Applied Mathematics, 2020, 11, 35-41
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Fast Diffusion Monte Carlo Sampling via Conformal Map

Chi-Ok Hwang[1]*, Mi Do[2]

[1]Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
[2]University of Washington, Seattle, USA
Email: *chwang@gist.ac.kr

How to cite this paper: Hwang, C.-O. and Do, M. (2020) Fast Diffusion Monte Carlo Sampling via Conformal Map. *Applied Mathematics*, **11**, 35-41.
https://doi.org/10.4236/am.2020.111004

Received: December 10, 2019
Accepted: January 6, 2020
Published: January 9, 2020

Abstract

In diffusion Monte Carlo methods, depending on the geometry continuous diffusion can be simulated in many ways such as walk-on-spheres (WOS), walk-on-planes (WOP), walk-on-rectangles (WOR) and so on. The diffusion ways are conformally the same satisfying the Laplace equation with the given boundary geometry. In this paper, using the WOP and the conformal map, we sample the WOS diffusion and show that the indirect sampling is more efficient than the direct WOS sampling. This signifies that fast diffusion Monte Carlo sampling via conformal map can be possible.

Keywords

Monte Carlo, Sampling, Conformal Map

1. Introduction

Monte Carlo methods use simulations of random events using random numbers [1] [2]. In the processes of Monte Carlo methods, the sampling of random events is the most crucial part and it takes most of the simulation time. Therefore, if we can save time in the sampling, it will expedite the Monte Carlo simulation.

Based on the probabilistic potential theory [3] [4], elliptic partial differential operators can be interpreted as diffusion motion so that via diffusion simulation we can solve some elliptic partial differential equations such as Laplace, Poisson equations and so on. Also, there is one-to-one correspondence between electrostatic problems and diffusion motion expectation one because both problems satisfy the same partial differential equations. Using the correspondence, Green's function first-passage algorithms have been developed.

In diffusion Monte Carlo [5] [6] [7] [8], we need simulate continuous diffusion via walk-on-spheres (WOS) [5], walk-on-planes (WOP) [9], walk-on-rectangles

(WOR) [10] [11] and so on, depending on the diffusion geometry. WOS uses spheres for diffusion simulation. When diffusion starts from the center of the sphere, first-passage probability is uniform over the spherical surface so that we can sample the next diffusion point uniformly over the spherical surface. WOP uses an infinite plane. In WOP, a diffusion starts from a point at a distance from the plane and makes a fist-passage on the plane. WOP algorithm uses cubes instead of spheres. In WOP, a diffusion starts from the center of the cube and makes a first-passage on the surfaces of the cube. Among them, in continuous space diffusion WOS is the most used algorithm. We should note that all the algorithms are conformally equivalent and WOP sampling is the simplest one. The conformal equivalence opens a new way of sampling.

In this paper, using the WOP and the conformal map we sample the WOS diffusion and show that the indirect sampling is more efficient than the direct WOS sampling. This signifies that fast diffusion Monte Carlo sampling via conformal map can be possible.

2. Conformal Map between WOP and WOS

In this section, we show how to do the conformal map between WOP and WOS. In **Figure 1**, we show the conformal map between WOP and WOS. It shows the intersection view of the inversion of a sphere to a plane: O is the center of the inversion sphere (red solid line with radius R_c) and through the inversion, charge q at the distance p from the center of the sphere (blue solid line with radius a) becomes $qR_c/(a+p)$ at $R_c^2/(a+p)-2a$ from the plane and $\sigma(r,\theta,\phi)$ changes to $\sigma(\rho)$, that is, $\left[R_c/\sqrt{R_c^2+\rho^2}\right]^3\sigma(r,\theta,\phi)$. The inverting sphere with radius a corresponds to the WOS and the inverted plane to the WOP.

Using the method of image charge [12], induced charge distribution on the spherical grounded conducting surface by a charge q inside of the sphere is given by (see **Figure 1**):

$$\sigma(a,\theta,\phi)=\frac{q\left(a^2-p^2\right)}{4\pi a\left(a^2-2ap\cos(\theta)+p^2\right)^{3/2}}. \tag{1}$$

Also, induced charge distribution at the distance $\rho=\sqrt{x^2+y^2}$ on the *xy*-plane, the infinite grounded conducting plate, by a charge q located at $(0,0,d)$ is obtained as:

$$\sigma(\rho)=\frac{1}{2\pi}\frac{1}{\sqrt{d^2+\rho^2}}. \tag{2}$$

In WOP algorithm, when the diffusion starts at the distance d from the WOP plane, the ρ sampling, the radial distance sampling from the center of WOP place, is simply,

$$\frac{\rho}{d}=\sqrt{\frac{1-U^2}{U^2}},\quad U\in[0,1]. \tag{3}$$

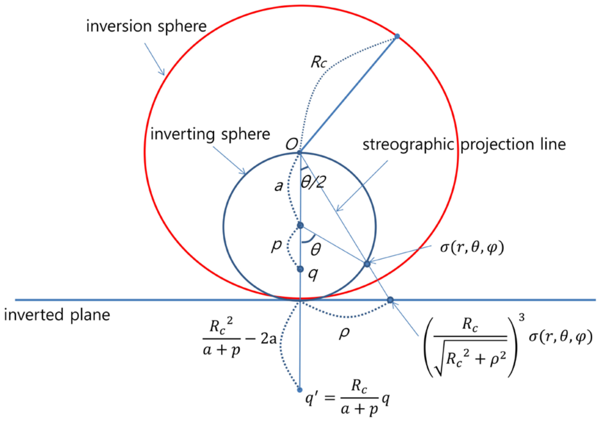
inversion sphere
R_c
inverting sphere
O
streographic projection line
a
$\theta/2$
p
θ
q
$\sigma(r,\theta,\varphi)$
inverted plane
$\frac{R_c{}^2}{a+p} - 2a$
ρ
$\left(\frac{R_c}{\sqrt{R_c{}^2+\rho^2}}\right)^3 \sigma(r,\theta,\varphi)$
$q' = \frac{R_c}{a+p} q$

Figure 1. Intersection view of the inversion of a sphere to a plane: O is the center of the inversion sphere (red solid line with radius R_c) and through the inversion, charge q at the distance p from the center of the sphere (blue solid line with radius a) becomes $qR_c/(a+p)$ at $R_c^2/(a+p)-2a$ from the plane and $\sigma(r,\theta,\phi)$ changes to $\sigma(\rho)$, that is, $\left[R_c/\sqrt{R_c^2+\rho^2}\right]^3 \sigma(r,\theta,\phi)$.

The induced charge distribution over the spherical surface and the one over the plate can be related by conformal map. **Figure 1** shows the conformal map. We should note that sphere inversion and streographic projection are special cases of the conformal map.

According to inversion transformation [13] (see **Figure 1**), one of the conformal maps, the charge q inside of the sphere with radius a at a distance p from the center of the sphere is inverted to,

$$q' = \frac{R_c}{a+p} q, \tag{4}$$

at the distance,

$$d = \frac{R_c^2}{a+p} - 2a, \tag{5}$$

from the inverted plane. Also, from the other point of view of the charge distribution, the charge distributions, $\sigma(\boldsymbol{r})$ (before the inversion) and $\tilde{\sigma}(\tilde{\vec{r}})$ (after the inversion) are related as follows:

$$\tilde{\sigma}(\tilde{\boldsymbol{r}}) = \left(\frac{R}{\tilde{r}}\right)^3 \sigma(\boldsymbol{r}), \tag{6}$$

where R is radius of the inversion sphere, and $\tilde{r} = |\tilde{\boldsymbol{r}}|$.

After sampling over the infinite plane (WOP) and converting over the spherical surface (WOS), we need to rotate the sampling point in regard to the point of the previous diffusion point. Here, we need the following Rodrigues' rotation formula.

If $\boldsymbol{v}$ is a vector in R^3 and $\boldsymbol{k}$ is a unit vector describing an axis of rotation about which $\boldsymbol{v}$ rotates by an angle θ according to the right hand rule, the Rodrigues' rotation formula gives the rotated vector $\boldsymbol{v}_{\text{rot}}$,

$$\boldsymbol{v}_{\mathrm{rot}} = \boldsymbol{v}\cos\theta + (\boldsymbol{k}\times\boldsymbol{v})\sin\theta + \boldsymbol{k}(\boldsymbol{k}\cdot\boldsymbol{v})(1-\cos\theta). \tag{7}$$

In this way, we can sample a diffusion point over the spherical surface (WOS sample).

3. Results

In this section, we check that the sampling via conformal map works well and compare two samplings via operation counting and running time. Counting operations are good for computing performance. In this performance comparison, counting operations consist of numbers of addition/substraction, multiplication, division and special function.

In Figure 2, we can see that the sampling via conformal map works well. The solid line is the analytic cumulative distribution over the spherical surface and the circles are the simulation sampling points combining the sampling on the infinite plane and the conformal map. The two sampling agrees well with each other.

The first table (Table 1) shows the comparison of the counting operations between the direct sampling and the indirect sampling via the conformal map. We can reduce some computing operations. In real computation, Table 2 shows the overall efficiency of the indirect sampling. Usually, we run a Monte Carlo program over a week for a real data. In this regard, roughly speaking we can save a couple of days.

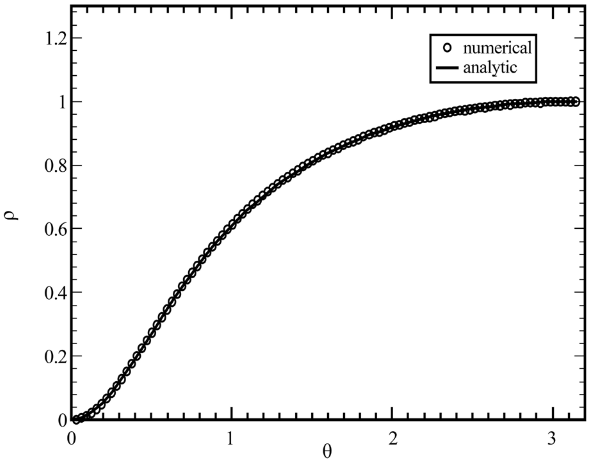

Figure 2. This graph shows that the sampling via conformal map works well: circles are numerical results and solid line is the analytic result of the cumulative sampling probability. The x-axis is the sampling angle θ in WOS sampling and y-axis the cumulative probability density.

Table 1. Operation counting for the sampling via conformal map and the direct sampling.

Algorithm	Addition/Substraction	Multiplication	Division	Special function
Sampling via conformal map	7	9	5	13
Direct sampling	20	32	14	17

Table 2. CPU time for the sampling via conformal map and the direct sampling. The number of random walks per run was 10^6.

Algorithm	CPU time per run (s)
Sampling via conformal map	29.77
Direct sampling	59.51

4. Conclusions

In Monte Carlo methods, sampling of random events using random numbers takes significant computing time. So, if we can sample fast, that can improve the performance of the Monte Carlo simulation. In diffusion Monte Carlo algorithms, fast diffusion simulation is essential. For fast diffusion simulation, we use many diffusion algorithms depending on the surrounding geometry like walk-on-spheres (WOS), walk-on-planes (WOP), walk-on-rectangles (WOR) and so on. Among them, WOP is the fastest sampling and for diffusion in free space WOS is the most-used algorithm.

In this paper, using the WOP and the conformal map we sample the WOS diffusion and show that the indirect sampling is more efficient than the direct WOS sampling. This signifies that fast diffusion Monte Carlo sampling via conformal map can be possible.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2017R1E1A1A03070543) funded by the Korea government (Ministry of Science and Information & Communication Technology (ICT)). In addition, this work was supported by the GIST Research Institute (GRI) grant funded by GIST in 2019.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Sobol, I.M. (1994) A Primer for the Monte Carlo Method. CRC Press, Washington, D.C., USA.

[2] Lemieux, C. (2009) Monte Carlo and Quasi-Monte Carlo Sampling. Springer-Verlag, New York, NY, USA.

[3] Freidlin, M. (1985) Functional Integration and Partial Differential Equations. Princeton University Press, Princeton, NJ, USA.

[4] Chung, K.L. and Zhao, Z. (1995) From Brownian Motion to Schrödinger's. Springer-Verlag, Berlin.

[5] Muller, M.E. (1956) Some Continuous Monte Carlo Methods for the Dirichlet Problem. *The Annals of Mathematical Statistics*, **27**, 569-589. https://doi.org/10.1214/aoms/1177728169

[6] Torquato, S. and Kim, I.C. (1989) Efficient Simulation Technique to Compute Effective Properties of Heterogeneous Media. *Applied Physics Letters*, **55**, 1847. https://doi.org/10.1063/1.102184

[7] Given, J.A., Hwang, C.-O. and Mascagni, M. (2002) First- and Last-Passage Monte Carlo Algorithms for the Charge Density Distribution on a Conducting Surface. *Physical Review E*, **66**, Article ID: 056704. https://doi.org/10.1103/PhysRevE.66.056704

[8] Hwang, C.-O., Hong, S. and Kim, J. (2015) *Journal of Computational Physics*, **303**, 331-335. https://doi.org/10.1016/j.jcp.2015.10.002

[9] Mansfield, M.L., Douglas, J.F. and Garboczi, E.J. (2001) Intrinsic Viscosity and the Electrical Polarizability of Arbitrarily Shaped Objects. *Physical Review E*, **64**, Article ID: 061401. https://doi.org/10.1103/PhysRevE.64.061401

[10] Booth, T.E. (1982) Regional Monte Carlo Solution of Elliptic Partial Differential Equations. *Journal of Computational Physics*, **47**, 281-290. https://doi.org/10.1016/0021-9991(82)90079-1

[11] Torquato, S., Kim, I.-C. and Cule, D. (1999) Effective Conductivity, Dielectric Constant, and Diffusion Coefficient of Digitized Composite Media via First-Passage-Time Equations. *Journal of Applied Physics*, **85**, 1560. https://doi.org/10.1063/1.369287

[12] Smythe, W.R. (1939) Static and Dynamic Electricity. McGraw-Hill, New York and London.

[13] Amaral, R.L.P.G., Ventura, O.S. and Lemos, N.A. (2017) Kelvin Transformation and Inverse Multipoles in Electrostatics. *European Journal of Physics*, **38**, Article ID: 0252061. https://doi.org/10.1088/1361-6404/aa55a0

Appendix

The python function below can be used to generate direct sampling model.

```
# calculate point's coordinates from charge using random walk
# return point's coordinates
def bsurf(x, y, z, r) :
    temp = a/r
    p2 = np.random.rand(1)
    costh = -(1.0-temp)**2 + 2*(1.0-temp)*(1+(temp)**2.0)*p2
        + 2.0*temp*(1+(temp)**2)*(p2)**2
    costh = costh/((1.0-temp+2.0*temp*p2)**2)
    sinth = np.sqrt(np.abs(1.0-(costh**2)))

    yyy = np.random.rand(1)
    phi = 2*np.pi*yyy
    cosph = np.cos(phi)
    sinph = np.sin(phi)

    xold = x
    yold = y
    zold = z
    p = np.sqrt(xold**2 + yold**2)
    x = a*(sinth*cosph*xold*zold/(p*r) - sinth*sinph*yold/p + costh*xold/r)
    y = a*(sinth*cosph*yold*zold/(p*r) + sinth*sinph*xold/p + costh*yold/r)
    z = a*(-sinth*cosph*p/r + costh*zold/r)

    return (x, y, z)

# sampling pts points
# return thetas as cumulative density from -pi/2 to pi/2
pts = 1000000000
for step in range(pts):
    # create a vector v_x, v_y, v_z --> theta = (Ox, v) from -pi/2 to pi/2
    (v_x, v_y, v_z) = bsurf(x, y, z, r)
    sintheta = (np.sqrt((v_x-(x/2))**2+(v_y-(y/2))**2+(v_z-(z/2))**2))/(2*a)
    theta = np.arcsin(sintheta[0])
    thetas.append(theta*2)
```

The python function below can be used to generate sampling via conformal map.

```
# sampling pts points over the infinite plane
# converting those points over the spherical surface
# then rotating using Rodrigues's rotation formula
# return thetas2 as cumulative density from -pi/2 to pi/2

pts1 = 1000000
thetas2 = []
for step in range(pts1):
    x = np.random.rand(1)
    rho_d = np.sqrt((1-x**2)/(x**2))
    rho = rho_d * d
    (xA, yA, zA) = (rho,0,-4)
    r = np.sqrt(rho**2 + (-4)**2)
    sinthe = rho/r
    v1 = Rc**2/r
    xA1 = v1*sinthe
    zA1 = v1*Rc/r

# Rodrigues's rotation for vector v = point(rho_d, Rc)
theta = np.random.uniform(np.pi*2)
i = np.cos(theta)
j = np.sin(theta)
vX = xA1*i
vY = xA1*j
vZ = zA1*i + 0 + zA1*(1-i)

costheta2 = np.sqrt(vX**2 + vY**2 + vZ**2)/Rc
thetas2.append(2*np.arccos(costheta2))
```