Algorithm for generating a Brownian motion on a sphere

Tobias Carlsson[1], Tobias Ekholm[2], and Christer Elvingson[1,*]

January 29, 2010

[1]*Department of Physical and Analytical Chemistry, Physical Chemistry, Uppsala University, Box 259, S-751 05 Uppsala, Sweden*

[2]*Department of Mathematics Uppsala University, Box 480, S-751 06 Uppsala, Sweden*

Keywords: Brownian motion on a sphere; random walk; diffusion; S^2; Brownian dynamics

*Corresponding author (e-mail: Christer.Elvingson@fki.uu.se)

Abstract

We present a new algorithm for generation of a random walk on a 2-dimensional sphere. The algorithm is obtained by viewing the 2-sphere as the equator in the 3-sphere surrounded by an infinitesimally thin band with boundary which reflects Brownian particles and then sapplying known effective methods for generating Brownian motion on the 3-sphere. To test the method, the diffusion coefficient was calculated in computer simulations using the new algorithm and, for copmparision, also using a commonly used method in which the particle takes a Brownian step in the tangent plane to the 2-sphere and is then projected back to the spherical surface. The two methods are in good agreement for short time steps, while the method presented in this paper continues to give good results also for larger time steps, when the alternative method becomes unstable.

1 Introduction

Brownian motion is of fundamental importance for understanding a large number of processes in almost any scientific field.[1] The translational diffusion in ordinary Euclidean space can be modelled by a random walk generated from a Gaussian distribution.[2] In many applications, however, one is instead interested in diffusion processes on a curved surface, usually the sphere, S^2, of fixed radius in three dimensional Euclidean space, $\mathbb{R}^3$. The examples range from diffusion of proteins or fluorescent marker molecules in cell membranes or on other types of particles,[3–6] to the swimming of bacteria[7] or the migration of elephant seals in the oceans.[8] The random walk on S^2 has also been analyzed in connection with spin dynamics[9–11] or surface smoothing in computer graphics applications[12], as well as in more general statistical analyses.[13–14] An important application of random motion is in Brownian dynamics simulations when modeling the structure and dynamic processes in colloidal or polymer systems.[15–17] The random part of this motion is generated by drawing random numbers from the solution of the corresponding diffusion equation for the particular geometry. In flat $\mathbb{R}^n$, this is reasonably straightforward by generating Gaussian random numbers.[16]

For a diffusion process on unit sphere, S^n, in $(n+1)$-dimensional Euclidean space $\mathbb{R}^{n+1}$ however, it is more involved. For particles moving on S^3, a random walk can be generated for any physically reasonable set of parameters by defining the diffusion coefficient, the radius, and the time step.[18] On S^2, however, the standard solution of the diffusion equation is not in a form suitable for easily generating random numbers necessary in a Brownian dy-

namics simulation. On spheres of arbitrary dimensions, S^n, there are closed formulas for odd but not for even n, see e.g. Caillol[19] or Taylor[20]. One can also note that in Monte Carlo simulations there is no explicit time dependence, and the random displacements can be chosen more freely, as long as one samples the phase space in a correct way, given a certain ensemble.[21] In simulations mimicking the diffusion of particles on S^2, a commonly used approach is to approximate the curved surface with its tangent plane and use a two-dimensional Gaussian distribution to generate the random walk. This is valid for small time steps and large radii. We propose below a novel and more exact method to generate a random walk on S^2 useful in molecular simulations. Our method is obtained by viewing S^2 as an equator of the 3-dimensional sphere S^3 surrounded by a thin band $S^2 \times [-\epsilon, \epsilon]$ centered on the equator, letting the boundary of the band reflect Brownian particles, using known effective methods to generate random walks on S^3,[18] and taking the limit as $\epsilon \to 0$.

2 Theory

To generate a diffusion process in a computer simulation, one can draw random numbers from the probability distribution corresponding to the solution of the diffusion equation given the appropriate boundary conditions. In $\mathbb{R}^n$ this would mean drawing n Gaussian random numbers corresponding to the random walk in each dimension.[15–17,21] For other geometries, however, there might not be a closed expression for the distribution function or the solution can be a slowly converging sum. In this paper we are concerned with spheres

$S^n = \{(x_1, \ldots, x_{n+1}) \in \mathbb{R}^{n+1} \colon x_1^2 + \cdots + x_{n+1}^2 = R^2\}$, where R is the radius of the sphere. A special case which is of great general interest is the motion of particles confined to the spherical surface, S^2. An often used procedure is to approximate the sphere locally with a planar surface and to use Gaussian random numbers to generate a random walk in the tangent plane followed by a projection of the particle to get back to S^2.

A more exact procedure starts from the diffusion equation on S^2:

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} = D_0 \nabla^2 \rho(\mathbf{r}, t) \tag{2.1}$$

with the initial condition

$$\rho(\mathbf{r}, 0) = \delta(\mathbf{r}_0, \mathbf{r}) \delta(t, 0) \tag{2.2}$$

where ∇^2 is the Laplace-Beltrami operator and D_0 is the diffusion coefficient. A separation of variables leads to a Fourier series solution which together with the addition theorem for spherical harmonics results in[22]

$$\rho(\mathbf{r}, t) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi r^2} e^{-D_0 l(l+1)t/r^2} P_l(\cos\theta) \tag{2.3}$$

where r is the S^2-radius and $P_l(\cos\theta)$ is the l:th Legendre polynomial. This solution, although exact, converges slowly and also oscillates for typical values of the parameters even if a large number of terms are taken into account. Equation (2.3) is thus not suitable to use as a function for drawing random numbers to generate a random walk. Because of this other approximate methods, in particular the tangent plane method mentioned above, are more commonly used.

We are interested in finding a solution to Eq. (2.1) which can be used in molecular simulations. On S^3 one obtains, using the relation between the Laplacian on the sphere and in the ambient Euclidean space,[18–20]

$$\rho(\mathbf{r},t) = \frac{A(\beta)}{\sin\theta}\sum_{k=-\infty}^{k=\infty}(\theta+2k\pi)e^{-(\theta+2k\pi)^2/4\beta} \tag{2.4}$$

where θ is the angular distance between two points on S^3, A is a normalisation factor, and $\beta = D_0 t/R^2$. For physically reasonable values of β the sum is rapidly converging and can be used for generating a random walk on S^3.

The expression which corresponds to Eq. (2.4) on S^3 for S^2 can be derived as follows. Using the Dirichlet-Mehler integral[23]

$$P_l(\cos\theta) = \frac{\sqrt{2}}{\pi}\int_\theta^\pi \frac{\sin(l+\frac{1}{2})s\,ds}{\sqrt{\cos\theta-\cos s}} \tag{2.5}$$

and the Jacobi theta-function relation[24,25]

$$e^{-\alpha/4}\sum_{n=0}^{\infty}e^{-n(n+1)\alpha}\cos(2n+1)\kappa = \sqrt{\frac{\pi}{4\alpha}}\sum_{n=-\infty}^{\infty}(-1)^n e^{-(\kappa+\pi n)^2/\alpha} \tag{2.6}$$

gives the following expression[25] for the solution of the diffusion equation on S^2

$$\rho(\mathbf{r},t) = K\sum_{k=-\infty}^{\infty}(-1)^k\int_\theta^\pi \frac{1}{\sqrt{\cos\theta-\cos s}}(s+2\pi k)\exp^{-\frac{r^2(s+2\pi k)^2}{4D_0 t}}ds \tag{2.7}$$

which, although having a rapidly decreasing exponential term with increasing k in line with Eq. (2.4), also contains a non-trivial integral which makes it difficult to use as a source for random number generation and it is thus not a straightforward task to find an efficient and robust way to generate a random walk on S^2. Our approach here is to use Eq. (2.4) in combination

with geometry in a thin band around the equator on S^3 rather than Eq. (2.7) to solve this problem.s

The spherical surface, S^2, can be obtained from the three-dimensional surface S^3 in $\mathbb{R}^4$, constraining one of the coordinates to a constant value. For instance, letting a plane in $\mathbb{R}^4$ orthogonal to the x_4-axis cut this axis at $x_4 = R\cos\varphi$, would generate the surface $x_1^2+x_2^2+x_3^2 = r^2$ where the radius of the S^2-sphere $r = R\sin\varphi$. In essence, our algorithm produces a random walk on the equator, corresponding to $R = 1$ and $\varphi = \pi/2$ but since we want the algorithm to be useful in many types of simulations and since we have specific applications in mind, we have described a more general case, transporting the Brownian motion to spheres with an arbitrary fixed x_4-coordinate.

Since we already know how to generate a Brownian motion on S^3,[18] we can now use this to also obtain a Brownian motion on S^2. In the Appendix we give a more detailed argument motivating the procedure which is described in an algorithmic way below.

We begin with a point $\mathbf{p}_i = (\mathbf{p}_1, p_4)$ where $\mathbf{p}_1 \in S^2$. Mapping $\mathbf{p}_i$ onto the equator of S^3 gives a point $\tilde{\mathbf{p}}_1$

$$\tilde{\mathbf{p}}_1 = R\frac{\mathbf{p}_i - (\mathbf{p}_i \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}}{|\mathbf{p}_i - (\mathbf{p}_i \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}|} \tag{2.8}$$

where $\hat{\mathbf{n}}$ is the unit vector in the x_4-direction. Let the particle at $\tilde{\mathbf{p}}_1$ now make a random move on S^3 according to the algorithm described in ref. 18. This will give a new position $\mathbf{q} = (\mathbf{q}_1, q_4)$ on S^3. Now map the point $\mathbf{q}$ along the geodesic through $\mathbf{q}$ and $\hat{\mathbf{n}}$ onto the equator of S^3. Finally bring this point back to S^2 generating the new position on S^2

$$\mathbf{p}_{new} = \left(\frac{r}{|\ \mathbf{q}_1|}\mathbf{q}_1, p_4\right) \tag{2.9}$$

In this way we can generate a Brownian motion on S^2 from a random walk on S^3 (algorithm 1).

To test the algorithm, we will in the next section compare it to the method of generating a random walk in the tangent plane at the position of the particle, drawing Gaussian random numbers in $\mathbb{R}^2$, after which the particle is projected onto the surface of the sphere along the line connecting the centre of the sphere with the new particle position in the tangent plane (algorithm 2). For comparison, we also tested an alternative way of using a random walk in the tangent plane using Gaussian random numbers (algorithm 3), by scaling the distance so that the final position in the plane would correspond to a distance along the geodesic on S^2 (after projection) equal to the magnitude of the random motion in algorithm 2. These two methods of projecting the particle from the tangent plane back to the surface of the sphere gave similar (though not identical) results for the diffusion coefficient, and we present below the results moving the particles in the tangent plane without any scaling.

Generating a Brownian motion for various time steps and S^2-radii, we have investigated the stability and accuracy of the methods. The analysis of the resulting trajectories was made by fitting the random walk to the function[19]

$$\langle \cos\theta(t) \rangle = e^{-2Dt/r^2} \tag{2.10}$$

For particles performing a Brownian motion on the equatorial S^2 in S^3 constructed as described above, the diffusion coefficient D will agree with the corresponding quantity in S^3. Our mapping of the random walk on the equa-

torial S^2 to a sphere of radius $R\sin\varphi$ reduces all distances by a factor $R\sin\varphi$ and it follows that one has to scale the obtained diffusion coefficient, $D^{'}$, as follows: $D = D^{'}/\sin^2\varphi$, since in determining the diffusion coefficient, one in principle analyzes the distance squared as a function of time.

3 Results and discussion

To test the new method for generating a diffusion process on S^2 described above, we first simulated a random walk using Eqs. (2.8)-(2.9) for various angles φ, where φ is the angle which determines the distance from the x_4-axis. All simulations consisted of 200 noninteracting particles, each making $1 \cdot 10^6$ steps, and the results are averages over 100 independent trajectories. The results are presented as D/D_0 as a function of the dimensionless time step $\tau = \Delta t D_0/r^2$. We can see in Table I that the diffusion coefficient is independent of φ, within the statistical fluctuations, as it should be. This is a necessary test since the present algorithm will be used in simulations of colloidal and polymer systems on S^3,[26] with the possibility to confine the corresponding solution between two surfaces.[27−28] On S^3 the confining surfaces will be 2-spheres, where the distance between the surfaces is $2(\pi-\varphi)$. For the case with adsorbed and movable molecules or charges on these confining surfaces, the algorithm above can be used to simulate the random motion of such particles. When generating the random motion in the tangent plane, the angle φ does not enter the calculations, and for a large enough radius and time small step, as were used in Table I, the estimated diffusion coefficient will have the same value as for the novel method described in the previous section. In the limit

of small time steps, we see that the methods should give essentially the same result by taking the logarithm of Eq. (2.10) resulting in $\langle x^2 \rangle \approx r^2\theta^2 \approx 4Dt$ which corresponds to a random walk in $\mathbb{R}^2$. We chose, somewhat arbitrarily, $\varphi = \pi/4$ when comparing our new algorithm with particles moving in the tangent plane, for the different time steps and radii below.

Using either of the above algorithms in a molecular simulation, one would, however, like to increase the time step to more efficiently sample the configuration space. To that end, we investigated the stability of the three methods for generating a Brownian motion on S^2 by changing the time step for different radii of the sphere. With particular applications in mind, we tested the algorithms for two different radii corresponding to a surface fraction, ϕ, of (in the present case non-interacting) particles corresponding to $\phi = 0.01$ and $\phi = 0.10$. Since the two approximate methods using Gaussian random numbers gave very similar results, we only present the data from the non-scaled motion in the tangent plane. In Fig. 1 the diffusion coefficient is plotted versus different time steps for a Brownian motion on a sphere with a large radius corresponding to $\phi = 0.01$. The results from both algorithms virtually coincide and give a correct result, until a timestep of $\tau \cdot 10^5 \approx 20$ where the algorithm using a motion in the tangent plane for the diffusion becomes unstable.

Decreasing the radius until $\phi = 0.10$ in Fig. 2, one can observe that also in this case, the two algorithms coincide until about $\tau \cdot 10^5 \approx 20$, after which the tangent plane method begins to deviate, although the deviation is initially more gradual compared to Fig. 1. After $\tau \cdot 10^5 \approx 35$, however, the deviation for the instability of the tangent plane method rapidly increases.

We have shown above that the novel method presented in this paper for generating a Brownian motion on S^2 is much more stable and give the correct diffusion behaviour for larger values of τ (i.e. larger time steps and smaller radii) compared to more approximate methods often used in simulations of diffusion processes on spherical surfaces. We will use this method in future work on confined polymer and colloid systems.

A Appendix

In this appendix we show that the method for constructing a Brownian motion on S^2 from a random walk on S^3 which is described in Section 2 really gives a Brownian motion in the limit $\Delta t \to 0$. Since there is nothing special about dimensions 2 and 3 we derive this result for general dimensions.

Let S^n be the unit sphere in $\mathbb{R}^{n+1}$ and let S^{n-1} be its equator: in standard coordinates $x = (x^1, \ldots, x^{n+1})$ on $\mathbb{R}^{n+1}$ we have

$$S^n = \left\{ x \colon \sum_{j=1}^{n+1} x_j^2 = 1 \right\} \quad \text{and} \quad S^{n-1} = \left\{ x \colon \sum_{j=1}^{n+1} x_j^2 = 1,\ x^{n+1} = 0 \right\}.$$

Consider the embedding $F \colon S^{n-1} \times (-\frac{\pi}{2}, \frac{\pi}{2}) \to S^n$:

$$F(\eta, \theta) = (\cos\theta \cdot \eta, \sin\theta).$$

Note that $F(\{\eta\} \times (-\frac{\pi}{2}, \frac{\pi}{2}))$ is a geodesic through η perpendicular to S^{n-1}. (An embedding with this property is often called "Fermi coordinates based on S^{n-1}"). We use the map F to define a projection

$$\pi \colon S^n - \{(0, \ldots, 0, -1), (0, \ldots, 0, 1)\} \ \to \ S^{n-1},$$

where $\pi(\xi) = \eta$ where $F(\eta, \theta) = \xi$ for some $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$.

In order to give a physical motivation for our method of construction of Brownian motion we consider heat propagation on S^n. More precisely, consider a band $B_\epsilon = F(S^{n-1} \times (-\epsilon, \epsilon)) \subset S^n$ of width 2ϵ around S^{n-1} and assume that B_ϵ is completely insulated. Then it is reasonable to expect that heat propagation in B_ϵ converges to heat propagation in S^{n-1} as $\epsilon \to 0$. From the point of view of statistical mechanics, B_ϵ being insulated means that its boundary $\partial B_\epsilon = F(S^{n-1} \times \{-\epsilon\}) \cup F(S^{n-1} \times \{\epsilon\})$ reflects particles.

Translating the physical motivation, we define a stochastic process on S^{n-1} as follows. For finite $\Delta t > 0$ and any $\eta \in S^{n-1}$ take

$$Q_{\Delta t}(\eta) = \pi(W_{\Delta t}(\eta, 0)),$$

where $W_t(\xi)$ is the time t Brownian motion on S^n with initial value $\xi \in S^n$. Since the probability that $W_t(\xi)$ hits $(0, \ldots, 0, \pm 1)$ equals zero this is well defined. Next define

$$Q_t(\eta) = \lim_{N \to \infty} Q_{\frac{t}{N}} \circ \overset{N}{\cdots} \circ Q_{\frac{t}{N}}(\eta). \tag{A.1}$$

Note that for $n = 3$, $Q_{\Delta t}(\eta)$ is the result of applying the algorithm described in Section 2 to $\eta \in S^2$. Thus the fact that our algorithm describes Brownian motion on S^2 in the limit $\Delta t \to 0$ is a consequence of the following lemma.

Lemma A.1. *The limit in* (A.1) *exists and agrees with Brownian motion on* S^{n-1}.

Proof. To see this we use the local coordinate characterization of Brownian motion: a stochastic process which looks like Brownian motion in local coordinates is Brownian motion, see e.g. ref. 29, Proposition 3.2.1 and

Example 3.3.5. We thus consider Brownian motion on S^n in local coordinates. Write $x = (x^1, \ldots, x^{n-1}, x^n) = (\bar{x}, x^n)$ around a point in S^{n-1} where $\bar{x} = (x^1, \ldots, x^{n-1})$ are coordinates on S^{n-1} and x^n is the coordinate of geodesics perpendicular to S^{n-1}, i.e. x^n corresponds to θ in the Fermi coordinate map F above. Then, locally, the Brownian motion on S^n is the stochastic process $X_t = (X_t^1, \ldots, X_t^n)$ given by

$$dX_t^i = \sigma^{ij}(X_t) dB_t^j + b^i(X_t),$$

where $(B_t^1, \ldots, B_t^n)$ is Brownian motion on $\mathbb{R}^n$, where

$$\sigma^{ij} = \sqrt{g^{ij}},$$

and where

$$b^i = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^j} \left(g^{ij} \sqrt{g} \right).$$

Here $g_{ij}(x) dx^i \otimes dx^j$ is the metric tensor of S^n, g^{ij} is the inverse of g_{ij}, and $g = \det(g_{ij})$. (In the equations above and throughout this proof we use the summation convention: sum over repeated indices.)

In the Fermi coordinates $x = (\bar{x}, x^n)$ under consideration, we have

$$g_{ij}(\bar{x}, x^n) = \begin{cases} (\cos x^n)^2 \, h_{ij}(\bar{x}) & \text{if } i, j < n, \\ 0 & \text{if } i = n, j \neq n \text{ or } i \neq n, j = n, \\ 1 & \text{if } i = j = n. \end{cases}$$

where h^{ij} is the metric on S^{n-1}. Consequently,

$$\sigma^{ij}(\bar{x}, x^n) = \begin{cases} (\cos x^n)^{-1} \, h^{ij}(\bar{x}) & \text{if } i, j < n, \\ 0 & \text{if } i = n, j \neq n \text{ or } i \neq n, j = n, \\ 1 & \text{if } i = j = n. \end{cases}$$

and

$$g = (\cos x^n)^{2(n-1)}\, h.$$

In particular,

$$\sigma^{ij}(\bar{x}, 0) = \bar{\sigma}^{ij}(\bar{x}), \quad 1 \leq i, j \leq n-1,$$

where $\bar{\sigma}^{ij} = \sqrt{h^{ij}}$ and

$$b^i(\bar{x}, 0) = \bar{b}^i(\bar{x}), \quad 1 \leq i, j \leq n-1$$

where

$$\bar{b}^i = \frac{1}{\sqrt{h}} \frac{\partial}{\partial x^j} \left(h^{ij} \sqrt{h} \right).$$

In our Fermi coordinate system we have $\pi(\bar{x}, x^n) = \bar{x}$ and thus in local coordinates $\bar{x} = (x^1, \ldots, x^{n-1})$ on S^{n-1} we find that $Q_t = \pi(X_t^1, \ldots, X_t^{n-1}, 0) = \bar{X}_t$ satisfies

$$d\bar{X}_t^i = \bar{\sigma}^{ij}(\bar{X}_t) dB_t^j + \bar{b}^i(\bar{X}_t)$$

and thus gives Brownian motion on S^{n-1} as claimed. □

Acknowledgements

TC acknowledges NGSSC for financial support. CE acknowledges Uppsala university computing centre (UPPMAX) for computer time. TE was sponsored by the Göran Gustafson Foundation for Research in Natural Sciences and Medicine

References

1. G.H. Weiss, *Aspects and Applications of the Random Walk* (North-Holland, Amsterdam, 1994).

2. N. Wax (Ed.), *Noise and Stochastic processes* (Dover publications, New York, 1954).

3. J. Faraudo, J.Chem. Phys. **116**, 5831 (2002).

4. M.M.G. Krishna, J. Samuel, and S. Sinha, J. Phys. A: Math. Gen. **33**, 5965 (2000).

5. I.F. Sbalzarini, A. Hayer, A. Helenius, and P. Koumoutsakos, Biophys. J. **90**, 878 (2006).

6. I.L. Novak, F. Gao, Y.-S. Choi, D. Resasco, J.C. Schoff, and B.M. Slepchenko, J. Comp. Phys. **226**, 1271 (2007).

7. G. Li, L.-K. Tam, and J.X. Tang, PNAS **105**, 18355 (2008).

8. D.R. Brillinger, J. Theor. Prob. **10**, 429 (1997)

9. W.h. Furry, Phys. Rev. **107**, 7 (1957).

10. L.D. Favro, Phys. Rev. **119**, 53 (1960).

11. M. Antoine, A. Comtet, J. Desbois, and S. Ouvry, J. Phys. A: Math. Gen. **24**, 2581 (1991).

12. T. Bülov, IEEE Trans. Pattern Analys. Machine Intellig. **26**, 1650 (2004).

13. E. Breitenberger, Biometrika **50**, 81 (1963).

14. P. Hartman and G.S. Watson, Annal. Probability **2**, 593 (1974).

15. I. Snook, *The Langevin and Generalised Langevin Approach to the Dynamics of Atomic, Polymeric and Colloidal Systems* (Elsevier, Amsterdam, 2007)

16. M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids* (OUP, Oxford, 1987).

17. C. Elvingson, J. Comp. Chem. **12**, 71 (1991).

18. J. Nissfolk, T. Ekholm, and C. Elvingson, J. Chem. Phys. **119**, 6423 (2003).

19. J.-M. Caillol, J. Phys. A: Math. Gen. **37**, 3077 (2004).

20. M. Taylor, *Partial Differential Equations II* (Springer-Verlag, New York, 1996).

21. D. Frenkel and B. Smit, *Understanding Molecular Simulation* 2:nd Ed. (Academic Press, San Diego, 2002).

22. V. Tulovsky and L. Papiez, Appl. Math. Lett. **14**, 881 (2001).

23. G. Szegö, *Orthogonal polynomials* (Americal methematical Society, New York, 1939).

24. W. Magnus, F Oberhettinger, F. G. Tricomi, *Tables of Integral Transforms Vol I* (McGraw-Hill Book Company, Inc, 1954).

25. W. Magnus, F Oberhettinger, F. G. Tricomi, *Higher Transcendental Functions Vol II* (McGraw-Hill Book Company, Inc, 1953)

26. P.J. Råsmark, T. Ekholm, and C. Elvingson, J. Chem. Phys. **122**, 184110 (2005).

27. A. Delville, J.-M. Pellenq, and J.M. Caillol, J. Chem. Phys. **106**, 7275 (1997).

28. E. Allahyarov, I. D'Amico, and H. Löwen, Phys. Rev. E **60**, 3199 (1999).

29. E. P. Hsu, *Stochastic Analysis on Manifolds* (American Mathematical Society, Providence R.I., 2002, Graduate studies in mathematics v. 38).

TABLE I. The diffusion coefficient, D/D_0, for freely diffusing particles on S^2. Algorithm 1 is the method presented in this article, with the S^2-surface being located at different angles relative to the x_4-axis. In algorithm 2 each particle will, for every time step, move in the plane tangent to the point where it is located after which is projected onto the spherical surface along the line from the centre of the particle to the centre of S^2. The size of the random step is obtained by drawing Gaussian random numbers. In algorithm 3, the size of the random displacement in the tangent plane is scaled so that after projection, the distance along the S^2 surface will be the same as the distance in the tangent plane in algorithm 2. The time step corresponds to $\tau \cdot 10^5 = 6.7$ and the surface fraction $\phi = 0.01$.

φ	Algorithm 1	Algorithm 2	Algorithm 3
15	1.006	0.997	0.997
30	1.007	0.997	0.997
45	1.002	0.997	0.997
60	1.001	0.997	0.997
85	1.007	0.997	0.997

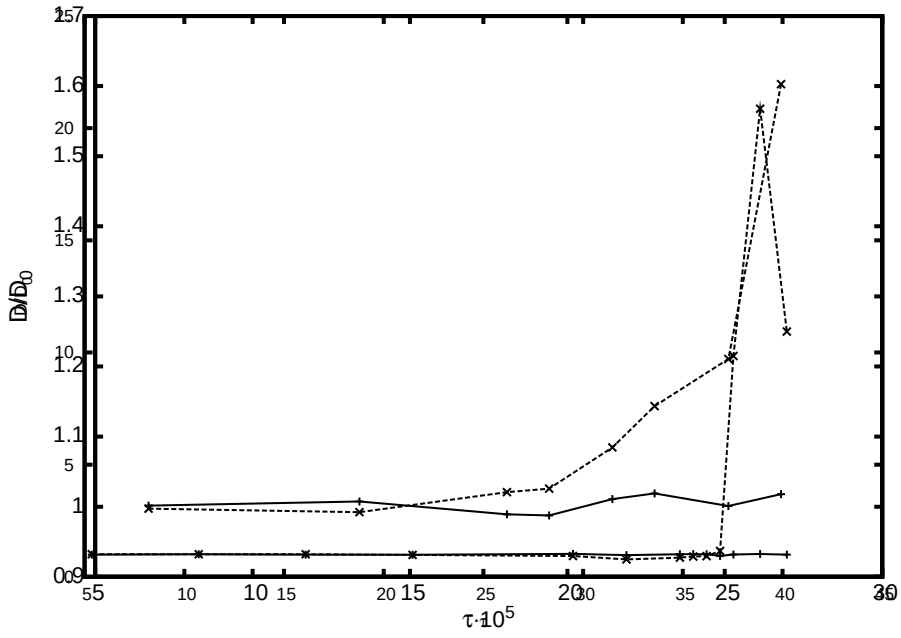

Figure 1: The diffusion coefficient vs reduced time step in simulation for a sphere with radius corresponding to $\phi = 0.01$

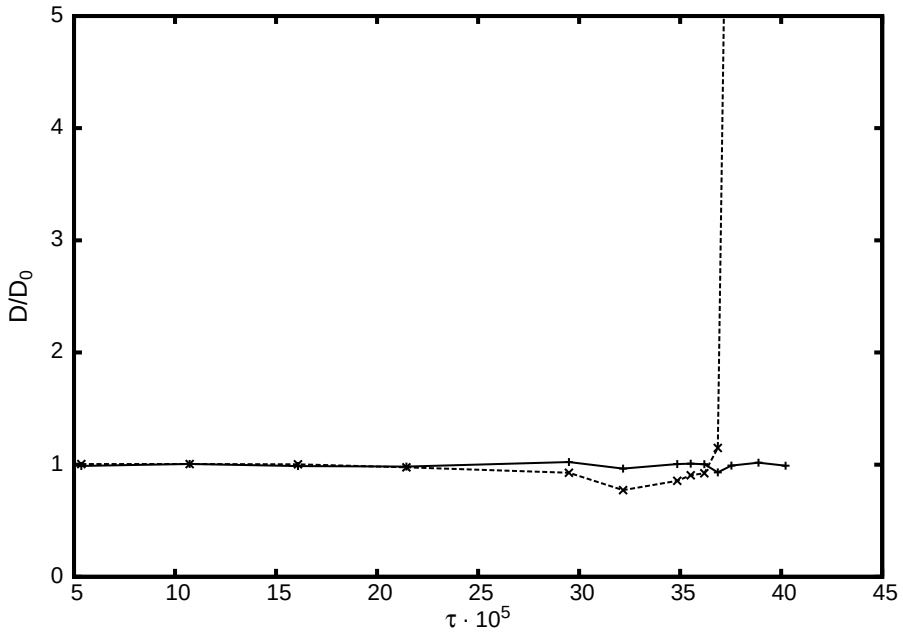

Figure 2: The diffusion coefficient vs reduced time step in the simulation for $\phi = 0.10$.