

Random walks for on-lattice DLA simulations

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1988 J. Phys. A: Math. Gen. 21 827

(<http://iopscience.iop.org/0305-4470/21/3/038>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 129.252.86.83

The article was downloaded on 01/06/2010 at 06:36

Please note that [terms and conditions apply](#).

Random walks for on-lattice DLA simulations

Richard Friedberg[†] and Olivier Martin[‡]

Department of Physics, University of Illinois at Urbana-Champaign, 1110 West Green Street, Urbana, IL 61801, USA

Received 31 March 1987

Abstract. Two efficient ways to accelerate on-lattice DLA simulations without introducing any bias are presented. The first requires one-dimensional tables but little arithmetic; the second needs no tables but necessitates the evaluation of a few elementary functions.

1. Introduction

When performing a Monte Carlo or simulating a stochastic process, it is almost always necessary to have an unbiased estimator for a Green function, and in addition, each realisation of the estimator should be positive, otherwise one must introduce negative weights and reliability is usually lost. Here, we consider the case of on-lattice DLA simulations [1, 2]. In the past two years, the cluster sizes achieved have become very large and a cross-over from a fractal to non-fractal form has been seen [3, 4]. It is believed that the underlying lattice anisotropy is responsible for this behaviour. However, the simulations have been performed by using a mixture of on- and off-lattice methods, so that one does not really know quantitatively the effect of the lattice grid. This reason for using this mixture of methods is a practical one: a single stepping random walk is very slow, and the walker takes a long time before it either sticks to the cluster or drifts far enough away so that it can be thrown away. To accelerate this random walk, one needs to increase the effective step size at each iteration. One method used is to keep the steps on axis, but simply increase their size, depending on how far one is from the cluster. This probably enhances significantly the effect of the anisotropy of the lattice, and thus should be avoided. Ball and Brady [3] introduced an algorithm which increases the step size by using the distribution of the first crossings of a square centred on the current location of the diffusing particle. Ball and Brady used the continuum Green function for this purpose and thus introduced a small bias. In this paper we show how one can use instead the lattice Green function. This distribution or Green function is given in the next section. Its tabulation for a number of square sizes provides a fast and unbiased DLA algorithm. In the subsequent section, we show how the use of tables can be avoided by using a decomposition of the relevant Green function into analytically simple pieces which are all positive.

[†] Visiting from Barnard College and Columbia University, New York, NY, USA.

[‡] Present address: Department of Physics, City College, City University of New York, NY 10031, USA.

2. First-passage Green function for a square

Consider a random walker (particle) on a two-dimensional lattice. At each time step, it hops randomly to a nearest-neighbour site. If the new position is adjacent to the cluster, the particle sticks, joining the cluster, and a new walk is initiated infinitely far away from the cluster. Of course, in practice, this means starting the new walker at a distance from the cluster much larger than its diameter and this introduces a small bias. Reducing the bias is very costly in CPU time if one is single stepping the particle, so the step size should be increased in some way. Since the problem on the square lattice is rather regular, it is possible in fact to start the random walker on a square surrounding the cluster rather than infinitely far away without introducing any bias. This is done by finding the distribution on the square of the first passages of a particle coming in from infinity, as shown in [5]. The square can be small enough so that it just encloses the cluster. This removes the inefficiency of having the random walker diffuse from infinity, and it can be used whenever the particle is far enough away from the cluster. However, nearer to the cluster this method cannot be used and one must find another way to make large steps. This is particularly important if the cluster is fractal and has holes on all length scales.

Given a point and a surrounding path on the lattice, we wish to find the distribution of the location where the particle encounters the path for the first time. We require that the cluster lie entirely outside the closed path. This distribution is determined by the solution to the discrete Laplace equation with a source term at the point of origin and zero boundary conditions on the enclosing path

$$\nabla^2 G(x, y) = -\delta_{x,x_0} \delta_{y,y_0}. \tag{1}$$

The distribution of interest is then given by the ‘current’ flowing into the points on the path [6, 7]. This is particularly simple for a square centred on the point of origin: the current is given by the gradient which is just the value of the adjacent site inside the square because the boundary conditions are $G = 0$. If we consider the new stochastic process where at each step we select an appropriate square size around the current particle location and choose a site on the edge with the above probability distribution, then one obtains the proper probability for sticking to the surface of the cluster at each site, i.e. the algorithm is unbiased. Since the step size can grow with the distance to the sticking centres, the algorithm is very efficient.

In [3], equation (1) was solved in the continuum. We now calculate the Green function $G(x, y)$ for the lattice problem. First we specify a square of width $2s$, thus containing $2s + 1$ points on each edge with the origin as indicated in figure 1. The solution of (1) with zero boundary conditions is of the form

$$G(x, y) = \sum_{m=0}^{m=2s-1} c_m \sin\left(\frac{m\pi x}{2s}\right) \sinh[\bar{k}_m(s - |y|)] \tag{2}$$

where

$$\cos\left(\frac{m\pi}{2s}\right) + \cosh \bar{k}_m = 2. \tag{3}$$

Imposing the condition of (1) at $y = 0$ leads to

$$c_{2n} = 0 \quad c_{2n+1} = (-1)^n [2s \cosh(\bar{k}_{2n+1}s) \sinh \bar{k}_{2n+1}]^{-1}. \tag{4}$$

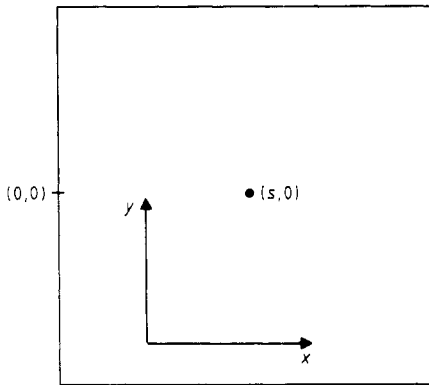


Figure 1. Origin of axes for the square centred at the current site.

In view of the symmetry, we need only consider one edge of the square, and the probability distribution of the first-passage walks is $G(1, y)$, where again by symmetry we can restrict y to be in $[0, s-1]$. Calculating $G(1, y)$ for these values of y requires s^2 operations, and the simplest thing to do is to tabulate these values for several values of s . The algorithm now proceeds by picking at random one of the eight half-sides of the square and then choosing y being careful not to overcount $y=0$.

Once $G(1, y)$ is obtained, one can create a table of values of the cumulant distribution and use this to generate values of y with the right probability. In practice, one may want to have tables for $s=2^k$ so that the effective step size can be large without excessive memory requirements. The choosing of y can be accelerated by using in addition an approximate table following standard practice. Another possibility is to use an accept-reject procedure. A simple trial distribution for y is obtained by generating a point uniformly in a quarter circle, $0 < r < 1$, $0 < \theta < \pi/2$, and taking $y = (s - \frac{1}{2})\sqrt{r} \cos \theta$ to the nearest integer. The trial probability $T(y)$ is easily calculated. One is to accept y with a probability proportional to $G(y)/T(y)$, the proportionality constant being most conveniently taken to force $y=0$ to be always accepted. We have coded this algorithm and find that the average acceptance is very insensitive to s , and is close to 75%.

3. Positive decomposition of the Green function

The use of tables for the cumulant of $G(y)$ in the DLA problem is computationally efficient because the entries are calculated once and for all at the beginning of the run and are used a large number of times in growing large clusters. In addition, since y is the only argument of the tables, the storage space is not excessive in general. However, there are other problems where it is not as practical to use this approach, either because the tables are not used as frequently, or one needs many more values of s . Since it is generally not efficient to tabulate G as a function of too many variables, one should find another method which is more practical. In addition, it is not worth tabulating G for very large values of s as the space required is large and such values are not used very often. Our suggestion is to obtain a representation of G in terms of a sum of positive analytically simple terms. One then samples these terms using an accept-reject method so that no tables are needed. We illustrate this method here in

the context of DLA. The Green function of (2) can be written as

$$G(y) = \sum_{n=0}^{s-1} \sum_{n'=-n}^n G_{n,n'}(y) \tag{5}$$

where

$$G_{n,n'}(y) = \frac{1}{2s} \sin \frac{\pi}{2s} (-1)^n \cos \frac{n'\pi}{s} h_n(y) \tag{6}$$

with

$$h_n(y) = \frac{\sinh[\bar{k}_{2n+1}(s-|y|)]}{\sinh \bar{k}_{2n+1} \cosh(\bar{k}_{2n+1}s)}. \tag{7}$$

If $G_{n,n'}(y)$ were always positive, we could realise G stochastically by selecting n, n' with weight $\sum_y G_{n,n'}(y)$, and then selecting y with weight $G_{n,n'}(y)$. Since there are negative weights, we try to eliminate them by decomposing G into positive linear combinations of the $G_{n,n'}$.

In figure 2, each (n, n') is represented by a dot. Our idea is to group the dots into cells each of positive weight. Since negative values of n' are not represented separately, each dot with $n' > 0$ must receive total weight $2G_{nn'}$, and those with $n' = 0$ only $G_{nn'}$. Seeing that $h_n(y)$ is a decreasing function of n , one's first thought is to pair two successive values of n for the same n' , so that the weight of the combination is proportional to

$$f_n(y) = h_n(y) - h_{n+1}(y). \tag{8}$$

If all dots are included, however, the prefactor will sometimes be negative. A second idea is to group four dots in an oblique parallelogram with weight

$$G_{2j,2j'}(y) + G_{2j+1,2j'}(y) + G_{2j,2j'+1}(y) + G_{2j+1,2j'+1}(y) = \frac{1}{2s} \sin \frac{\pi}{2s} \left(f_{2j}(y) \cos \frac{2j'\pi}{s} - f_{2j'}(y) \cos \frac{(2j'+1)\pi}{s} \right) \tag{9}$$

but this fails to count $n' > 0$ twice as often as $n' = 0$.

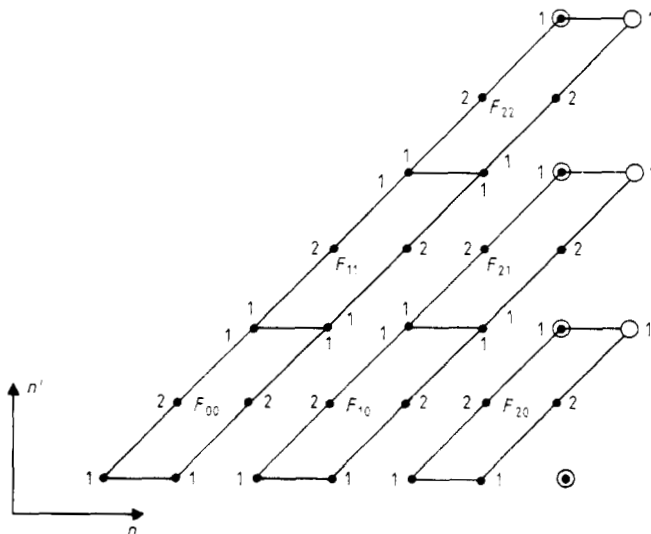


Figure 2. Decomposition of the Green function into cells $G_{n,n'}$ on the n, n' grid.

We are thus led to the scheme shown in figure 2, in which each cell contains six dots with the middle pair counted twice. This cell is the sum of the one described in (9) with an overlapping one having $2j \rightarrow 2j + 1, 2j' \rightarrow 2j' + 1$. Its weight is thus

$$F_{jj'}(y) = \frac{1}{2s} \sin \frac{\pi}{2s} \left(f_{2j}(y) \cos \frac{2j'\pi}{s} - 2f_{2j+1}(y) \cos \frac{(2j'+1)\pi}{s} + f_{2j+2}(y) \cos \frac{(2j'+2)\pi}{s} \right). \tag{10}$$

If we sum over all cells for which $2j' \leq 2j \leq s - 3$, each dot in figure 2 is counted the right number of times except some at the extreme right. Assuming that s is odd, we find that all dots with $n = s - 1, n' = 0, 2, \dots, s - 1$ (circled dots) are undercounted by 1, and non-existent dots with $n = s, n' = 2, 4, \dots, s - 1$ (empty circles) are overcounted by 1. The total error in $G(y)$ is therefore

$$- \sum_{j'=0}^{(s-1)/2} G_{s-1,2j'}(y) + \sum_{j=1}^{(s-1)/2} G_{s,2j}(y) = \frac{1}{4s} \sin \frac{\pi}{2s} (-h_{s-1}(y) + h_s(y)) = 0 \tag{11}$$

since $\bar{k}_{2s+1} = \bar{k}_{2s-1}$ by (3).

We have found empirically that (10) is not always positive, so to obtain a positive decomposition, we sum analytically $F_{j,j'}(y)$ over j' for fixed j . We have (for odd s)

$$G(y) = \sum_{j=0}^{(s-3)/2} F_j(y) \tag{12}$$

where

$$\begin{aligned} F_j(y) &= \sum_{j'} F_{j,j'}(y) \\ &= \frac{1}{8s} \left(\cos \frac{\pi}{2s} \right)^{-1} \left[f_{2j}(y) \left(\sin \frac{(2j+1)\pi}{s} + \sin \frac{\pi}{s} \right) \right. \\ &\quad \left. - 2f_{2j+1}(y) \sin \frac{(2j+2)\pi}{s} + f_{2j+2}(y) \left(\sin \frac{(2j+3)\pi}{s} - \sin \frac{\pi}{s} \right) \right]. \end{aligned} \tag{13}$$

The equivalence of (12) and (13) with (7) to (2) with (4) follows immediately by combining terms in (12) having the same subscript for h . The double decomposition (5) was needed only for motivation. We have checked numerically that (13) is positive for $s < 400$, and have no doubt that it is generally true, but we have not been able to prove positivity for all s .

To make a stochastic selection of j , we need the analytic sum

$$\begin{aligned} \bar{h}_n &= \sum_{y=-(s-1)}^{s-1} h_n(y) = \frac{1}{2 \sinh^2(\bar{k}_{2n+1}/2)} \left(1 - \frac{1}{\cosh(\bar{k}_{2n+1}s)} \right) \\ &= \frac{1}{2 \sin^2[(2n+1)\pi/4s]} \left(1 - \frac{1}{\cosh(\bar{k}_{2n+1}s)} \right) \end{aligned} \tag{14}$$

by (3). The probability $p_j = 4 \sum_y F_j(y)$ to be associated with each j (assuming we have picked one of the four sides at random) is given by (13) with h_n replaced by $4\bar{h}_n$. To pick j quickly with a good acceptance rate, one needs to have a rapidly invertible reasonable estimate of p_j . We have found in practice that the first column has a much larger probability than the others and that p_j falls off at least as fast as $1/j(j+1)$. Using this dependence for all $j \neq 0$ (this distribution is easily inverted), we find that the acceptance rate is quite high, greater than 90%, even for very large s .

The choice of j can be made very rapid by the following procedure. A small coefficient a is chosen so that $a/j(j+1) > p_j$ for $0 < j \leq (s-3)/2$. We have found that $a = 0.1$ is sufficient. A random number r is chosen from 0 to 1. If $r > a$, we set $j = 0$. If $r < a$, we use r/a to select a $j > 0$ with probability $1/j(j+1)$. We accept this j with probability $p_j j(j+1)a^{-1}$ if $j < (s-1)/2$, or 0 if $j \geq (s-1)/2$. If this trial fails we set $j = 0$ with no further calculation. Since the first choice gives $j = 0$ for 90% of the time, this part of the calculation usually requires no work at all.

Finally, once a value of j is accepted, it is necessary to select y with distribution $F_j(y)/p_j$ using an accept-reject procedure. If one uses the trial distribution given in the previous section, one finds that the acceptance probability is sometimes greater than 1. This is easily cured by rescaling the acceptance probability by a factor such as 0.95. We have found numerically that the optimal factor converges to 1 as $s \rightarrow \infty$, but the acceptance rate is sufficiently high ($>70\%$ for all s) that it is not worth the extra effort to optimise this factor. Generally speaking, the stochastic method allows one to use any value of s , and has absolutely no memory requirements. However, it does require the frequent calculation of elementary functions but this may be the only method available for some problems less regular than DLA.

4. Conclusion

In summary, we have presented two algorithms for accelerating on-lattice DLA simulations without introducing any bias. The first one relies on the tabulation of a Green function: it is computationally very fast but has some memory requirements. The second does a sampling which relies on a tractable decomposition of the Green function into positive pieces: it needs no tables, but requires more evaluations of elementary functions.

Acknowledgments

We are grateful to Peter Garik for an enlightening discussion. RF wishes to thank the Department of Physics of the University of Illinois at Urbana-Champaign for its hospitality, and OM acknowledges support through grant NSF-PHY-82-01948.

References

- [1] Witten T A and Sander L M 1981 *Phys. Rev. Lett.* **47** 1400
- [2] Meakin P 1983 *Phys. Rev. A* **27** 604, 1495
- [3] Ball R C and Brady R M 1985 *J. Phys. A: Math. Gen.* **18** L809
- [4] Meakin P 1985 *J. Phys. A: Math. Gen.* **18** L661; 1986 *Phys. Rev. A* **33** 3371
- [5] Friedberg R 1987 *J. Phys. A: Math. Gen.* **20** 4823
- [6] McCrea W H and Whipple F J N 1940 *Proc. R. Soc. Edin.* **60** 281
- [7] Garik P, Richter R, Hautman J and Ramanlal P 1985 *Phys. Rev. A* **32** 3156