribourg,
Chemin du musée 23, 1700 Fribourg, Switzerland

Present Address:
C. Thäle (✉)
Institute of Mathematics, University of Osnabrück, Albrechtstr. 28a,
49076 Osnabrück, Germany
e-mail: christoph.thaele@uni-osnabrueck.de

1 Introduction

1.1 General Introduction

In this paper we consider a one-parameter family $sg(m)$, $m = 2, 3, 4, \ldots$ of self-similar fractals, which we call generalized Sierpinski gaskets, because the classical Sierpinski gasket is included as the special case $m = 2$ (see Falconer (2003) for the classical theory of self-similar fractals). For an illustration of the sets $sg(2)$, $sg(3)$, $sg(4)$ and $sg(5)$ see Fig. 1, the precise definition will follow in Section 2.1. These fractal sets belong to the class of so-called nested self-similar fractals, which opens the door to the machinery of analysis on fractals developed in Barlow (1998) or Kigami (2001) (for an introduction to analysis on fractals we also cite the survey article Freiberg (2005)). The results of this beautiful theory show that it is possible to construct a natural Laplacian on the family of sets $sg(m)$. To analyze further properties of this operator, it is necessary to calculate explicitly the so-called energy scaling factors $\rho(m)$. These energy scaling factors are strongly connected with the spectral dimensions of the fractal sets $sg(m)$. The spectral dimension $d_S(m)$ is thereby twice the exponent of the leading term in the eigenvalue counting function of the natural Laplacian on $sg(m)$, see Kigami (2001, Chap. 4). Denoting by $N(m)$ the number of similitudes of the iterated function system (IFS) generating $sg(m)$, we have the following formula

$$d_S(m) = \frac{2 \ln N(m)}{\ln(N(m)\rho(m))},$$

see Kigami and Lapidus (1993) or Kigami (2001). On the other hand, it is well known from classical probability theory that there exists a strong relationship between the Laplace operator and the Brownian motion process. More precisely, 1/2 times the d-dimensional Laplacian is the infinitesimal generator of the standard Brownian motion on $\mathbb{R}^d$, see Kallenberg (2002) for example. This relationship can also be utilized in the fractal setting. Once we have constructed a Laplacian on a fractal set, we also have a Brownian motion on it and vice versa. Also well known in the classical setting is the fact that a standard Brownian motion can be obtained as the limit of a sequence of suitably rescaled random walks (Donsker's invariance principle). On finitely ramified self-similar fractals there is a natural sequence of graphs approximating the fractal and hence it is natural to construct a Brownian motion on the fractal as the limit of appropriately time-changed simple random walks on these pre-fractal graphs. In order to obtain the correct time-scaling factors it is—because of the Markov property of the random walks and the self-similarity of the underlying sets—sufficient to regard the time-space scaling on the first-order approximations of the fractals $sg(m)$. We call these first-order approximating graphs the generalized Sierpinski graphs in this paper and denote them by $SG(m)$ (the exact mathematical definition will follow in Section 2.1).

A fundamental parameter of a fractal as $sg(m)$ that is related to the Brownian motion on it is the so-called walk dimension, which expresses the natural time-space

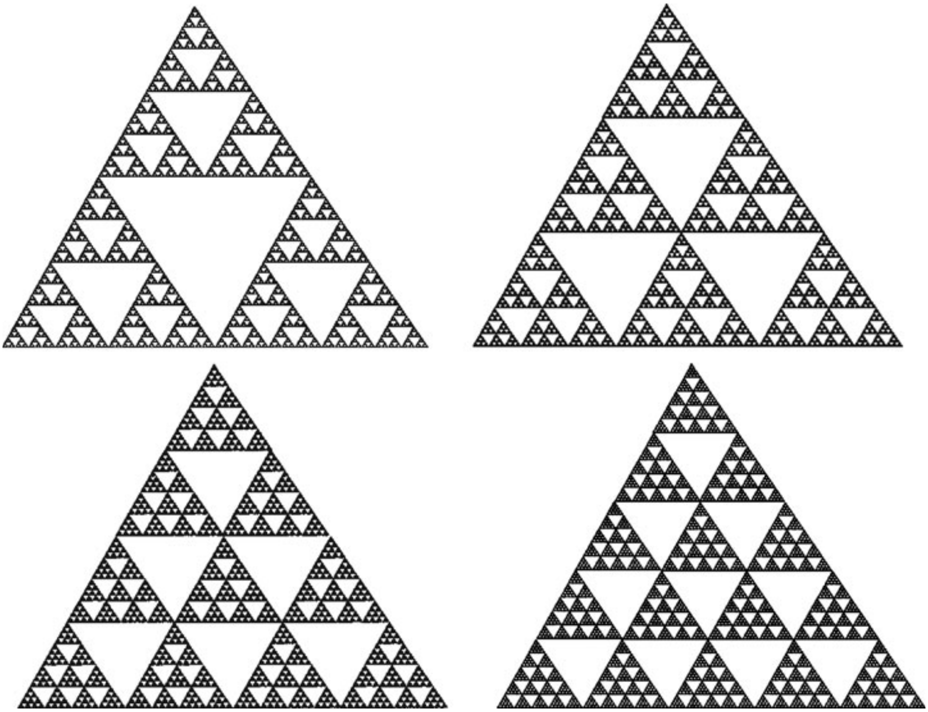

Fig. 1 The generalized Sierpinski gaskets *sg*(2)–*sg*(5)

scaling of the stochastic process. In order to define the walk dimension $d_W(m)$ of $sg(m)$ we denote by $\tau(B(x, R))$ the first exit time of the Brownian motion on $sg(m)$ from a ball centered at x with radius R. Then, the walk dimension is given by

$$d_W(m) := \frac{\ln \mathbb{E}_x \tau\ (B(x, R))}{\ln R},$$

where $\mathbb{E}_x$ stands for expectation conditioned on starting at $x \in sg(m)$. If the stochastic process is translation invariant, the dependence of x can be omitted. We will later on concentrate on the case where x is a vertex of one of the triangles in the fractal. One readily verifies that these points form a dense set. Note that in graph theory the limit as $R \to \infty$ is taken, but we will see that due to the self-similarity we will have a reasonable value for the walk dimension independently of the radius R.

Furthermore, the geometry of the fractals $sg(m)$ in connection with the arising natural strong reflection principle of the corresponding random walks on the pre-fractal graphs allows to reformulate the walk dimension in terms of the mean crossing time of the natural random walk on the generalized Sierpinski graph $SG(m)$. This mean crossing time will be denoted by $T(m)$ and is the mean value of the time (i.e.

the number of steps in this discrete setting), a simple random walk on $SG(m)$ starting at a vertex A needs to reach vertex B or C, when A, B and C are the vertices of the regular triangle generating $SG(m)$ and $sg(m)$, see Section 2.1. In this setting we can rewrite $d_W(m)$ as

$$d_W(m) = \frac{\ln \mathbb{E}T(m)}{\ln m},$$

where we have written $\mathbb{E}T(m)$ instead of $\mathbb{E}_A T(m)$. Moreover, the so-called Einstein relation holds, which connects the Hausdorff dimension d_H—reflecting the geometry—, the spectral dimension d_S—reflecting analytic properties—and the walk dimension d_W—reflecting the stochastic properties—of $sg(m)$, see Hilfer and Blumen (1988), Telcs (2006) and Zhou (1993). Einstein's relation in its dimensional form reads

$$2d_H(m) = d_S(m) \cdot d_W(m). \tag{1}$$

For the definition of the Hausdorff dimension of a self-similar fractal we refer to Falconer (2003). Its value is given by

$$d_H(m) = \frac{\ln N(m)}{\ln m},$$

where $N(m)$ is again the number of similitudes generating $sg(m)$ having joint contraction ratio $1/m$. Using the definition of the three different types of dimensions we can also reformulate the Einstein relation as

$$N(m)\rho(m) = \mathbb{E}T(m). \tag{2}$$

We would like to point out that the knowledge of the precise values of the parameters in Eq. 1 or Eq. 2 is of particular interest in physics. Transport phenomena in porous media are often modelled with the help of (partial) differential equations on fractals and properties of those materials can be derived from the different dimensions, see Freiberg (2005) and Hilfer and Blumen (1984) and the references cited therein.

1.2 Problem and Overview

In this subsection we describe a naive approach for the calculation of the mean crossing times $\mathbb{E}T(m)$ through the graphs $SG(m)$ and give a short survey of our results and the structure of the paper. We have already described in the previous section that due to the self-similarity of the fractal sets $sg(m)$ and the Markov property of the simple random walks on $SG(m)$ we have to calculate the mean crossing times $\mathbb{E}T(m)$ through $SG(m)$ to obtain the correct time-scaling, see Barlow (1998) or Kigami (2001). To illustrate the problem, we specialize for a moment to the simple example $m = 2$, see Fig. 2. It is easily verified by writing down a linear system

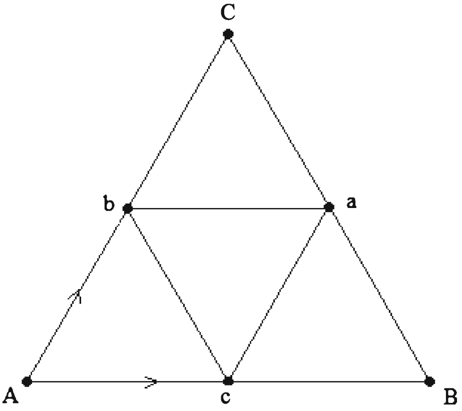

Fig. 2 The Sierpinski graph $SG(2)$

of equations that the mean crossing time through $SG(2)$ equals 5. Indeed, writing $\mathbb{E}_x\tau$ for the mean number of steps a simple random walk needs to reach vertex B or C when starting at x, we have

$$\begin{aligned}
\mathbb{E}_A\tau &= \frac{1}{2}\left(\mathbb{E}_c\tau + \mathbb{E}_b\tau\right) + 1,\\
\mathbb{E}_c\tau &= \frac{1}{4}\left(\mathbb{E}_A\tau + \mathbb{E}_B\tau + \mathbb{E}_a\tau + \mathbb{E}_b\tau\right) + 1,\\
\mathbb{E}_b\tau &= \frac{1}{4}\left(\mathbb{E}_A\tau + \mathbb{E}_c\tau + \mathbb{E}_a\tau + \mathbb{E}_C\tau\right) + 1,\\
\mathbb{E}_a\tau &= \frac{1}{4}\left(\mathbb{E}_B\tau + \mathbb{E}_C\tau + \mathbb{E}_b\tau + \mathbb{E}_c\tau\right) + 1,\\
\mathbb{E}_B\tau &= 0, \qquad \mathbb{E}_C\tau = 0,
\end{aligned}$$

which yields after a simple calculation the solution $\mathbb{E}_A\tau = 5$, $\mathbb{E}_b\tau = \mathbb{E}_c\tau = 4$ and $\mathbb{E}_a\tau = 3$. This means that the simple random walk on $SG(2)$ is 5-times slower than a random walk on the unilateral triangle ΔABC. Having calculated $\mathbb{E}T(2) = 5$, we are now able to proceed with analysing further properties of the Brownian motion and the Laplacian on $sg(2)$. For example, it is easily seen that

$$N(2) = 3 \quad \text{and} \quad \rho(2) = \frac{5}{3},$$

where for the last equality we have used Eq. (2). Furthermore, using the three dimension formulas from above, we obtain

$$d_H(2) = \frac{\ln 3}{\ln 2}, \qquad d_W(2) = \frac{\ln 5}{\ln 2}, \qquad d_S(2) = \frac{\ln 9}{\ln 5}.$$

However, only for a very few numbers of m, these characteristic properties of the fractal sets $sg(m)$ are explicitly known, which is yet of particular physical interested, see Hilfer and Blumen (1984). To be more precise, we recall that Given and Mandelbrot (1983) have calculated the mean crossing times of $SG(2)$–$SG(4)$ and in Freiberg and Thäle (2008) the authors confirmed these earlier results and extended the list until $m = 7$ by developing a first version of an algorithm based on Markov chains. It was a main disadvantage of this algorithm that in order to calculate the mean crossing time of $SG(m)$, a so-called connection matrix has to be entered by hand into a program. The aim of this paper is twofold. On the one hand side, we develop the mentioned algorithm from Freiberg and Thäle (2008) further and overcome the main difficulty. The only input of the new algorithm is simply the natural number m. This allows us to calculate explicitly the values of the mean crossing time for considerably larger values of m, namely until $m = 37$. However, the runtime of the algorithm is rather slow so that explicit expressions for the mean crossing times $\mathbb{E}T(m)$ for larger values of m are currently out of reach. For this reason, we will also present another approach, which allows approximations of $\mathbb{E}T(m)$, the walk and the spectral dimensions $d_W(m)$ and also of $d_S(m)$ by using Einstein's relation. Moreover, it will be shown, how (at least for small m, i.e. $m \leq 50$ for example) inequalities or confidence intervals for the true values can be obtained from these approximations. Our approach is based on a rather new technique, namely that of quasi-random rotor walks. These deterministic walks replace the random walks on $SG(m)$ and reduce the original problem to a simple counting procedure. The recent theory (Holroyd and Propp 2010) shows that the convergence behaviour of these rotor walks is much better than that of their random counterparts. Also our numerical results will confirm this observation: The rotor walks yield quite accurate approximations, whereas it was not possible for us to obtain results with only nearly the same precision with the help of random walks using the same number of steps of the algorithm (Monte-Carlo simulation). We are convinced that the new method presented here has applications far beyond the family of generalized Sierpinski gaskets and we hope that some new aspects may be explored in the near future. It should be possible to modify our algorithms in order to get the exact expected crossing times (or estimations for them) for general nested fractals (in particular for fractal n-gons) and even p.c.f. fractals. Even more appealing would be to calculate expected crossing times for certain classes of random fractals with our methods.

Both of our methods allow to obtain new insight into the asymptotic behaviour of the mean crossing times, the walk and the spectral dimensions of $sg(m)$ for large values of m. This was considered from a theoretical viewpoint in Hambly and Kumagai (2002), where it was proved that the mean crossing time of $SG(m)$ asymptotically behaves as m^2. In contrast to this we will see that the growth rate is rather different for small m, say m between 2 and 100. We will observe that the mean crossing times behaves for such m as $m^{2.3}$ and the decay down to m^2 is extremely slow.

2 An Exact Algorithm

The current Section presents an exact algorithm for the determination of the mean crossing times of generalized Sierpinski graphs based on Markov chain methods. It is an advancement and a considerable improvement of an earlier algorithm from Freiberg and Thäle (2008). At first, some basic notions and notation are fixed.

2.1 Basic Notation

In this subsection, we formally introduce the class of generalized Sierpinski gaskets $sg(m)$ and explain how they can be approximated by a sequence of nested networks. Hereby, some "building pattern"—the graph $SG(m)$ will play a crucial role.

Let us be given a unilateral triangle ΔABC with unit side length and apply the following construction: Divide each side of the triangle into m, $m \geq 2$, pieces of the same length $1/m$ and connect all pairs of dividing points such that the arising connection line is parallel with one of the original triangle sides. We end up with m^2 small equilateral triangles having side length $1/m$, $\frac{m(m+1)}{2}$ of them pointing up, and $\frac{m(m-1)}{2}$ of them pointing down. Now, we "waste" all the triangles pointing downwards, and then we iterate the procedure with the remaining (upward pointing) triangles, which means that we replace each of the triangles we kept by a figure of the same shape and suitable size. Proceeding that way, we obtain in the limit a self-similar set, which we will call generalized Sierpinski gasket and which we denote by $sg(m)$. The sets $sg(m)$ for $m = 2, 3, 4, 5$ are shown in Fig. 1.

In the language of Iterated Function Systems (IFS's), see for example Falconer (2003) or Kigami (2001), the set $sg(m)$ is the unique compact non-empty set which is invariant under the family of contractive similitudes

$$\Psi_m := \left\{\psi_i, i = 1, \ldots, \frac{m(m+1)}{2}\right\}$$

acting on the real plane $\mathbb{R}^2$ with

$$\psi_i(x) := \frac{1}{m}x + t_i, \quad x \in \mathbb{R}^2,$$

for some $t_i \in \mathbb{R}^2, i = 1, \ldots, \frac{m(m+1)}{2}$. In order to introduce some "numbering" on the smaller triangles, we agree upon the following notation: Denote by $\mathbf{u} := \overrightarrow{AB}$ and $\mathbf{v} := \overrightarrow{AC}$ the two-dimensional unit vectors pointing from the vertex A to the vertex B and the vertex C, respectively. Then,

$$\left\{t_i : i = 1, \ldots, \frac{(m+1)m}{2}\right\}$$

is the set of all vectors

$$\left\{\frac{k}{m}\mathbf{u} + \frac{l}{m}\mathbf{v} : k, l \geq 0, k + l < m\right\}$$

and we choose the indices i in such a manner that the triangles $\psi_i(\Delta_{ABC})$ are ordered from the bottom to the top, and in each row from the left to the right.

Due to a result of Hutchinson, for which we refer to Hutchinson (1981), the Hausdorff dimension of the generalized Sierpinski gasket $sg(m)$ is given by

$$d_H(m) = \frac{\ln(m+1) + \ln m - \ln 2}{\ln m}. \tag{3}$$

Now, fix some $m \geq 2$. The fractal $sg(m)$ can be approximated by an increasing sequence of nested graphs as follows: Define $\Gamma_0 = (V_0, E_0)$ to be the graph with vertex set $V_0 := \{A, B, C\}$ and edges $E_0 := \{AB, BC, AC\}$. We define an increasing sequence of vertex sets by

$$V_{n+1}^{(m)} := \bigcup_{i=1}^{\frac{(m+1)m}{2}} \psi_i(V_n^{(m)}), \qquad n \geq 0.$$

Note, that

$$\overline{\bigcup_{n\geq 0} V_n^{(m)}} = sg(m),$$

where the closure is taken with respect to the Euclidean metric in the plane.

We say that two points $p, q \in V_n^{(m)}$ are connected by an edge of $E_n^{(m)}$ if there is a n-tuple $(i_1, \ldots, i_n)$ of indices from the index set $\{1, \ldots, (m+1)m/2\}$ such that

$$p = \psi_{i_1} \circ \ldots \circ \psi_{i_n}(P)$$

and

$$q = \psi_{i_1} \circ \ldots \circ \psi_{i_n}(Q)$$

for some $P, Q \in V_0$ and some $\psi_{i_1}, \ldots, \psi_{i_n} \in \Psi_m$. By the self-similarity of $sg(m)$, the graph $\Gamma_n^{(m)} = (V_n^{(m)}, E_n^{(m)})$ consists of $[(m+1)m/2]^n$ just-touching copies of graph $\Gamma_1^{(m)} = (V_1^{(m)}, E_1^{(m)})$. We denote this 'building graph' by $SG(m)$.

As already discussed in the introduction, when investigating such fractals, one is in particular interested in calculating their spectral and walk dimension. Besides the Hausdorff dimension $d_H(m)$ of $sg(m)$, which reflects the geometric properties of the set, these numbers are related to the analytic and stochastic features of $sg(m)$. The spectral dimension is twice the leading exponents in the asymptotic expansion of the eigenvalue counting function of the canonical Laplacian on $sg(m)$ (see Kigami and Lapidus 1993; Kigami 2001). The walk dimension $d_W(m)$ of $sg(m)$ reflects the time-space-scaling of the canonical Brownian motion on $sg(m)$. This time continuous stochastic process can be constructed as the limit of a suitable normalized sequence of random walks on the approximating graphs $\Gamma_n^{(m)} = (V_n^{(m)}, E_n^{(m)})$, see Lindstrøm (1990). To be more precise, we regard a simple random walk on the graph

$\Gamma_n^{(m)} = (V_n^{(m)}, E_n^{(m)})$ and ask how long it takes the random walker to pass from A to B or C, i.e. we ask for the mean crossing time $\tau_{m,n}$ through a triangle which has (graph theoretical) side length m^n. The walk dimension of $sg(m)$ is given by (see Freiberg 2011; Telcs 2006; Zhou 1993)

$$d_W(m) := \lim_{n\to\infty} \frac{\ln \tau_{m,n}}{\ln(m^n)}.$$

We make the following crucial observation: From the self-similarity and finite ramification of the graphs $\Gamma_n^{(m)}$, it follows that $\tau_{m,n} = \tau_{m,1}^n$. The proof can be given by a simple induction. We look at the crossing through $\Gamma_2^{(m)}$ and regard this graph as the "union" of small $\Gamma_1^{(m)}$-graphs. Then we use a black-box-argument which yields $\tau_{m,2} = \tau_{m,1} \cdot \tau_{m,1}$.

Thus, we get

$$d_W(m) = \frac{\ln \tau_{m,1}}{\ln m} = \frac{\ln \mathbb{E}T(m)}{\ln m}.$$

Hence, calculation of the walk dimension of $sg(m)$ can be reduced to the determination of the mean crossing time $\mathbb{E}T(m)$ through the building graph $SG(m)$.

2.2 An Example

The aim of this subsection is to demonstrate the mathematical idea of our algorithm in the case of the generalized Sierpinski graph $SG(3)$. To this end, we start with the

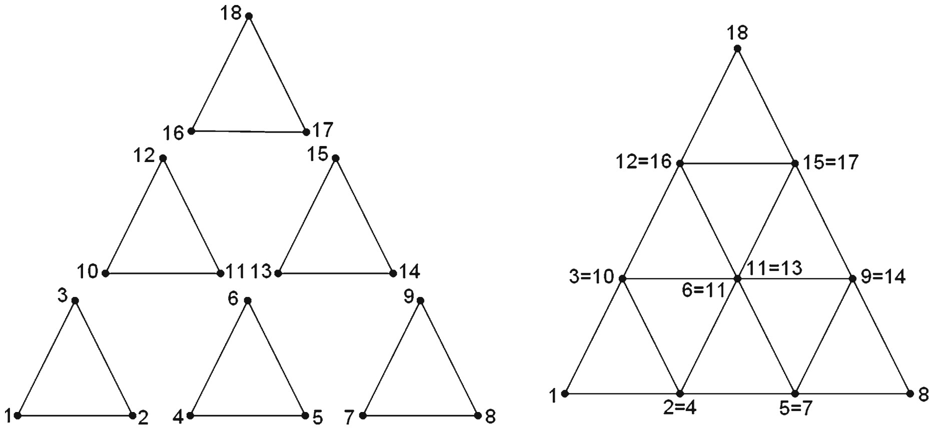

Fig. 3 The disconnected and the connected 'version' of the graph $SG(3)$

regular triangle ΔABC with vertices $A = (0, 0)$, $B = (1, 0)$ and $C = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$. As in the last subsection, let $\psi_1, \ldots, \psi_6$ be the six similarities mapping ΔABC onto the 6 disconnected triangles shown in Fig. 3. We furthermore introduce the 3×3 adjacency matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

and the 18×18 block diagonal matrix $\mathbf{M} := diag(\mathbf{A}, \mathbf{A}, \mathbf{A}, \mathbf{A}, \mathbf{A}, \mathbf{A})$. From the structure of $\mathbf{M}$ we see that $\mathbf{M}$ is the adjacency matrix of a graph consisting of 6 disconnected triangles shown in Fig. 3. In order to build from these six copies of ΔABC the generalized Sierpinski graph $SG(3)$, we have to connect several vertices of the currently unconnected graph. To this end, we introduce the connection matrix

$$\mathbf{C} = \begin{pmatrix} 2 & 3 & 5 & 6 & 9 & 11 & 12 & 15 \\ 4 & 10 & 7 & 11 & 14 & 13 & 16 & 17 \end{pmatrix},$$

in which all necessary identifications of vertices are coded up. Here, for example, the column $\begin{pmatrix} 2 \\ 4 \end{pmatrix}$ means that vertex with number 2 has to be connected with vertex having number 4. We now use the information coded in the connection matrix $\mathbf{C}$ in order to connect the six small copies of ΔABC. To do so, we read out the columns $\begin{pmatrix} c_{1,k} \\ c_{2,k} \end{pmatrix}$, $1 \leq k \leq 8$, and add row and column $c_{1,k}$ to row and column $c_{2,k}$ of the block diagonal matrix $\mathbf{M}$ and call the resulting matrix $\mathbf{M}'$. This matrix is the adjacency matrix of a graph consisting of 6 connected triangles and 18 nodes, but with several nodes identified. To remove these double points (nodes) we use again the connection matrix $\mathbf{C}$ and delete row and column $c_{1,k}$ for $1 \leq k \leq 8$. The result is the matrix

$$\mathbf{M}'' = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

We see that $\mathbf{M}''$ is the adjacency matrix of the generalized Sierpinski graph $SG(3)$.

It is the next goal to define a simple random walk on the approximating graph $SG(3)$. To this end we transform the matrix $\mathbf{M}''$ into a stochastic matrix (by normalizing each row such that it sums up to one), which yields the transition matrix of a simple random walk on the Sierpinski graph $SG(3)$. To calculate the mean crossing time $\mathbb{E}T(3)$ of $SG(3)$ we introduce the two cemetery (absorbing) states B and C by removing row and column associated with the two vertices B and C. The resulting matrix will be

$$\mathbf{M}''' = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{pmatrix}.$$

Note, that this matrix is no longer a stochastic matrix. In each row, which corresponds to a neighbouring point of an absorbing state, the row sum is strictly less than one.

From the theory of Markov chains (see for example Kemeny and Snell 1983) it is well known that the vector $\mathbf{T} = (\mathbb{E}_1 T, \mathbb{E}_{2=4} T, \ldots, \mathbb{E}_{18} T)$ of expected crossing times (i.e. $\mathbb{E}_k T$ is the mean number of steps the Markov chain stays in the non-absorbing component, provided we start in vertex k) is connected with the sub-stochastic matrix $\mathbf{M}'''$ by

$$\mathbf{T} = \mathbf{M}'''\mathbf{T} + \mathbf{1},$$

where $\mathbf{1} := (1, \ldots, 1)^T$. Thus, the mean crossing time $\mathbb{E}T(3) = \mathbb{E}_1 T$ is the first component of the vector

$$\left(\mathbf{E} - \mathbf{M}'''\right)^{-1} \mathbf{1},$$

where **E** is the identity matrix of suitable size (compare also with the linear system for $\mathbb{E}T(2)$ presented in Section 1.2). In our case we calculate

$$\left(\mathbf{E}-\mathbf{M}'''\right)^{-1}=\begin{pmatrix} \frac{15}{7} & \frac{16}{7} & \frac{8}{7} & \frac{16}{7} & \frac{15}{7} & \frac{6}{7} & \frac{8}{7} & \frac{6}{7} \\ \frac{8}{7} & \frac{853}{315} & \frac{79}{63} & \frac{587}{315} & \frac{15}{7} & \frac{277}{315} & \frac{65}{63} & \frac{263}{315} \\ \frac{4}{7} & \frac{79}{63} & \frac{113}{63} & \frac{65}{63} & \frac{11}{7} & \frac{55}{63} & \frac{43}{63} & \frac{41}{63} \\ \frac{8}{7} & \frac{587}{315} & \frac{65}{63} & \frac{853}{315} & \frac{15}{7} & \frac{263}{315} & \frac{79}{63} & \frac{277}{315} \\ \frac{5}{7} & \frac{10}{7} & \frac{22}{21} & \frac{10}{7} & \frac{19}{7} & \frac{20}{21} & \frac{22}{21} & \frac{20}{21} \\ \frac{3}{7} & \frac{277}{315} & \frac{55}{63} & \frac{263}{315} & \frac{10}{7} & \frac{523}{315} & \frac{41}{63} & \frac{257}{315} \\ \frac{4}{7} & \frac{65}{63} & \frac{43}{63} & \frac{79}{63} & \frac{11}{7} & \frac{41}{63} & \frac{113}{63} & \frac{55}{63} \\ \frac{3}{7} & \frac{263}{315} & \frac{41}{63} & \frac{277}{315} & \frac{10}{7} & \frac{257}{315} & \frac{55}{63} & \frac{523}{315} \end{pmatrix}$$

and hence

$$\mathbb{E}T(3)=e_1\left(\mathbf{E}-\mathbf{M}'''\right)^{-1}\mathbf{1}=e_1\left(\frac{90}{7},\frac{83}{7},\frac{59}{7},\frac{83}{7},\frac{72}{7},\frac{53}{7},\frac{59}{7},\frac{53}{7}\right)^T=\frac{90}{7},$$

where $e_1=(1,0,\ldots,0)$. In the next subsection we algorithmise the idea shown here and generalize to arbitrary integers $m\geq 2$.

2.3 The Algorithm

We will now present the main steps of our algorithm using a pseudo-code, which should be readable without knowledge of a specific programming language.

Step 1: **Input.** In contrast to the algorithm presented in Freiberg and Thäle (2008) the input is here simply the natural number $m\geq 2$:

```
input m;
```

Step 2: **Build isolated copies.** In this step the block diagonal matrix $\mathbf{M}=diag(\underbrace{\mathbf{A},\ldots,\mathbf{A}}_{m(m+1)/2})$ with

$$\mathbf{A}=\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}$$

is created:

```
M:=DiagonalMatrix(A,m(m+1)/2);
```

As explained in Section 2.2, this is the adjacency matrix of a graph consisting of $\frac{m(m+1)}{2}$ disconnected triangles, which are the scaled copies of a regular triangle ΔABC.

Step 3: Create the connection matrix. Now, the connection matrix **C** is computed. This is a matrix with 2 rows and $m^2 - 1$ columns

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,m^2-1} \\ c_{2,1} & c_{2,2} & \dots & c_{2,m^2-1} \end{pmatrix}$$

with the property that $c_{1,k} < c_{1,l}$ and $c_{1,k} < c_{2,k}$ for all $1 \leq k, l \leq m^2 - 1$ with $k < l$. The column $\begin{pmatrix} c_{1,k} \\ c_{2,k} \end{pmatrix}$ in the matrix **C** means that node $c_{1,k}$ is identified with node $c_{2,k}$. The algorithmic computation of **C** was one of the main problems to solve. Here, we will only use the 'black-box-command' `ConnectionMatrix(m)`:

```
C:=ConnectionMatrix(m);
```

and refer to the Appendix, where a MATLAB-solution is presented. The connection matrix of the generalized Sierpinski graph $SG(3)$ was already shown in Section 2.2. For $m = 4$ we have for example

$$\mathbf{C} = \begin{pmatrix} 2 & 3 & 5 & 6 & 8 & 9 & 12 & 14 & 15 & 17 & 18 & 21 & 23 & 24 & 27 \\ 4 & 13 & 7 & 14 & 10 & 17 & 20 & 16 & 22 & 19 & 23 & 26 & 25 & 28 & 29 \end{pmatrix},$$

when using the canonical enumeration of the nodes as shown in Fig. 4 (left). The graph with associated adjacency matrix **M** is shown in Fig. 4 (right) together with a visualization of the action of the connection matrix **C**. Observe that in the k-th row of triangles we have nodes with numbers

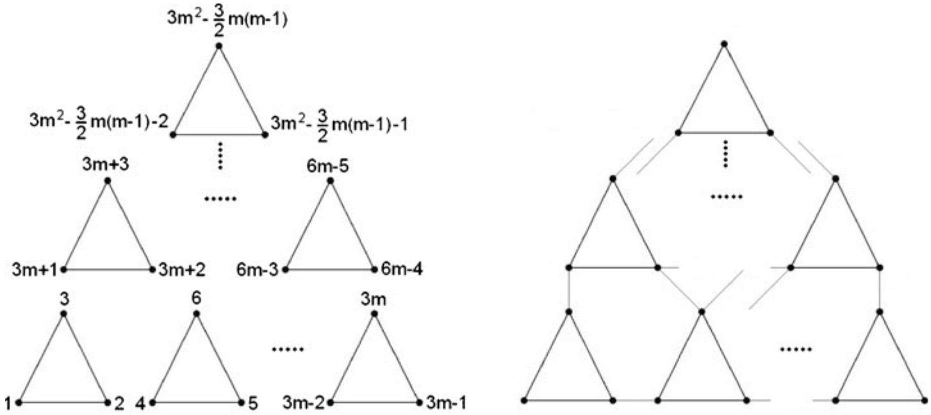

Fig. 4 Graph with adjacency matrix **M** and canonical enumeration (*left*); Visualization of the action of the connection matrix **C** (*right*)

$3(k-1)m - \frac{3}{2}(k-1)(k-2) + 1$ up to $3km - \frac{3}{2}k(k-1)$, which follows from simple combinatorial arguments.

Step 4: **Connect the isolated copies.** The information coded in the connection matrix $\mathbf{C}$ is now used to connect the isolated triangles of the graph with adjacency matrix $\mathbf{M}$. In order to glue together node k with node l, $k < l$, we add row and column k of the matrix $\mathbf{M}$ to row and column l:

```
for n from 1 to m^2-1 do AddRow(M,C[1,n],C[2,n]);
AddColumn(M,C[1,n],C[2,n]); end do;
```

The matrix $\mathbf{M}$ (this was the matrix $\mathbf{M}'$ in Section 2.2) is now the adjacency matrix of a graph consisting of isolated triangles but with several nodes identified. The double points are removed in

Step 5: **Remove double points.** Here, we only use the first row of the connection matrix $\mathbf{C}$, which consists of all nodes $c_{1,k}$ identified with nodes $c_{2,k}$, where $c_{1,k} < c_{2,k}$. In order to obtain a graph without double points we remove now all rows and columns of $\mathbf{M}$ with numbers $c_{1,k}$, $1 \leq k \leq m^2 - 1$:

```
for n from 1 to m^2-1 do RemoveRow(M,C[1,n]);
RemoveColumn(M,C[1,n]); end do;
```

The resulting matrix $\mathbf{M}$ is after Step 5 (this was the matrix $\mathbf{M}''$ in Section 2.2) the adjacency matrix of the generalized Sierpinski graph $SG(m)$.

Step 6: **Transform into a stochastic matrix.** We transform here the matrix $\mathbf{M}$ into the transition matrix of a Markov chain associated with the graph $SG(m)$:

```
for n from 1 to Dimension(M) do r:=RowSum(M,n);
MultiplyRow(M,1/r); end do;
```

We can think of the resulting Markov chain as a simple random walk on $SG(m)$.

Step 7: **Remove the absorbing states B and C.** In this step we remove the rows and columns associated with the the two corners B and C of the parent triangle ΔABC:

```
RemoveRow(M,r(B)); RemoveColumn(M,r(B));
RemoveRow(M,Dimension(M));
RemoveColumn(M,Dimension(M));
```

Here $r(B)$ is the row associated with the corner B and can be calculated using the first row of the connection matrix $\mathbf{C}$. Details can be found in the programming code in the Appendix. The matrix $\mathbf{M}$ after the final removal (this was in Section 2.2 the matrix $\mathbf{M}'''$) is the transition matrix of a Markov chain with two cemeteries B and C, i.e. an absorbing Markov chain in the sense of Kemeny and Snell (1983). Note that $\mathbf{M}$ has full rank, which will be important in the next step of our algorithm.

Step 8: **Solve the linear system.** In this last step we apply the theory of Markov chains as developed for example in Kemeny and Snell (1983) to calculate the mean time $\mathbb{E}T(m)$ (i.e. expected number of steps) the Markov chain with transition matrix $\mathbf{M}$ is in the non-absorbing component, i.e. on the nodes of the graph $SG(m)$ with the two corners B and C removed. This time can be computed with the formula

$$\mathbb{E}T(m) = e_1(\mathbf{E} - \mathbf{M})^{-1}\mathbf{1},$$

where again $\mathbf{E}$ is the identity matrix, $\mathbf{1}$ is the vector $\mathbf{1} = (1, 1, \dots, 1)^T$ and $e_1 = (1, 0, \dots, 0)$.

```
E:=IdentityMatrix; e:=Vector(1,Dimension(M));
T(m):=Multiply(Inverse(E-M),e)[1];
```

The output of our algorithm is an exact fraction for $\mathbb{E}T(m)$, which is the first component of the computed vector.

3 The Rotor-Router Approach

In this section we develop another approach for the calculation of the mean crossing times through generalized Sierpinski graphs, which does not yield exact values, but rather accurate approximations. Moreover, for small parameters m, reasonable confidence bounds or inequalities can be found.

3.1 Background Material

We describe here an alternative algorithmic approach for calculating or approximating the crossing times through the family of graphs $SG(m)$. The idea is to derandomize the problem and to study 'deterministic random walks'. The concept we will use is the so-called *rotor-router method*, which produces rotor walks on the graphs $SG(m)$ instead of random ones. By a *rotor walk* $(Y_n)_{n\in\mathbb{N}}$ on a graph $\Gamma = (V, E)$ with vertex set V and edge set $E \subseteq V \times V$ we mean a deterministic cellular automaton associated with the Markov chain introduced above, defined as follows: First, we are given an initial vertex $v_0 \in V$ and an initial rotor configuration $(r(v))_{v\in V}$, $r : V \to V$ is a function, which assigns to each vertex one of its neighbouring vertices. We can imagine $r(v)$ as an arrow pointing to one of the neighbours of v. Second, we are given a deterministic rule $s : V \to V$, which assigns to each neighbouring vertex v' of each vertex $v \in V$ a so-called successor $s(v')$, which is another neighbouring vertex of v. This means that with the help of the rule s we have introduced an ordering of all neighbouring vertices of a given vertex. The rotor walk develops according to the following transition rule: $Y_0 := v_0$ and for each $n \in \mathbb{N}$, $Y_{n+1} := r(Y_n)$, i.e. the walker walks along the current arrow assigned to the vertex Y_n. Moreover, the rotor $r(Y_n)$ is shifted after the transition to its successor, i.e. $r(Y_n) \mapsto s(r(Y_n))$. It is clear from the construction that once the rule s is fixed, the sequence $(Y_n)_{n\in\mathbb{N}}$ only depends on the initial position $Y_0 \in V$ and the initial rotor configuration $(r(v))_{v\in V}$. For an introduction to this topic we refer to Holroyd et al. (2008) and to Holroyd and Propp (2010).

We specialize now the ideas explained so far to the case, where Γ is the generalized Sierpinski graph $SG(m)$. This graph has seven different types of vertices: (1) vertex A, (2) vertex B, (3) vertex C with degree 2, each time $(m - 1)$ vertices with degree 4, (4) on the lower, (5) on the left and (6) the right boundary and (7) $\frac{1}{2}(m - 1)(m - 2)$

vertices with degree 6. For these types we show in Fig. 5 the ordering of the six directions named 1, 2, 3, 4, 5, 6, necessary to define the successor rule s. We define:

$$(1)\ s(n) := \begin{cases} 1 & : n = 2 \\ 2 & : n = 1 \end{cases} \qquad (2)\ s(n) := \begin{cases} 3 & : n = 4 \\ 4 & : n = 3 \end{cases}$$

$n \in \{1, 2\}$ $\qquad$ $n \in \{3, 4\}$

$$(3)\ s(n) := \begin{cases} 5 & : n = 6 \\ 6 & : n = 5 \end{cases} \qquad (4)\ s(n) := \begin{cases} 1 & : n = 4 \\ 2 & : n = 1 \\ 3 & : n = 2 \\ 4 & : n = 3 \end{cases}$$

$n \in \{5, 6\}$ $\qquad$ $n \in \{1, 2, 3, 4\}$

$$(5)\ s(n) := \begin{cases} 1 & : n = 6 \\ 2 & : n = 1 \\ 5 & : n = 2 \\ 6 & : n = 5 \end{cases} \qquad (6)\ s(n) := \begin{cases} 3 & : n = 6 \\ 4 & : n = 3 \\ 5 & : n = 4 \\ 6 & : n = 5 \end{cases}$$

$n \in \{1, 2, 5, 6\}$ $\qquad$ $n \in \{3, 4, 5, 6\}$

and furthermore

$$(7)\ s(n) := \begin{cases} 1 & : n = 6 \\ n + 1 & : n \in \{1, 2, 3, 4, 5\} \end{cases}$$

$n \in \{1, 2, 3, 4, 5, 6\}$.

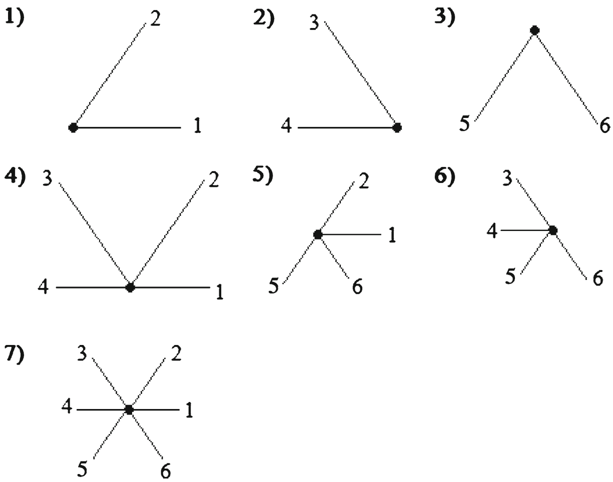

Fig. 5 Illustration of the successor rule $s(n)$ of the 7 different types of vertices

For the meaning of the direction indicating numbers 1 to 6 we refer to Fig. 5. In order to calculate the mean crossing time of $SG(m)$ we modify our rotor walk by adding the following rule: Whenever $Y_n \in \{B, C\}$, we put $Y_{n+1} := A$. This means that we move immediately to vertex A when we reach one of the vertices B or C. Furthermore, we define the initial vertex to be the vertex A, i.e. $Y_0 := A$ and denote by $N_n(W)$ the the number of times the rotor walk (Y_n) visits the set of vertices $W \subset V$ before time step n (but not including n), i.e.

$$N_n(W) = \#\{k < n : Y_k \in W\}.$$

We reformulate now a result from Holroyd and Propp (2010) (Theorem 2 therein), which connects the mean crossing time $\mathbb{E}T(m)$ of $SG(m)$ with the number $N_n(m) := N_n(\{B, C\})$ of visits of the rotor walk on $SG(m)$ of set of vertices $\{B, C\}$ for the special case of generalized Sierpinski graphs $SG(m)$:

Theorem 1 *There exists a finite constant $K(m)$ depending on m such that for all $n \in \mathbb{N}$ we have*

$$\left|\mathbb{E}T(m) - \left(\frac{n}{N_n(m)} - 1\right)\right| \leq \frac{K(m)}{N_n(m)}.$$

Thus, the average number of time steps $N_n(m)$ for the rotor walk on $SG(m)$ to get from A to B or C concentrates around the mean crossing time when n gets larger and larger. It is also known that the optimal constant $K(m)$ is given by

$$K(m) := \max_{v \in V^{(m)}} \mathbb{E}_v \tau_{\{B,C\}} + \frac{1}{2} \sum_{\substack{v \in V^{(m)} \\ u \in V \setminus \{B,C\}}} deg(u) p(u, v) \left|\mathbb{E}_u \tau_{\{B,C\}} - \mathbb{E}_v \tau_{\{B,C\}} - 1\right|,$$

where $V^{(m)}$ is the vertex set of $SG(m)$, which was denoted by $V_1^{(m)}$ in Section 2.1 and where $p(u, v)$ is the transition probability from vertex u to vertex v. In our case we have $p(u, v) = \frac{1}{deg(u)}$ for any vertex $u \in SG(m)$ ($deg(u)$ is the degree of vertex u, i.e. the number of adjacent edges) and so $K(m)$ reduces to

$$K(m) \leq \mathbb{E}T(m) + \frac{1}{2} \sum_{\substack{v \in V^{(m)} \\ u \sim v}} \left|\mathbb{E}_u \tau_{\{B,C\}} - \mathbb{E}_v \tau_{\{B,C\}} - 1\right|, \tag{4}$$

where $u \sim v$ means that u is a neighbour of v in $SG(m)$.

3.2 The Algorithm

In this section we present the algorithm used to produce the numerical values for the mean crossing times $\mathbb{E}T(m)$ which will be shown in Section 4. For the formulation of the algorithm, we will use again a pseudo-code, which should be readable without knowledge of a specific programming language.

Step 1: **Input.** The input of our algorithm is the model parameter m and a natural number n, which fixes the total number of steps the algorithm is running.

```
input m; input n;
```

Step 2: **Create the initial Rotor Matrix.** In order to avoid exceptional cases, we choose for the initial rotor position a random one, uniformly chosen from the set of all possible configurations. This configuration is coded up in the so-called *rotor matrix* R, which has 2 rows and $\frac{(m+1)(m+2)}{2}$ columns. In the first row the labels from 1 to $(m+1)(m+2)/2$ of all vertices of $SG(m)$ are written. In the second row we write for each vertex v a random rotor direction in $\{1, 2, 3, 4, 5, 6\}$ (compare with Section 3.1 and especially Fig. 5 for the enumeration of the possible rotor directions). For this we have to distinguish the 7 cases discussed above.

```
R:=CreateRandomRotorMatrix(2,(m+1)(m+2)/2);
```

A typical rotor matrix for $m = 2$ could be

$$R = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 5 & 1 & 6 & 4 \end{pmatrix},$$

whose meaning is explained in Fig. 6, first picture in the first row.

Step 3: **Perform the Rotor Walk.** In this step of the algorithm the rotor walk is running and the number $N_n(m)$ of hits of the vertices B and C is counted. In the programming code, this number is abbreviated by z. The current

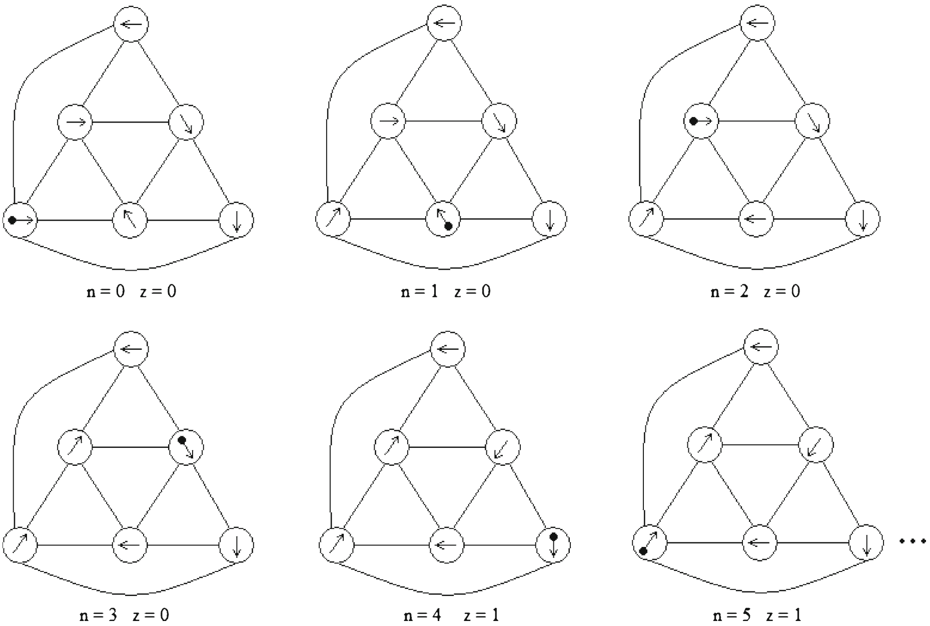

Fig. 6 A rotor walk for $n = 0, 1, 2, 3, 4, 5$; z counts the number of hits of the vertices B and C

position of the rotor walk will be the number w. We read out the entry $R(2, w)$ (second row and column with number w) of the rotor matrix R which is the direction assigned to vertex with number w. After this, we walk from w to the next vertex in direction associated with w and rotate the rotor of w according to the rule from Section 3.1. If we hit one of the vertices B or C we increase our counter z by 1 until we reach the previously fixed total number n of steps.

```
z:=0; w:=1; for k from 1 to n do
 direction:=ReadRotorMatrix([2,w]);
 w:=NextVertex(w,direction);
 RotateRotor(w);
 if ((w=B) OR (w=C)) then z:=z+1; end if;
end do;
```

For $m = 2$ and for the initial rotor matrix R from above the first 5 steps of the rotor walk are illustrated in Fig. 6.

Step 4: **Output.** In the last step of our algorithm we simply use the approximation from Theorem 1 and print the result of our simulation.

```
print(n/z-1);
```

It is important to note that the complexity of the algorithm does not grow with m, but with n and that with m growing complexity is main problem of the exact algorithm from Section 2.

In order to use Theorem 1 for the construction of confidence bounds (or maybe more precisely inequalities, because the confidence bounds we obtain here are not confidence bounds in the usual statistical sense) for the mean crossing time $\mathbb{E}T(m)$ of $SG(m)$ from the quantity $N_n(m)$ obtained by rotor-walk-simulation, we have to find an upper bound for the constant $K(m)$ given by Eq. (4). To this end we first note that the total number $|SG(m)|$ of verticies in the graph $SG(m)$ equals

$$|SG(m)| = 3 + 3(m-1) + \frac{1}{2}(m-1)(m-2),$$

which may be obtained simply by counting the number of vertices of type (1) to (7), see Section 3.1. Next, we apply (Hambly and Kumagai 2002, Corollary 2.3), which implies that the mean crossing time $\mathbb{E}T(m)$ is bounded from above by $m^2 \ln m$. This implies that the constant 4 is bounded from above by

$$\begin{aligned}
K(m) &\le \mathbb{E}T(m) + \frac{1}{2} \sum_{\substack{v \in SG(m) \\ u \sim v}} \underbrace{|\mathbb{E}_u \tau_{\{B,C\}} - \mathbb{E}_v \tau_{\{B,C\}} - 1|}_{\le \mathbb{E}T(m)} \\
&\le \mathbb{E}T(m) + \frac{1}{2} \cdot 6 |SG(m)| \mathbb{E}T(m) \\
&= \mathbb{E}T(m) \left(1 + 3\left(3 + 3(m-1) + \frac{1}{2}(m-1)(m-2)\right)\right) \\
&\le m^2 \left(\frac{3}{2}m^2 + \frac{9}{2}m + 4\right) \ln m.
\end{aligned}$$

Here we have used the fact that each vertex of $SG(m)$ has at most six neighbours. For $m = 2$ we have $K(2) \leq 53$, for $m = 10$, $K(10) \leq 45822$, for $m = 100$ we have $K(100) \leq 7.1169 \cdot 10^8$ and for $m = 1000$ we obtain $K(1000) \leq 1.0393 \cdot 10^{13}$.

For abbreviation we define

$$\hat{T}(m) := \frac{n}{N_n(m)} - 1,$$

$$\tilde{K}(m) := m^2 \left(\frac{3}{2} m^2 + \frac{9}{2} m + 4 \right) \ln m.$$

Now, the above considerations in connection with Theorem 1 show that the mean crossing time $\mathbb{E}T(m)$ of $SG(m)$ must lie in the interval

$$\left[\hat{T}(m) - \frac{\tilde{K}(m)}{N_n(m)}, \hat{T}(m) + \frac{\tilde{K}(m)}{N_n(m)} \right].$$

Moreover, for the walk and the spectral dimension of $sg(m)$ we have

$$\frac{\ln\left(\hat{T}(m) - \frac{\tilde{K}(m)}{N_n(m)} \right)}{\ln m} \leq d_W(m) \leq \frac{\ln\left(\hat{T}(m) + \frac{\tilde{K}(m)}{N_n(m)} \right)}{\ln m}$$

and

$$\frac{2(\ln m + \ln(m+1) - \ln 2)}{\ln\left(\hat{T}(m) + \frac{\tilde{K}(m)}{N_n(m)} \right)} \leq d_S(m) \leq \frac{2(\ln m + \ln(m+1) - \ln 2)}{\ln\left(\hat{T}(m) - \frac{\tilde{K}(m)}{N_n(m)} \right)}$$

by using Einstein's relation 1 from the introduction.

4 Results

We show at first the output of our exact algorithm developed in Section 2 for m from 2 to 37, which has been implemented in MAPLE in order to get explicit and exact fractions. For these values we restrict ourself to $m \in \{2, \ldots, 10\}$, because they become rather infeasible for growing m. For example we have for $m = 20$ the exact crossing time

$$\mathbb{E}T(20) = \frac{23404253201031219083581562604679692643002185140 30}{228095368167885399271082975905945502707768 6193}.$$

For m greater than 10 we show the numerical approximations of the mean crossing times $\mathbb{E}T(m)$, the exact values of $\mathbb{E}T(m)$ for m between 2 and 37 can be found in the technical note Freiberg and Thäle (2010), which is available on the second authors homepage. We like to point out that an explicit formula for $\mathbb{E}T(m)$ depending on m is still unknown and the task of finding it seems in view of the values quite intractable.

$$\mathbb{E}T(8) = \frac{69549468}{554417}, \; \mathbb{E}T(9) = \frac{5892677181}{35789105}, \; \mathbb{E}T(10) = \frac{27739580424985}{132150155017}$$

m	2	3	4	5	6	7	8	9	10
$\mathbb{E}T(m)$	5	$\frac{90}{7}$	$\frac{1030}{41}$	$\frac{8315}{197}$	$\frac{452739}{7025}$	$\frac{904836}{9823}$	$\mathbb{E}T(8)$	$\mathbb{E}T(9)$	$\mathbb{E}T(10)$
$d_W(m)$	2.3219	2.3247	2.3254	2.3255	2.3249	2.3244	2.3236	2.3228	2.3220
m	11	12	13	14	15	16	17	18	19
$\mathbb{E}T(m)$	261.39	319.24	383.60	454.60	532.35	616.97	708.56	807.21	913.02
$d_W(m)$	2.3212	2.3204	2.3196	2.3188	2.3180	2.3173	2.3165	2.3158	2.3152
m	20	21	22	23	24	25	26	27	28
$\mathbb{E}T(m)$	1026.1	1146.5	1274.2	1409.5	1552.3	1702.8	1860.9	2026.8	2200.5
$d_W(m)$	2.3145	2.3138	2.3132	2.3126	2.3119	2.3114	2.3108	2.3103	2.3097
m	29	30	31	32	33	34	35	36	37
$\mathbb{E}T(m)$	2382.1	2571.6	2769.0	2974.6	3188.2	3409.9	3639.9	3878.0	4124.5
$d_W(m)$	2.3091	2.3087	2.3082	2.3077	2.3072	2.3068	2.3063	2.3058	2.3054

For larger values of m, i.e. $m \in \{38, 39, \dots, 100\}$ we have used the power of MATLAB, which becomes noticeable especially in the step including the matrix inversion. More precisely, we have used the iterative generalized minimum residual method implemented in MATLAB.

At this place, we would like to give an interpretation of the results: In Hambly and Kumagai (2002), the authors have shown that the walk dimension of $sg(m)$ behaves asymptotically like

$$d_W(m) = 2 + \frac{\log\log m}{\log m} + \frac{O(1)}{\log m}, \qquad m \to \infty,$$

where $O(1)$ vanishes, as $m \to \infty$. Hence, $d_W(m) \to 2$, as $m \to \infty$. However, the rate of convergence is rather slow. It was observed in Freiberg and Thäle (2008) and confirmed by the values presented in Freiberg and Thäle (2010) that for small m the mean crossing times are much better approximated by $m^{2.3}$ than by m^2, which would be the correct asymptotical fit.

We now show the numerical results obtained for $\mathbb{E}T(m)$, $d_W(m)$ and $d_S(m)$ for m between 2 and 100 by simulation of the rotor walks introduced in Section 3. They are compared with the asymptotic upper bounds $m^2 \ln m$ for the mean crossing time, $2 + \frac{\ln\ln m}{\ln m}$ for the walk dimension and $2 - \frac{\ln\ln m}{\ln m}$ for the spectral dimension from Hambly and Kumagai (2002)—these are the green curves in Figs. 7 and 8:

m	2	10	20	30	40	50	60	70	80
$\mathbb{E}T(m)$	5.0000	209.91	1026.1	2571.5	4914.5	8102.7	12176	17163	23072
$d_W(m)$	2.3219	2.3220	2.3145	2.3087	2.3042	2.3006	2.2976	2.2951	2.2926

A comparison with the exact values for $\mathbb{E}T(m)$, $d_W(m)$ and $d_S(m)$ for m between 2 and 100 from above shows that the error between $\mathbb{E}T(m)$ and $\hat{T}(m)$ is of small order (here less than 19) and we have

$$\left| d_W(m) - \frac{\ln \hat{T}(m)}{\ln m} \right| \leq 10^{-4} \quad \text{and}$$

$$\left| d_S(m) - \frac{2(\ln m + \ln(m+1) - \ln 2)}{\ln \hat{T}} \right| \leq 10^{-4}.$$

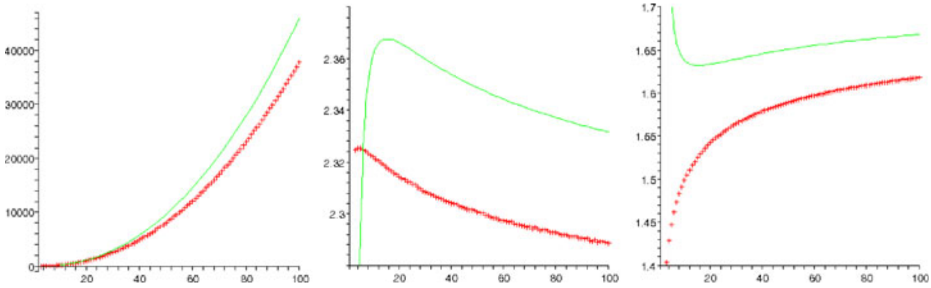

Fig. 7 Rotor-walk approximations for $\mathbb{E}T(m)$ (*left*), $d_W(m)$ (*middle*) and $d_S(m)$ (*right*) for $2 \leq m \leq 100$

For the simulations we have run our algorithm with the parameter $n = 150{,}000{,}000$.

The confidence bounds from Section 3.2 turn out to be reasonable good for small m, but unfortunately very rough for large ones (for the same parameter n). For $m = 2$, $m = 10$ and $m = 20$ we have

$$\begin{aligned}
\mathbb{E}T(2) &\in [5.0000, 5.0000], & \mathbb{E}T(10) &\in [209.84, 209.97],\\
\mathbb{E}T(20) &\in [1020.4, 1031.8], & d_Wsg(2) &\in [2.3219, 2.3219],\\
d_Wsg(10) &\in [2.3219, 2.3222], & d_Wsg(20) &\in [2.3126, 2.3163],\\
d_Ssg(2) &\in [1.3652, 1.3652], & d_Ssg(10) &\in [1.4989, 1.4991],\\
d_Ssg(20) &\in [1.5412, 1.5436].
\end{aligned}$$

The values for $m = 50$ are notedly more imprecise, i.e $\mathbb{E}T(50) \in [6524.9, 9677.9]$, $d_Wsg(50) \in [2.2452, 2.3460]$, $d_Ssg(50) \in [1.5583, 1.6282]$ and for $m = 100$ they are even unrealistic. The reason for this phenomenon lies in the constant $K(m)$, which grows to fast with m, i.e. $K(m)$ is of order $(3/2)m^4 \ln m$. Because of that it is an open and non-trivial problem to improve the confidence bounds from Section 3.2. Note, that a lower bound is provided by $m^2 \ln m$, which leaves some theoretical margin of m^2. Another way out is to increase considerably the number of steps the rotor walk is running. In this case the number $N_n(m)$ of hits of the vertices B and C grows and the bound of Theorem 1 gets sharper.

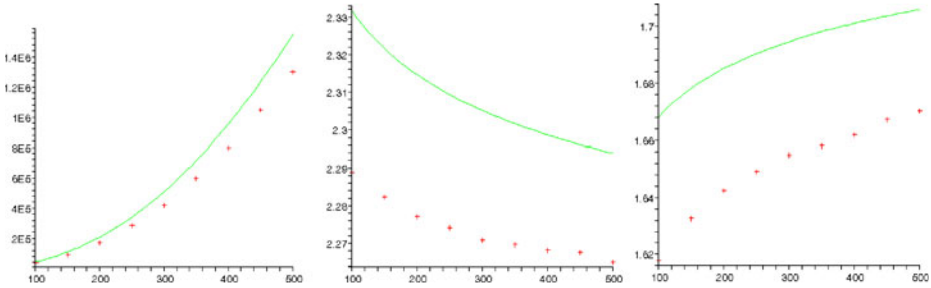

Fig. 8 Rotor-walk approximations for $\mathbb{E}T(m)$ (*left*), $d_W(m)$ (*middle*) and $d_S(m)$ (*right*) for $100 \leq m \leq 500$

In any case, the numerical results obtained with the rotor walk algorithm are quite accurate and much more precise than their random (Monte-Carlo) counterparts using the same total number of steps. When simulating a random walk on $SG(m)$ and using as an estimator for $\mathbb{E}T(m)$ the quotient of the total number of steps divided by the total number of hits of the vertices B and C we obtain with the same n as above

$$\begin{aligned} \mathbb{E}T(2) &\approx 6.9332, & \mathbb{E}T(10) &\approx 190.1471, & \mathbb{E}T(20) &\approx 691.8593,\\ d_W sg(2) &\approx 2.7935, & d_W sg(10) &\approx 2.2791, & d_W sg(20) &\approx 2.1829,\\ d_S sg(2) &\approx 1.1347, & d_S sg(10) &\approx 1.5272, & d_S sg(20) &\approx 1.6354. \end{aligned}$$

The comparison with the true values shows that these estimates are considerably more biased than the values obtained via rotor walks and that they are actually not contained in the above calculated confidence intervals. This is due to the fact that the variance of the crossing times grows very fast compared with the number of hits of the vertices B and C (it is at least of order $m^4 \ln^2 m$).

For m between 100 and 500 we have simulated with the rotor-walk method the following values as approximations for the mean crossing times and the walk dimensions:

m	100	150	200	250	300
$\mathbb{E}T(m)$	37,835	92,592	173,730	284,090	421,940
$d_W(m)$	2.2889	2.2823	2.2772	2.2742	2.2709
m	350	400	450	500	–
$\mathbb{E}T(m)$	596,660	797,870	1,049,000	1,298,700	–
$d_W(m)$	2.2697	2.2682	2.2676	2.2651	–

They may be compared with the values computed with the exact algorithm combined with numerical matrix inversion techniques, which yields

$$\begin{aligned} \left|\mathbb{E}T(m) - \hat{T}(m)\right| &\leq 261\,400\\ \left|d_W(m) - \frac{\ln \hat{T}(m)}{\ln m}\right| &\leq 0.0161,\\ \left|d_S(m) - \frac{2(\ln m + \ln(m+1) - \ln 2)}{\ln \hat{T}}\right| &\leq 0.0245. \end{aligned}$$

Note, that $d_S(m)$ can be computed from $d_H(m)$ and $d_W(m)$ by Eq. 1. Thus, the approximations for the walk and spectral dimension are fairly good, even if—due to the constants $K(m)$—the confidence bounds do not lead to reasonable results. For the computations for large the m we have used the parameter $n = 300{,}000{,}000$.

Appendix: MATLAB Code

We do not present here the full MATLAB code of our algorithm, but only the crucial steps that were not explained in detail in Section 2.

```
#Step 3: Calculate the connection matrix
C=zeros(2,m*m-1);
for k=1:m
 a=0; u=1;
 for l=(3*(k-1)*m-(3*(k-1)*(k-2)/2-1)):(3*k*m-3/2*k*(k-1))
 if (l~=((3*k*m-3/2*k*(k-1))-1))
  if (a<3) a=a+1; else a=1; u=u+1; end;
  b=1;
  for n=1:(m*m-1) if (C(1,n)~=0) b=b+1; end; end;
  if (((a==2)|(a==3))&(b<=m^2-1)) C(1,b)=l;
   if ((a==2)&(l<(3*k*m-3/2*k*(k-1))-1)) C(2,b)=l+2; end;
   if ((a==2)&(l>=(3*k*m-3/2*k*(k-1))-1))
      C(2,b)=3*m*(k+1)-3/2*k*(k+1)-1; end;
   v=3*k*m-(3/2*k*(k-1)+1)+3*(u-1)+2;
   w=3*k*m-(3/2*k*(k-1)+1)+3*(u-2)+3;
   if ((a==3)&(u==1)&(l~=(3*(k-1)*m-(3*(k-1)*(k-2)/2-1))))
      C(2,b)=v; end;
   if ((a==3)&(u>1)&(l~=(3*(k-1)*m-(3*(k-1)*(k-2)/2-1))))
      C(2,b)=w; end;
  end;
 end;
 end;
end;

#Step 7: Remove row and column associated with the corners B
and C a=0;
for k=1:(m*m-1)
 if C(1,k)<(3*m-1) a=a+1; end;
end;
M(col,:)=[]; M(:,col)=[]; M(3*m-1-a,:)=[]; M(:,3*m-1-a)=[];
```

References

Barlow M (1998) Diffusions on fractals. Lecture notes in mathematics, vol 1690. Springer, New York

Falconer KJ (2003) Fractal geometry, 2nd edn. Wiley, New York

Freiberg U (2005) Analysis on fractal objects. Meccanica 40:419–436

Freiberg U (2011) Einstein relation on fractal objects. To appear in Discrete Contin Dyn Syst, Ser B

Freiberg U, Thäle C (2008) A Markov chain algorithm for determining crossing times through nested graphs. In: DMTCS proc. AI, pp 505–522

Freiberg U, Thäle C (2010) Exact values for the mean crossing times of generalized sierpinski graphs. Technical note

Given JA, Mandelbrot BB (1983) Diffusion on fractal lattices and the fractal Einstein relation. J Phys A 16:L565–L569

Hambly BM, Kumagai T (2002) Asymptotics for the spectral and walk dimension as fractals approach Euclidean space. Fractals 10:403–412

Hilfer R, Blumen A (1984) Renormalization of Sierpinski-type fractals. J Phys A 17:L537–L545

Hilfer R, Blumen A (1988) Probabilistic interpretation of the Einstein relation. Phys Rev A 37:578–581

Holroyd AE, Levine L, Mészáros K, Peres Y, Propp J, Wilson DB (2008) Chip-firing and rotor-routing on directed graphs. Prog Probab 60:331–364

Holroyd AE, Propp J (2010) Rotor walks and Markov chains. Algorithmic Probability and Combinatorics. Contemp Math 520:105–126

Hutchinson JE (1981) Fractals and self–similarity. Indiana Univ Math J. 30:713–747

Kallenberg O (2002) Foundations of modern probability, 2nd edn. Springer, New York

Kigami J, Lapidus ML (1993) Weyl's problem for the spectral distribution of Laplacians on p.c.f. self–similar fractals. Commun Math Phys 158:93–125

Kigami J (2001) Analysis on fractals. Cambridge University Press, Cambridge, UK

Kemeny JG, Snell JL (1983) Finite Markov chains. Springer, New York

Lindstrøm T (1990) Brownian motion on nested fractals. Mem Am Math Soc 420

Telcs A (2006) The Einstein relation for random walks on graphs. J Stat Phys 122:617–645

Zhou XY (1993) The resistance dimension, random walk dimension and fractal dimension. J Theor Probab 6:635–652