COMMUN. MATH. SCI.
Vol. 15, No. 1, pp. 237–259

COMPUTATION OF THE LOCAL TIME OF REFLECTING BROWNIAN MOTION AND THE PROBABILISTIC REPRESENTATION OF THE NEUMANN PROBLEM*

YIJING ZHOU†, WEI CAI‡, AND ELTON HSU§

Abstract. In this paper, we propose numerical methods for computing the boundary local time of reflecting Brownian motion (RBM) for a bounded domain in $\mathbb{R}^3$ and the probabilistic solution of the Laplace equation with the Neumann boundary condition. Approximations of RBM based on walk-on-spheres (WOS) and random walk on lattices are discussed and tested for sampling RBM paths and their applicability in finding accurate approximation of the local time and discretization of the probabilistic representation of the Neumann problems using the computed local time. Numerical tests for several domains (a cube, a sphere, an ellipsoid, and a non-convex non-smooth domain made of multiple spheres) have shown the convergence of the numerical methods as the time length of RBM paths and number of paths sampled increase.

Keywords. Reflecting Brownian Motion, Brownian motion, boundary local time, Skorohod problem, WOS, random walk, Laplace equation.

AMS subject classifications. 65C05, 65N99, 78M25, 92C45.

1. Introduction

Traditionally numerical solutions of boundary value problems for partial differential equations (PDEs) are obtained by using finite difference, finite element, or boundary element methods with both space and/or time discretizations. This usually requires spatial mesh fine enough to ensure accuracy, which results in considerable storage space requirement and computation time. Moreover, the solution process is global, namely, the solutions of the PDEs have to be found together at all mesh points. In many scientific and engineering applications, local solutions are sometimes all we need, such as the local electrostatic potential on a molecular surface where molecular binding activities are most likely to occur or the stress field at specific locations where the materials are susceptive to failure. Therefore, it is of practical importance to have a numerical approach which can give a local solution of the PDEs at some locality of our choice. In the case of elliptic PDEs, this kind of local numerical method can be constructed using the well-known probabilistic representation and the associated Feynman–Kac formula [12, 13], which relate Itô diffusion paths to the solution of an elliptic PDE. By sampling diffusion paths, the evaluation of the solution at any point in the domain can be done through an averaging process of the boundary (Dirichlet or Neumann) data under some given measure on the boundary. Moreover, this method avoids expensive mesh generations required by mesh-based deterministic methods [27].

Our previous work [28], using the Feynman–Kac formula for the Laplace equation with Dirichlet data, has produced a local method for computing the DtN (Dirichlet-to-

*Received: July 29, 2015; accepted (in revised form): May 23, 2016. Communicated by Qiang Du. National Science Foundation (DMS-1315128), the National Natural Science Foundation of China (No. 91330110) and a Simons Foundation Collaboration Grant for Mathematicians.

†Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223-0001, (yzhou16@uncc.edu).

‡Corresponding author, INS, Shanghai Jiao Tong University, Shanghai 200240, P.R. China, and Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223-0001, (wcai@uncc.edu).

§Department of Mathematics, Northwestern University, Evanston, IL 60208, (ehsu@math.northwestern.edu).

Neumann) mapping for the Laplace operator. In this paper, we will focus on solving the following Neumann boundary value problem of the elliptic PDE using a probabilistic approach:

$$\begin{cases} \Delta u = f \text{ on } D \\ \dfrac{\partial u}{\partial n} = \phi \text{ on } \partial D, \end{cases} \tag{1.1}$$

where D is a bounded domain in $\mathbb{R}^3$, Δ is the Laplace operator, f is a measurable function, and ϕ is a bounded measurable function on the boundary ∂D satisfying $\int_{\partial D} \phi d\sigma = \int_D f dx$. Equation (1.1) becomes the Laplace equation when $f = 0$, which is the subject of our work.

The PDE (1.1) originates from either the Poisson equation for electrostatic potentials [11], an implicit time discretization of the heat equation or the momentum equation of the Navier–Stokes equation with an additional lower-order term in the latter cases. Historically, Brownian motion (BM) has been used in solving PDEs due to its effectiveness and easy implementation regardless of dimensions [18]. The well-known probabilistic representation can solve the elliptic equation with the Dirichlet boundary condition by using the first exit time τ_D of BM when $f = 0$, i.e.,

$$u(x) = E^x(\phi(X_{\tau_D})). \tag{1.2}$$

In the above formula, only the values at the hitting positions on the boundary are used in the computation of the mathematical expectation (average) to obtain $u(x)$. Using the construct of killed Brownian motion [28] in conjunction with Monte Carlo methods, we can easily obtain an estimate of $u(x)$.

However, for the Neumann problem to be studied here, in contrast to the usual Brownian motion used in Equation (1.2), reflecting Brownian motion (RBM) [20, 26] will be needed to produce a similar probabilistic solution. The theory has been developed in [1, 5, 15] by employing the concept of the boundary local time whose one dimensional predecessor was introduced by Lévy in [19]. In [15], the boundary local time of a one dimensional BM was extended to high dimensions and an explicit form, shown in Equation (2.4), was obtained for domains with smooth boundaries. For higher dimensions, similar results had also been found by Brosamler [5]. It should be noted that the boundary local time is related to the Skorohod problem [7] and plays a significant role in the theoretical development of the probabilistic approach to the Neumann problem.

One-dimensional local time of Brownian motion has been studied by many authors [7, 16, 17, 19, 23]. Numerical methods using Euler discretizations of the underlying diffusion process have also been used to compute the reflecting Brownian motions and the solutions to parabolic equations with mixed boundary conditions [4, 9]. Meanwhile, Morillon [22] proposed random walk on a grid to treat the reflecting paths and considered the related Feynman–Kac formula for the Poisson problem with various boundary conditions. Recently, in [21] a walk-on-spheres (WOS) method, as well as the Euler discretization method with a localization kernel approach for computing the local time of reflecting Brownian motions [16], was used to sample the Brownian motions inside the domain and a finite difference discretization of the Neumann boundary condition to obtain approximation of the solution near the boundary with the help of randomization. In this paper, we will propose numerical methods, using the WOS technique, explicitly for computing the local time of reflecting Brownian motions for a bounded domain in $\mathbb{R}^3$

based on a rigorous probabilistic theory and apply the resulting methods to implement computationally the probabilistic representation for the Neumann problem.

This paper is organized as follows. In Section 2, we introduce the RBM via the Skorohod problem and the concept of local time of the RBM and the explicit probabilistic solution to the Neumann problem. In Section 3, the WOS method is reviewed and discussed for its application to the RBM. In Section 4, a numerical method, the WOS combined with a Monte Carlo method, is proposed for an approximation to the Neumann problem. Numerical results for a cube, a sphere, an ellipsoid, and a non-smooth convex domains will be given in Section 5. Section 5 also includes discussions on convergence rates, various numerical issues of the algorithms, and the comparison with other Monte Carlo and deterministic methods. Finally, we draw conclusions from our Monte Carlo simulations and discuss possible further work in Section 6.

2. Boundary local time of RBM and Neumann problem

Before discussing a probabilistic solution for the Neumann problem for elliptic PDEs, we introduce the boundary local time for reflecting Brownian motion through the Skorohod problem.

DEFINITION 2.1 (Skorohod problem). *Assume D is a bounded domain in R^d with a C^2 boundary. Let $f(t)$ be a (continuous) path in R^d with $f(0)\in\bar{D}$. A pair $(\xi(t),L(t))$ is a solution to the Skorohod problem $S(f;D)$ if the following conditions are satisfied:*

1. *ξ is a path in $\bar{D}$;*
2. *$L(t)$ is a nondecreasing function which increases only when $\xi\in\partial D$, namely,*

$$L(t)=\int_0^t I_{\partial D}(\xi(s))L(ds); \tag{2.1}$$

3. *the Skorohod equation holds*

$$S(f;D): \qquad \xi(t)=f(t)-\frac{1}{2}\int_0^t n(\xi(s))L(ds), \tag{2.2}$$

where $n(x)$ stands for the outward unit normal vector at $x\in\partial D$.

If $f(t)$ is replaced by a standard Brownian motion B_t, the corresponding ξ_t will be a standard reflecting Brownian motion (RBM) X_t. Just as the name suggests, a RBM behaves like a BM as long as its path remains inside the domain D, but it will be reflected back inwardly along the normal direction of the boundary when the path attempts to pass through the boundary. The fact that X_t is a diffusion process can be proven by using a martingale formulation and showing that X_t is the solution to the corresponding martingale problem with the Neumann boundary condition [15]. The result gives an intuitive and direct way to construct RBM from BM, which will be discussed in detail in Section 4.

The boundary local time $L(t)$ for a RBM measures the amount of time the RBM spends near the boundary and at the same time the frequency that the RBM hits the boundary. We have the following properties of $L(t)$:

(a) it is the unique continuous nondecreasing process that appears in the Skorohod equation (2.2) [15, 20, 26];

(b) it measures the amount of time the standard reflecting Brownian motion X_t spending in a vanishing neighborhood of the boundary during the time interval $[0,t]$. If D has a C^3 boundary, we have

$$L(t)\equiv\lim_{\epsilon\to 0}\frac{\int_0^t I_{D_\epsilon}(X_s)ds}{\epsilon}, \tag{2.3}$$

where D_ϵ is a strip region of width ϵ containing ∂D and $D_\epsilon \subset D$. This limit exists both in L^2 and P^x-a.s. for any $x \in \overline{D}$.

Alternatively, we have the following explicit formula for $L(t)$ derived in [15],

$$L(t) = \sqrt{\frac{\pi}{2}} \int_0^t I_{\partial D}(X_s) \sqrt{ds}, \tag{2.4}$$

where the right-hand side of Equation (2.4) is understood as the limit of

$$\sqrt{\frac{\pi}{2}} \sum_{i=1}^{n-1} \max_{s \in \Delta_i} I_{\partial D}(X_s) \sqrt{|\Delta_i|}, \quad \max_i |\Delta_i| \to 0, \tag{2.5}$$

where $\Delta = \{\Delta_i\}$ is a partition of the interval $[0,t]$ and each Δ_i is an element in Δ. We will discuss the implementation of both Equations (2.3) and (2.4) in Section 4.

We are now ready to consider the following elliptic PDE in $\mathbb{R}^3$ with a Neumann boundary condition,

$$\begin{cases} \left(\dfrac{\Delta}{2} + q\right) u = 0, \text{ on } D \\ \dfrac{\partial u}{\partial n} = \phi, \text{ on } \partial D \end{cases}. \tag{2.6}$$

When the bottom of the spectrum of the operator $\Delta/2 + q$ is negative a probablistic solution of Equation (2.6) is given by

$$u(x) = \frac{1}{2} E^x \left[\int_0^\infty e_q(t) \phi(X_t) L(dt) \right], \tag{2.7}$$

where X_t is a RBM starting at x and $e_q(t)$ is the Feynman–Kac functional [15]

$$e_q(t) = \exp\left[\int_0^t q(X_s) ds \right].$$

From the definition of the local time in (2.3), we have the following approximation for small ϵ

$$L(t) \approx \frac{\int_0^t I_{D_\epsilon}(X_s) ds}{\epsilon}. \tag{2.8}$$

Plugging Equation (2.8) into Equation (2.7), we have

$$u(x) \approx \frac{1}{2\epsilon} E^x \left[\int_0^\infty e_q(t) \phi(X_t) \int_t^{t+dt} I_{D_\epsilon}(X_s) ds \right]. \tag{2.9}$$

The solution defined in (2.7) should be understood as a weak solution for the classical PDE (2.6). The identification of Equation (2.7) with the classical solution can be achieved by using the martingale formulation [15]. If the weak solution satisfies some smoothness condition [5, 15], it can be shown that it is also a classical solution to the Neumann problem. This formula is the basis for our numerical approximations to the Neumann problem (2.6). To compute the expectation in this formula, we rely on Monte Carlo random samplings to simulate Brownian paths and then take the average.

In the present work, as we only consider the Laplace equation ($q=0$), we have

$$u(x)\approx\frac{1}{2\epsilon}E^x\left[\int_0^\infty\phi(X_t)\int_t^{t+dt}I_{D_\epsilon}(X_s)ds\right], \tag{2.10}$$

and we will show how this formula is implemented with the Monte Carlo and WOS methods in Section 4.

Remark 2.1. Formula (2.7) is very similar to the probabilistic solution (1.2) for the Laplace equation with the Dirichlet boundary condition. In the Dirichlet case, killed Brownian paths were sampled by running random walks until they are absorbed on the boundary upon a first hit on the boundary and $u(x)$ is evaluated as an average of the boundary values at the first hitting positions on the boundary, namely, $u(x)=E^x[\phi(X_{\tau_D})]$ where ϕ is the Dirichlet boundary data. On the other hand, for the Neumann condition, while $u(x)$ is also given as a weighted average of the Neumann data at multiple hitting positions of RBM on the boundary, the weight is related to the boundary local time of RBM. This is a noteworthy point when we compare the probabilistic solutions of the two boundary value problems and try to understand the formula in Equation (2.7).

3. Method of walk on spheres (WOS)

Random walk on spheres (WOS) method was first proposed by Müller [24], which can solve the Dirichlet problem for the Laplace operator efficiently with the probabilistic representation of the solution (1.2). The expectation in Equation (1.2) is taken over all sample paths starting from x and τ_D is the first exit time for the domain D. For the Neumann boundary condition, similar formulas can be obtained [22]. However different measures on the boundary ∂D is used in the mathematical expectation.

In order to illustrate the WOS method for the Dirichlet problem, formula (1.2) can be rewritten in terms of a measure μ_D^x defined on the boundary ∂D,

$$u(x)=E^x(\phi(X_{\tau_D}))=\int_{\partial D}\phi(y)d\mu_D^x, \tag{3.1}$$

where μ_D^x is the so-called harmonic measure defined by

$$\mu_D^x(F)=P^x\{X_{\tau_D}\in F\},F\subset\partial D,x\in D. \tag{3.2}$$

It can be shown that the harmonic measure is related to the Green's function for the domain with a homogeneous boundary condition [8],

$$p(x,y)=-\frac{\partial g(x,y)}{\partial n_y}, \tag{3.3}$$

where

$$\begin{cases}-\Delta g(x,y)=\delta(x-y), & x\in D,\\ g(x,y)=0, & x\in\partial D\end{cases}. \tag{3.4}$$

If the starting point x of a Brownian motion is at the center of a ball, the probability of the BM exiting a portion of the boundary of the ball will be proportional to the portion's area. It is known that all sample paths of a Brownian motion starting in the domain intersects the boundary ∂D almost surely [24]. Therefore, sampling a Brownian

path by drawing balls within the domain, regardless of how the path navigates in the interior of the ball, can significantly reduce the path sampling time. To be more specific, given a starting point x inside the domain D, we simply draw a ball of the largest possible radius fully contained in D and then the next location of the Brownian path on the surface of the ball can be sampled, using a uniform distribution on the sphere, say at x_1. Treat x_1 as the new starting point, draw a second ball fully contained in D, make a jump from x_1 to x_2 on the surface of the second ball as before. Repeat this procedure until the path hits an absorption ϵ-shell of the domain [14]. When this happens, we assume that the path has hit the boundary ∂D (see Figure 3.1(b) for an illustration).

(a) WOS within the domain.

(b) WOS (with a maximal step size for each jump) within the domain.

FIG. 3.1. *Walk on Spheres method.*

Next we define an estimator of (3.1) by

$$u(x) \approx \frac{1}{N} \sum_{i=1}^{N} \phi(x_i), \tag{3.5}$$

where N is the number of Brownian paths sampled and x_i is the first hitting point of each path on the boundary. Using a jump size (radius of the ball) δ on each step for the WOS, we expect to take $O(1/\delta^2)$ (δ is chosen independently of ϵ) steps for a Brownian path to reach the boundary [2]. To speed up, maximum possible size for each step would allow faster first hitting on the boundary. Most of the numerical results in this paper will use the WOS approach as illustrated in Figure 3.1(b).

4. Numerical methods

4.1. Simulation of reflecting Brownian paths. A standard reflecting Brownian motion path can be constructed by reflecting a standard Brownian motion path back into the domain whenever it crosses the boundary. So in principle, the simulation of RBM is reduced to that of BM.

It is known that standard Brownian motion can also be constructed as the scaling limit of a random walk on a lattice so we can model BM by a random walk with proper scaling (see Appendix A for details). We find out that the WOS method is the better method to simulate BM for our purpose [25] (see Remark 4.2 for details). As mentioned before, an ϵ-shell is chosen around the boundary as the termination region in the Dirichlet case. Here, we follow a similar strategy by setting up a ϵ-region but allowing the process X_t to continue moving after it reaches the ϵ-region instead of being absorbed.

Figure 4.1 shows a strip region with width ϵ near the boundary for a bounded domain. In a spherical domain, the ϵ-region is simply an ϵ-shell near the boundary

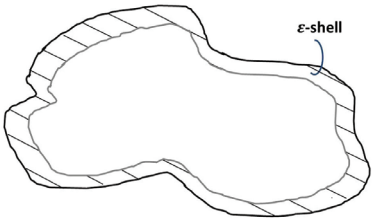


FIG. 4.1. *A ϵ-region for a bounded domain in $\mathbb{R}^3$.*

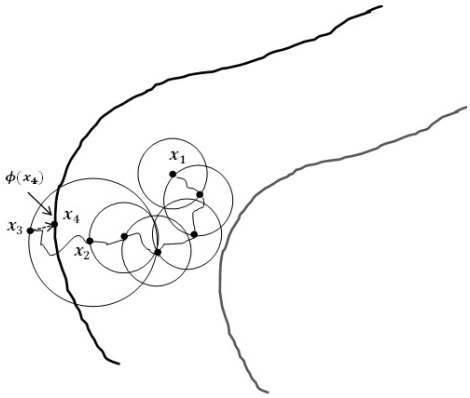


FIG. 4.2. *WOS in the ϵ-region. BM path hits x_1 in ϵ-region for the first time. Then the radius of sphere is changed to Δx, the path continues until it arrives at x_2 whose distance to ∂D is smaller or equal to Δx. When this happens, the radius of the ball is enlarged to $2\Delta x$ so that the path has a chance to run out of the domain at x_3. Then, we pull x_3 back to x_4 which is the closest point to x_3 on the boundary. Record $\phi(x_4)$, and continue the WOS-sampling of the path starting at x_4.*

of width ϵ. Denote $M_\epsilon(D)$ as the ϵ-region and $I_\epsilon(D)$ as the remaining interior region $D \backslash M_\epsilon(D)$.

Recall the discussion of the WOS in the previous section. For a BM starting at a point x in the domain, we draw a ball centered at x. The Brownian path will hit the spherical surface with a uniform probability as long as the ball does not overlap the domain boundary ∂D. The balls are constructed so that the jumps are as large as possible by taking the radius of the ball to be the distance to the boundary ∂D. We repeat this procedure until the path reaches the region $M_\epsilon(D)$. Here, we continue the WOS in $M_\epsilon(D)$ but with a fixed radius Δx much smaller than ϵ. In order to simulate the path of RBM, at some point of time the BM path will run out of the domain. For this to happen, the radius of WOS is increased to $2\Delta x$ when the path is close to boundary at a distance less than Δx. In this way, the BM path will have a chance to get out of the domain, and when that happens, we then pull it back to the nearest point on the boundary along the normal of the boundary. Afterwards, the BM path will continue as before.

In summary, a reflecting Brownian motion path is simulated by the WOS method inside D. Once it enters the ϵ-region $M_\epsilon(D)$, the radius of WOS changes to a fixed

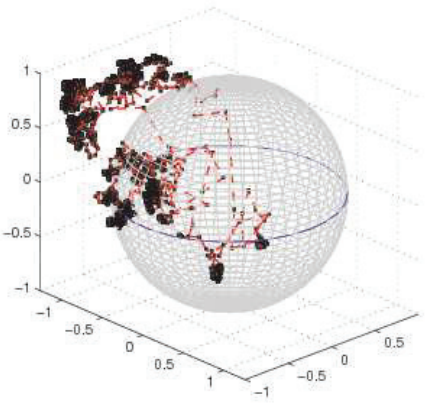


FIG. 4.3. *A RBM path within a cube in* $\mathbb{R}^3$

value, either Δx or $2\Delta x$, depending on its current distance of the Brownian particle to the boundary. When the path reaches a point on the boundary after the reflection, the radius of WOS changes back to Δx. Figure 4.2 illustrates the movement of RBM in the ϵ-region $M_\epsilon(D)$. As time progresses, we expect the path hits the boundary at some time instances and lies in either $I_\epsilon(D)$ or $M_\epsilon(D)$ at others. A RBM path is shown in Figure 4.3 within a cube of size 2.

4.2. Computing the boundary local time $L(t)$**.** Two equivalent forms of the local time have been given in Equations (2.3) and (2.4). Here we will show how the ϵ-region for the construction of the RBM in Figure 4.2 can also be used for the calculation of the local time. When the ϵ-region is thin enough, i.e. $\epsilon \ll 1$, an approximation of Equation (2.3) is given in Equation (2.8), which is the occupation time that RBM X_s sojourns within the ϵ-region during the time interval $[0,t]$. A close look at Equation (2.8) reveals that only the time spent near the boundary is involved and the specific moment when the path enters the ϵ-region has no effect on the calculation of $L(t)$.

Suppose $x \in D$ is the starting point of a Brownian path, which is simulated by the WOS method. Once the path enters the ϵ-region, the radius of WOS is changed to Δx or $2\Delta x$. It is known that the elapsed time Δt for a step of a random walk on average is proportional to the square of the step size, in fact, $\Delta t = (\Delta x)^2/d, d = 3$ when Δx is small (see Appendix A), which also applies to WOS moves (See Remark 4.2 for details). Therefore, we can obtain an approximation of the local time $L(dt)$ by counting the number of steps the path spent inside $M_\epsilon(D)$ multiplied by the time elapsed for each step, i.e.

$$L(dt) = L(t_j - t_{j-1}) \approx \frac{\int_{t_{j-1}}^{t_j} I_{D_\epsilon}(X_s) ds}{\epsilon} = (n_{t_j} - n_{t_{j-1}}) \frac{(\Delta x)^2}{3\epsilon}, \tag{4.1}$$

where dt is defined as time increment $t_j - t_{j-1}$ and $n_{t_j} - n_{t_{j-1}}$ is the number of steps that WOS steps remain in the ϵ-region during the time interval $[t_{j-1}, t_j]$. Note that in our method within the ϵ-region, the radius of the BM may be Δx or $2\Delta x$, which means the corresponding elapsed time of one step for local time will be either $(\Delta x)^2/3$ or $(2\Delta x)^2/3$. If we absorb the factor 4 into n_t, we will still have Equation (4.1). Figure

4.4 gives a sample path of the simulated local time associated with the RBM in Figure 4.3.

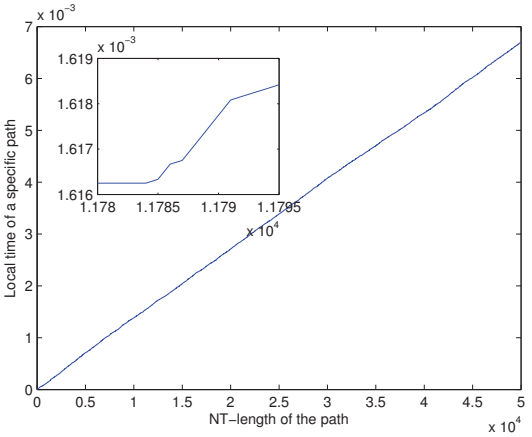


FIG. 4.4. *Boundary local time (4.1) increases when the path runs into the region $M_\epsilon(D)$. The insert shows the piecewise linear profile of the local time path with flat level regions. By our construction, most of the path will fall into the ϵ-region. If x-axis changes to time line, the graph should be more flatter than it appears here which implies that boundary local time increases only on a small set.*

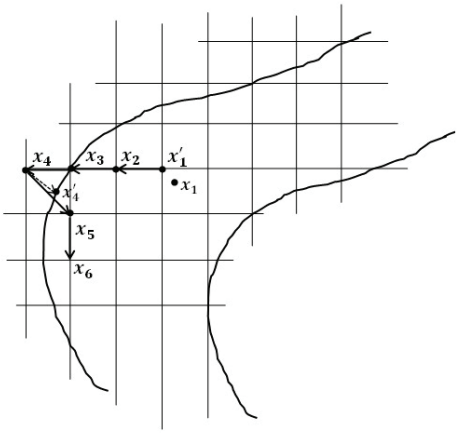


FIG. 4.5. *Random walks on the lattice in ϵ-region. A BM path hits $x_1 \in M_\epsilon(D)$ by the WOS method. Replace x_1 by the nearest grid point x_1'. Then, several steps of random walks will make a path as $x_2 \to x_3 \to x_4$. Since $x_4 \notin D$, we push it back along the normal line (dash arrow) to x_4' then replace it by the closest grid point within domain (solid arrow) x_5. Here path crosses the boundary at $x_4' \in \partial D$. Then continue the random walk as usual to x_6.*

In practical implementation, we treat n_t as a vector of entries of increasing value, the increment of each component of n_t over the previous one after each step of WOS will be 0, 1 or 4, corresponding to the scenarios that X_t is out of the ϵ-region, in the ϵ-region while sampled on the sphere of a radius Δx, or in the ϵ-region while sampled on the sphere of a radius $2\Delta x$, respectively.

REMARK 4.1 (Alternative way to compute local time $L(t)$). From Equation (2.4), the local time increases if and only if the RBM path hits the boundary, which implies that the time before the path hits the boundary makes no contribution to the increment of the local time. Thus, a WOS method with a changing radius can also be used with Equation (2.4). Specifically, we divide the time interval $[0,t]$ into N small subintervals of equal length. In each $[t_i,t_{i+1}]$ the Brownian path will move $2\Delta x$ or Δx with the WOS method when the current path lies within a distance less or more than Δx to the boundary. If the path hits or crosses the boundary within $[t_i,t_{i+1}]$, then $L(t)$ will increase by $\sqrt{\pi/2}\sqrt{t_{i+1}-t_i}$.

REMARK 4.2 (Approximating RBM by WOS or random walks on a lattice – a comparison). There are two ways to find approximation to Brownian paths inside the region $M_\epsilon(D)$ and construct their reflections once they get out of the domain. One way is by using the WOS approach as described in Section 4.1. The other is in fact to use a random walk on a lattice inside $M_\epsilon(D)$. Both belong to the random walk techniques while WOS prevails in homogeneous media without consideration of the whole trajectories of paths and random walk on a lattice is widely used in various other situations. In the second approach, as illustrated in Figure 4.5, a grid mesh is set up over $M_\epsilon(D)$ and the random walk takes a one-step walk on the lattice until the path goes out of the domain and then it will be pushed back to the nearest lattice point inside $M_\epsilon(D)$. And the elapsed time for a Δx walk is again on average $(\Delta x)^2/3$ as shown in Appendix A. The boundary local time $L(t)$ can still be calculated as in Equation (4.1). The problem with this approach is that a random walk on the lattice only considers six directions in $\mathbb{R}^3$ while a Brownian motion actually should have equal probability to go in all directions in the space. This limitation was found in our numerical tests to lead to insufficient accuracy in simulating reflecting Brownian motions.

Meanwhile, the WOS method in the ϵ-region $M_\epsilon(D)$ has a fixed radius Δx, which enables us to calculate the boundary local time by Equation (4.1) since the elapsed time of a Δx move in $\mathbb{R}^3$ on average remains to be $(\Delta x)^2/3$. This conclusion can be heuristically justified by considering points on the sphere as linear combinations of the directions along the three axes, which implies that the average time that the path hits the sphere with a radius Δx should also be the same. As discussed before, if the path comes within a distance very close to the boundary, say less than Δx, the radius of the WOS method is increased to $2\Delta x$ so that it will have a chance to run out of the domain and then be pushed back to the nearest point on the boundary to realize a hit of the RBM on the boundary.

4.3. Probabilistic representation for the Neumann problem. Finally, with the boundary local time of RBM available, we can discuss the approximation of the Neumann problem solution $u(x)$ using the probabilistic approach (2.10). First of all, we will need to truncate the infinite time duration required for the RBM path X_t in Equation (2.10) to a finite extent for computer simulations. The exact length of truncation will have to be numerically determined by increasing the length until a convergence is confirmed (namely, the approximation to $u(x)$ does not improve within a prescribed error tolerance between two different choices of truncation times under same number of sampled paths). Assume that the time period is limited to from 0 to T, then by a Monte Carlo sampling of RBM paths, an approximation of Equation (2.10) will be

$$\widetilde{u}(x) = \frac{1}{2\epsilon}\sum_{i=1}^{N}\left[\int_0^T \phi(X_t^i) I_{\partial D}(X_t^i)\int_t^{t+dt} I_{D_\epsilon}(X_s^i)ds\right], \tag{4.2}$$

where $X_t^i, i=1,...,N$ are stochastic processes sampled according to the law of RBM.

Next, let us see how the RBM can be incorporated into the representation formula once its path is obtained.

Associate the time interval $[0,T]$ with the number of steps NT of a sampling path, NT will give the total length of each path. Then, the integral inside the square bracket in Equation (4.2) can be transformed into

$$\sum_{j'=1}^{NT}\left(\phi(X_{t_j}^i)I_{\partial D}(X_{t_j}^i)\int_{t_{j-1}}^{t_j} I_{D_\epsilon}(X_s^i)ds\right), \tag{4.3}$$

where j' stands for the j'th step of the WOS method, and j corresponds to a step for which $X_{t_j}^i \in \partial D$ and summation is only done over those j's.

As the integral in Equation (4.3) is in fact the occupation time as shown in Equation (4.1), Equation (4.3) becomes

$$\sum_{j'=1}^{NT}\left(\phi(X_{t_j}^i)I_{\partial D}(X_{t_j}^i)(n_{t_j}-n_{t_{j-1}})\frac{(\Delta x)^2}{3}\right). \tag{4.4}$$

As a result, an approximation to the PDE solution $\widetilde{u}(x)$ becomes

$$\widetilde{u}(x)=\frac{1}{2\epsilon}\sum_{i=1}^{N}\left[\sum_{j'=1}^{NT}\left(\phi(X_{t_j}^i)I_{\partial D}(X_{t_j}^i)(n_{t_j}-n_{t_{j-1}})\frac{(\Delta x)^2}{3}\right)\right]. \tag{4.5}$$

Theoretically speaking, ϵ should be chosen much larger than Δx. Here, we take $\epsilon=k\Delta x$, $k>1$ is an integer, which will increase as Δx vanishes to zero. Then, Equation (4.5) reduces to

$$\begin{aligned}\widetilde{u}(x)&=\frac{1}{2k\Delta x}\sum_{i=1}^{N}\left[\sum_{j=1}^{NT}\left(\phi(X_{t_j}^i)I_{\partial D}(X_{t_j}^i)(n_{t_j}-n_{t_{j-1}})\frac{(\Delta x)^2}{3}\right)\right]\\&=\frac{\Delta x}{6k}\sum_{i=1}^{N}\left[\sum_{j=1}^{NT}\left(\phi(X_{t_j}^i)I_{\partial D}(X_{t_j}^i)(n_{t_j}-n_{t_{j-1}})\right)\right],\end{aligned} \tag{4.6}$$

which is the final numerical approximation for the Neumann problem. In the following we present the general implementation of the numerical algorithm.

Let x be any interior point in D where the solution $u(x)$ for the Neumann problem is sought. First, we define the ϵ-region $M_\epsilon(D)$ near the boundary. For each one of N RBM paths, the following procedure will be executed until the length of the path reaches a prescribed length given by NT:

1. If $x \notin M_\epsilon(D)$, predict next point of the path by the WOS with a maximum possible radius until the path locates near the boundary within a certain given distance ϵ, say $\epsilon=3\Delta x$ (hit the ϵ-region $M_\epsilon(D)$). If $x\in M_\epsilon(D)$, $l(t_i)$=1 or 4; otherwise, $l(t_i)=0$. Here $l(t)$ is the unit increment of $L(t)$ at time t.
2. If $x\in M_\epsilon(D)$, use the WOS method with a fixed radius Δx (if $\text{dist}(x,\partial D)\geq\Delta x$) or $2\Delta x$ (if $\text{dist}(x,\partial D)<\Delta x$) to predict the next location for Brownian path. Then, execute one of the two options:

Option 1. If the path happens to hit the domain boundary ∂D at x_{t_i}, record $\phi(x_{t_i})$.

Option 2. If the path crosses the domain boundary ∂D, then pull the path back along the normal to the nearest point on the boundary. Record the Neumann value at the boundary location.

Due to the independence of the paths simulated with the Monte Carlo method, we can run a large number of paths simultaneously on a computer with many cores in a perfectly parallel manner, and then collect all the data at the end of the simulation to compute the average. Algorithm 1 gives a pseudo-code for the numerical realization of implementing the WOS in both $I_\epsilon(D)$ and $M_\epsilon(D)$ regions.

As described in this section, it is quite clear that calculation of the distance to the boundary accounts for a large portion of computing time in our algorithm, especially when the Brownian path is out of the ϵ-shell. For simple domains like a cube and a sphere in $\mathbb{R}^3$, this distance can be found easily and thus consumes little time. In the ellipsoid case, the distance is still computable which involves the calculation of normal directions and thus requires much more time than the former cases. For more general domains, more efficient numerical methods are desired.

5. Numerical results

In this section, we give the numerical results for the Neumann problem in a cube, a sphere, an ellipsoid, and a non-smooth domain.

To monitor the accuracy of the numerical approximation of the solutions, we select a circle inside the domain, where the solution of the PDE $u(x)$ will be found by the proposed numerical methods, defined by

$$\{(x,y,z)^T = (r\cos\theta_1\sin\theta_2, r\sin\theta_1\sin\theta_2, r\cos\theta_2)^T\} \tag{5.1}$$

with $r=0.6$, $\theta_1 = 0, 2\pi/30, 4\pi/30, ..., \pi$, $\theta_2 = 13\pi/30$ with 15 different θ_1 in ascending order. In addition, a line segment will also be selected as the locations to monitor the numerical solution, the endpoints of the segment are $(0.4,0.4,0.6)^T$ and $(0.1,0,0)^T$, respectively. Fifteen uniformly spaced points on the circle or the line are chosen as the locations for computing the numerical solutions.

The true solution of the Neumann problem (1.1) with the corresponding Neumann boundary data is taken to be

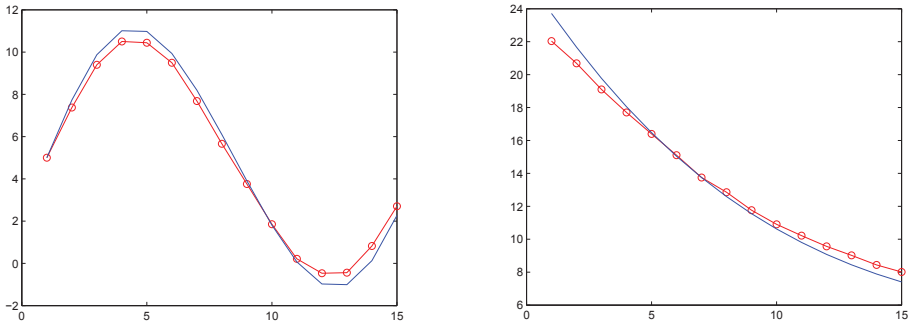
$$u(x) = \sin 3x \sin 4y \; e^{5z} + 5. \tag{5.2}$$

In the figures of numerical results given below, the (blue) curves are the true solutions and the (red) curves with circles are the approximations. The numerical solutions are shifted by a constant so they agree with the exact solution at one point as the Neumann problem is only unique up to an arbitrary additive constant. "Err" indicates the relative error of the approximations.

5.1. Convergence rate study. The analysis of the errors of our numerical methods is complex as it involves several inter-connected factors, the time truncation T, the radius of the WOS sphere inside the ϵ-layer and the layer's thickness, and the number of Brownian paths.

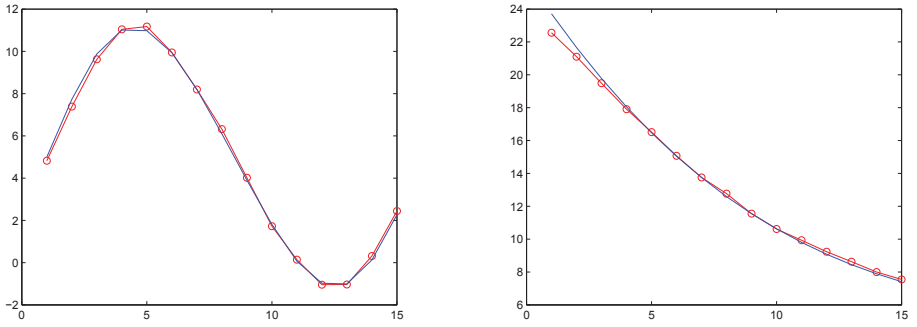
A cubic domain of size 2 is selected to test the choice of the number of paths and the length of the paths in the numerical formula (4.6). Taking the cubic domain avoids a source of errors in computing the projection of the paths onto the domain boundary.

First we consider the proper choice of the truncation length NT parameter of the Brownian paths. The step-size $\Delta x = 5 \times 10^{-4}$ is used as the radius of the WOS inside the ϵ-region $M_\epsilon(D)$, namely, the step-size of the random walk approximation of the RBM near the boundary. The strip width ϵ is chosen to be $3\Delta x$. The number of paths is taken as $N = 2 \times 10^5$. Two choices for the path length parameter $NT = 2.7 \times 10^4$ and $NT = 2.9 \times 10^4$ for the circle ($NT = 2.4 \times 10^4$ and $NT = 2.5 \times 10^4$ for the line segment) are compared to gauge the convergence of the numerical formula in terms of the path truncation. Figure 5.1 and Figure 5.2 show the solution and the relative errors in both cases, which indicates that $NT = 2.9 \times 10^4$ and $NT = 2.5 \times 10^4$ will be sufficient to give an error below 3% for the circle and the line segment, respectively, as shown in Figure 5.2.



(a) $\epsilon = 3\Delta x$, Err = 10.50%, NT = 2.7×10^4 (b) $\epsilon = 3\Delta x$, Err = 3.83%, NT = 2.4×10^4

FIG. 5.1. *Cubic domain: number of paths* $N = 2 \times 10^5$. *(Left): Solution on the circle. (Right): Solution on a line segment.*



(a) $\epsilon = 3\Delta x$, Err = 2.87%, NT = 2.9×10^4 (b) $\epsilon = 3\Delta x$, Err = 1.78%, NT=2.5×10^4

FIG. 5.2. *Cubic domain: number of paths* $N = 2 \times 10^5$. *(Left): Solution on the circle. (Right): Solution on a line segment.*

Algorithm 1: The algorithm for the probabilistic solution of the Laplace equation with the Neumann boundary condition

---

Data: Select integers N and NT, a starting point $X_0 \in D$, step size h and ϵ-region $M_\epsilon(D)$ near the boundary.
Output An approximation of $u(X_0)$.

Initialize $L[NT], v[NT], u[N]$, $X = X_0$, $i \leftarrow 1$ and $j \leftarrow 1$;

While $i \le N$ **do**
  Set $S_i = 0$.
  While $j \le NT$ **do**
    If $X \in I_\epsilon(D)$ **then** /* If the path has not touched the ϵ-region */
      Set $L[j] \leftarrow 0$; /*Increment of local time at each step. */
      Set $r \leftarrow d(X, \partial D)$; /* Find the distance to the boundary */
      Randomly choose a point X_1 on $B(X, r)$ then set $X \leftarrow X_1$.
    Else /* The path enters the ϵ-region */
      Set $r \leftarrow$ h (2h); /* If $d(X, \partial D) > h$ or $=0$ $(0 < d(X, \partial D) \le h)$ */
      $L[j] \leftarrow 1(4)$; /*local time increases */
      Randomly choose a point X_1 on $B(X, r)$ then set $X \leftarrow X_1$.
      If $X \notin \bar{D}$, **then**
        Find X_j to be the nearest point on ∂D to X and pull X back onto ∂D at X_j;
        Set $X \leftarrow X_j$;
        Set $v[j] \leftarrow \phi(X_j)$
      End
    End
    $j \leftarrow j + 1$;
  End
  count $\leftarrow 0$;
  For k=1:NT
    count $\leftarrow$ count + $L[k]$;
    If $v[k] \sim = 0$ **then**
    $u[i] \leftarrow u[i] + \phi(X_k)\cdot$count;
    count $\leftarrow 0$;
  End
 $i \leftarrow i + 1$;
End
Return $\widetilde{u}(X_0) = h \sum_{k=1}^{N} u[k]/N/(6k)$

It should be noted that the parameter NT for the length of the path (in terms of number of WOS steps) does not correspond to the physical time T. This is due to the fact that no elapsed time estimate is known for a WOS step of a large sphere radius inside the interior of the domain. In theory, the larger the truncation time T, the more accurate is the probabilistic solution for the Neumann solution. Theoretical variance estimate on the truncation of the time T has been given in [21]. However, for a fixed spatial mesh size Δx, a too long time integration will result in the accumulation of time discretization error for the Brownian paths, thus leading to the degeneracy of the

numerical solutions as shown in our numerical experiments. Therefore, the choices for NT and Δx for our method are more complicated than that for the number of paths N, and have been tested, as discussed above, to give about 3% relative errors in our simulations.

The traditional Monte Carlo methods for computing high dimensional integrals has a $1/\sqrt{N}$ convergence rate where N is the number of samples in the simulation. However, in computing the path averages in the Feynman–Kac formula for the Neumann solution, the accuracy for the local time will also affect the overall accuracy of numerical methods. Regarding the former, Révész [10] proposed several approximations of the local time for 1D Brownian motions with convergence rates ranging from $O(\Delta t^{1/4})$ to $O(\Delta t^{1/3})$. Such an analytical result is expected to hold in higher dimensional Brownian motions. As such, we expect the overall convergence rate for our Neumann solution will be limited by that of the local time.

In Table 5.1, we have included the relative errors of the Neumann solution monitored along a circle and a line segment in terms of the number of paths N. For all simulations, $\Delta x = 5\times 10^{-4}$ and the strip width ϵ is chosen to be $3\Delta x$. The numerical results show that the convergence rate is around $O(1/N^{\alpha}), \alpha = 0.29$, which is less than the $O(1/\sqrt{N})$ convergence rate of Monte Carlo integrations. Meanwhile, the relative error of 2.87×10^{-2} for the Neumann solution is approximately at the same order of $\Delta t^{1/4} = 1.68\times 10^{-2}$ for $\Delta t = (\Delta x)^2/3, \Delta x = 5\times 10^{-4}$, reflecting the error estimate of local time in [10]. In

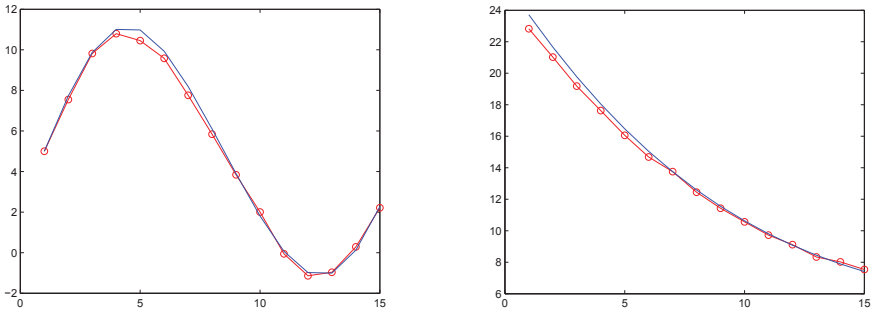
	$NT = 2.5\times 10^4$	α	$NT = 2.9\times 10^4$	α
N	errors on the line		errors on the circle	
2×10^2	0.2316		1.0158	
2×10^3	0.0364	0.37	0.4395	0.52
2×10^4	0.0345	0.15	0.0717	0.59
2×10^5	0.0178	**0.29**	0.0287	**0.39**

TABLE 5.1. *Convergence rate of relative errors as $O(1/N^{\alpha})$ for the Neumann solution in a cube where $\Delta x = 5\times 10^{-4}, \epsilon = 3\Delta x$.*

the rest of the numerical tests below, we will set $\Delta x = 5\times 10^{-4}$ and the number of path $N = 2\times 10^5$, but change NT for different boundaries.

5.2. Spherical domain. The unit ball is centered at the origin. Similar numerical results are obtained as in the case of the cube domain. Here, the reflected points of Brownian path are the intersection of the normal and the domain. Though both Figure 5.3(a) and (b) shows some deviations in the middle, the overall approximation are within an acceptable relative error less than 5.26%.

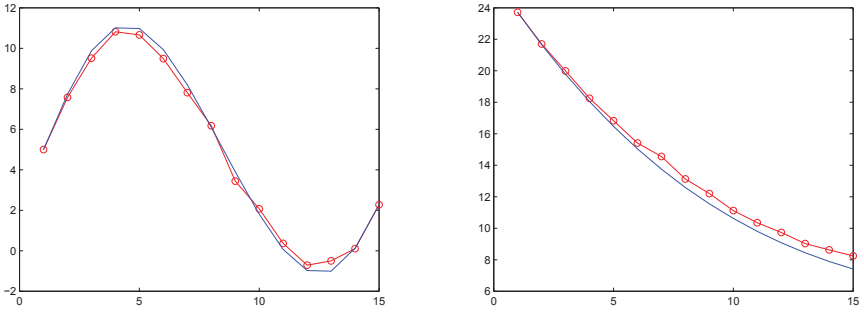
5.3. Ellipsoidal domain. We use the ellipsoid with axis lengths (3, 2, 1) centered at the origin. Δx remains to be 5×10^{-4}. The numerical results (Figure 5.4) on the circle are less accurate than those for the cubic and spherical domains. A possible reason is as follows. An ellipsoid has "corners" around the longest axis if the lengths of three axis are not the same. When the initial point of the Brownian particle is far away from the "corners", the Brownian paths have a smaller probability to run into the regions close to those "corners". This implies that a large number of sampled paths starting at those points may stay away from the "corners", which may undermine the calculation accuracy of $u(x)$ since $u(x)$ is the weighted average over of Neumann data



(a) $\epsilon = 3\Delta x$, Err = 5.26%, NT = 5×10^4 (b) $\epsilon = 3\Delta x$, Err = 2.00%, NT = 4.5×10^4

FIG. 5.3. *Spherical domain: number of paths* $N=2\times10^5$. *(Left): Solution on the circle. (Right): solution on a line segment.*

at hitting positions of RBM on the whole boundary. The numerical solutions along the line segment show a better accuracy.



(a) $\epsilon = 3\Delta x$, Err = 8.85%, NT = 5.025×10^4 (b) $\epsilon = 3\Delta x$, Err = 1.69%, NT = 4.525×10^4

FIG. 5.4. *Ellipsoidal domain: number of paths* $N=2\times10^5$. *(Left): Solution on the circle. (Right): solution on a line segment.*

5.4. Exterior Neumann problem for a domain of multiple spheres. Now we consider the exterior Neumann problem of a domain whose boundary is formed by multiple spheres, as shown in Figure 5.5. Thirty small hemispheres, with varying radii ranging from 0.12 to 0.18, are superimposed on the surface of a unit sphere with no overlapping between the small spheres. The solution domain is formed by the big sphere and thirty small hemispheres. We will calculate the potential on the boundary for the exterior Neumann problem of this domain. Using the Feynman–Kac formula, we can find the potential at any single point within the domain. By our algorithm, we can simply place the point on the boundary and start the reflecting Brownian path from that point. Solutions at fourteen points with ten points on the unit sphere and four points on the small hemispheres are calculated. For this case, the analytical solution is set to be $10/\sqrt{(x-0.5)^2+y^2+z^2}$. Figure 5.6 shows the satisfactory results of the algorithm on the boundary with a relative error around 2%. Note that for the exterior

problem, the Brownian path may go to infinity so a large boundary is needed to mark the paths considered to be far away from the domain boundary, namely, escaping to infinite, and here a much larger sphere with radius $20R(R=1)$ is chosen. Finally, the number of Monte Carlo path samples is taken to be 2×10^4, giving the obtained results.

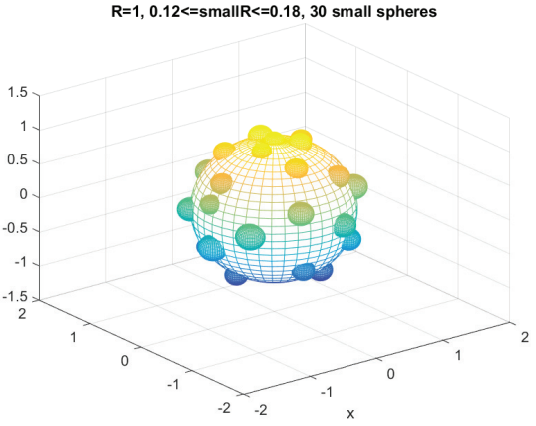


FIG. 5.5. *A domain with a boundary formed by multiple small hemispheres and a unit sphere.*

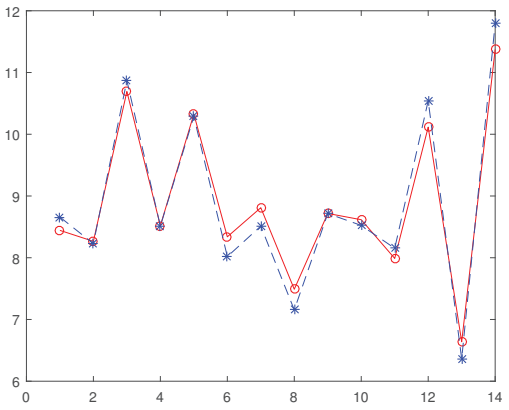


FIG. 5.6. *Exterior Neumann problem for the domain in Figure 5.5: Exact potentials (asterisks) and numerical solutions (circles) on the boundary with a relative error of 2.31%.* $\Delta x=5\times10^{-4}$, $\epsilon = 3\Delta x$, $N=2\times10^4$, $NT=5\times10^4$.

5.5. Some numerical issues and comparison with existing MC and grid-based methods.

5.5.1. Calculation of distance and parallel implementation. It is clear that the calculation of the distance to the boundary accounts for a large portion of computing time for the WOS algorithm. In the multiple-spheres example, it takes a little time to find the closest point on the boundary by looping over all the thirty small

hemispheres. But, consider the case of a boundary formed by many more small spheres, e.g. 1000 spheres, it will be time-consuming to calculate the distance. However, if the number of unique radii of the small spheres is small, some existing packages, for instance nanoflann library [3], can be used to speed up the closest-point query. But for a general domain, a regular Cartesian mesh covering the whole domain can be used to facilitate the calculation. As the signed distance function is a smooth function of position if it is given a negative value of the absolute distance outside the domain, we can pre-calculate the signed distance function at the regular mesh points, and the distance for any interior point to the boundary can be approximated accurately by appropriate high order interpolations of the distance function values on the mesh points. This greatly reduces the computing time at each step of the WOS algorithm. Also, as our algorithm requires the exact reflecting point on the boundary when the path exits the domain, a Newton root-finding method will be used to locate the nearest point on the boundary along the inward normal.

The MC approach of our method is intrinsically parallel as paths initiating from the location where the PDE solution is sought after are independent, a large number of paths can be sampled simultaneously on a multiple-core/nodes computer in a perfectly parallel manner and only at the end of the simulation the data is collected to compute the average in the Feynman–Kac formula.

5.5.2. Memory use and CPU time for sampling paths. For each sample path, the hitting locations and corresponding local time are recorded to evaluate the average of contributions of each path. Therefore, the memory storage required for the algorithm is $O(N \cdot NT)$ given N paths, each of length NT. If distance functions are pre-calculated, as needed for a domain of general shape, additional memory of $O(N_1^3)$ (N_1 be the number of subdivision along each dimension of the domain), is needed to store the distance function values on the mesh. The CPU time of the algorithm mainly arises from the calculation of the distance to the boundary and the reflecting points. For cube and sphere domains, it is trivial and thus quick to compute. For general domains, one should also take into account of CPU time spending on the Newton method used to find the reflecting points on the boundary. Naturally, the overall CPU time depends on the number of sample paths, length of each path in proportion.

5.5.3. Comparison to an existing MC for the Neumann problem. A Monte Carlo method for the Neumann problem of the Poisson equation was proposed in [21] where WOS method was also used to simulate Brownian motions. The main difference from the algorithm in this paper is about how to treat the Brownian paths once they enter the ϵ-layer of the boundary. In [21], once the path enters the ϵ-layer, the path is then projected onto the boundary, say at (x,y). And, in order to continue the walk back into the domain, with the help of the Neumann data $\phi(x)$, a Taylor expansion for the PDE solution is used to relate the solutions at a grid point of a local regular mesh of size h inside the domain, say $(x+h,y)$, and at the boundary in the form of $u(x,y)=u(x+h,y)+2h\phi(x)$. A randomization of this relation shows the score of the walk should be increased by an amount of $2h\phi(x)$ and the walk arrives at the new position $(x+h,y)$ inside the domain. Moreover, in order to introduce the elapsed time for each walk, a kinetic approximation of the diffusion operator by a neutron transport operator is used such that the new position of the walk can now be at $(x+hv_xt_c, y+hv_yt_c)$, where (v_x,v_y) is the velocity of the walk, t_c is a small parameter, and h^2t_c will be the elapsed time to arrive this new position. A Taylor expansion of the solution will give a similar afore-mentioned relation, which indicates a different score increment $4\phi(x)h/\pi$. And, the motion of the walk continues until the pre-set truncation

time of the path is reached.

In terms of complexity and memory usage, the MC algorithm of [21] is similar to the method introduced in this paper as it only needs to record the locations when the Brownian path enter ϵ-layer and the projected position onto the boundary and the times (or stepsizes) at each walk step.

However, there is one difference in the way when a path is considered having hit the boundary and then reflected back into the domain. In our algorithm, the path has to "physically" cross the boundary before it is treated as a RBM path. In the approach of [21], once a path enters the ϵ-layer, it is then considered as a RBM path, as a result, this approach may treat a Brownian path which enters the layer, then leaves it without hitting the boundary for a while, also as behaving as a RBM path, in fact the local time increment for this path before it re-enters the ϵ-layer should be zero. The effect of this happening to the averages in the Feynman–Kac formula is not clear and should be investigated in relation to the thickness of the ϵ-layer .

5.5.4. Comparison to grid-based method. Comparing to grid based method such as finite element method with a mesh size $\Delta x = \Delta y = \Delta z = \dfrac{L}{N}$ for a cube of $[0, L]^3$, an unstructured mesh as in a general domain will require $O(N^3)$ memory while the generation of the mesh itself will also take large amount of CPU time. An iterative solver such as GMRES or conjugate gradient will take $O(pN^3)$ flops for p-iterations. The main difference is that the grid-based method has to find the solution in the whole solution domain while the Monte Carlo method allows the solution at one single point. For the Monte Carlo method used in this paper, to find the solution at one single point, we only need to record the times and locations where the reflecting Brownian path hits the boundary for each of the N paths and no mesh generation and linear solver is required either.

6. Conclusions and discussions

In this paper we have proposed numerical methods for computing the local time of reflecting Brownian motion and the probabilistic solution of the Laplace equation with the Neumann boundary condition. Without knowing the complete trajectories of RBM in space, we are able to use the WOS to sample the RBM and calculate its local time, based on which a discrete probabilistic representation (4.6) was obtained to produce satisfactory approximations to the solution of the Neumann problem at one single point. Numerical results validated the stability and accuracy of the proposed numerical methods.

In addition, random walk on a lattice was also investigated as an alternative way to sample RBM. However, numerical experiments show that the numerical results are inferior to those obtained by the WOS method. A possible reason is that formula (2.3) for the local time is valid for a smooth path while a random walk approximation of the the Brownian path contains inherent errors.

The local time can also be computed by a mathematically equivalent formula (2.4), for which the implementation is discussed briefly in Section 4.2. Again the numerical results based on Equation (2.4) are inferior to those obtained using the original limiting process of Lévy in [19] . We believe that this fact may result from the time discretization error of Brownian paths especially when long time truncation is employed in the probabilistic representation.

Various issues affecting the accuracy of the proposed numerical methods remain to be further investigated, such as the number of random walk or WOS steps and the truncation of duration time T for the paths [21], the choice of the thickness for the

ϵ-region, the size of Δx for the lattice, etc. Regarding the convergence rate of the proposed method as discussed in Section 5.1, more theoretical work is needed on the exact rate of convergence.

Appendix A. For the random walk on a lattice as in Figure A.1 to converge to a

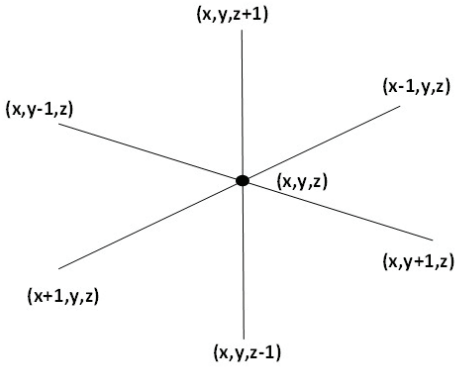


FIG. A.1. *Central difference scheme in* $\mathbb{R}^3$.

continuous BM, a relationship between Δt and Δx in $\mathbb{R}^3$ will be needed and is shown to be

$$\Delta t = \frac{(\Delta x)^2}{3}. \tag{A.1}$$

The following is a proof of this result (See [6] for a reference). The density function of standard BM satisfies the following PDE [15]

$$\frac{\partial p}{\partial t} = \frac{1}{2}\Delta_x p(t,x,y)\ . \tag{A.2}$$

By using a central difference scheme and changing p to v, Equation (A.2) becomes

$$\frac{v^{n+1}_{i,j,k} - v^n_{i,j,k}}{\Delta t} = \frac{1}{2}\frac{v^n_{i+1,j,k} + v^n_{i-1,j,k} + v^n_{i,j+1,k} + v^n_{i,j-1,k} + v^n_{i,j,k+1} + v^n_{i,j,k-1} - 6v^n_{i,j,k}}{(\Delta x)^2}\ , \tag{A.3}$$

where i,j,k are the indices of grid points on the lattice with respect to the three axes.

Reorganizing Equation (A.3) and letting $\lambda = \Delta t/(2(\Delta x)^2)$, we have

$$v^{n+1}_{i,j,k} = \lambda v^n_{i+1,j,k} + \lambda v^n_{i-1,j,k} + \lambda v^n_{i,j+1,k} + \lambda v^n_{i,j-1,k} + \lambda v^n_{i,j,k+1} + \lambda v^n_{i,j,k-1} + (1-6\lambda)v^n_{i,j,k}\ , \tag{A.4}$$

By setting $\lambda = \dfrac{1}{6}$, we have

$$v^{n+1}_{i,j,k} = \frac{1}{6}v^n_{i+1,j,k} + \frac{1}{6}v^n_{i-1,j,k} + \frac{1}{6}v^n_{i,j+1,k} + \frac{1}{6}v^n_{i,j-1,k} + \frac{1}{6}v^n_{i,j,k+1} + \frac{1}{6}v^n_{i,j,k-1}. \tag{A.5}$$

Using the initial condition ϕ, we have

$$v^{n+1}_{i,j,k} = \sum_{i',j',k'} C_{i',j',k'}\,\phi\left(\sum_{l=1}^{n}\vec{\eta_l}\right), \tag{A.6}$$

where

$$\vec{\eta_l} = \begin{cases} (-h,0,0)^T, & \text{prob} = \dfrac{1}{6} \\ (h,0,0)^T, & \text{prob} = \dfrac{1}{6} \\ (0,h,0)^T, & \text{prob} = \dfrac{1}{6} \\ (0,-h,0)^T, & \text{prob} = \dfrac{1}{6} \\ (0,0,h)^T, & \text{prob} = \dfrac{1}{6} \\ (0,0,-h)^T, & \text{prob} = \dfrac{1}{6} \end{cases}, \tag{A.7}$$

and

$$\sum_{l=1}^{n} \vec{\eta_l} = \begin{pmatrix} -n+2i'+i \\ -n+2j'+j \\ -n+2k'+k \end{pmatrix} h. \tag{A.8}$$

Let $\vec{\eta_l} = (x_l, y_l, z_l)^T$, then

$$x_l = \begin{cases} -h, \text{ prob} = \dfrac{1}{6} \\ h, \text{ prob} = \dfrac{1}{6} \\ 0, \text{ prob} = \dfrac{2}{3} \end{cases}, \tag{A.9}$$

for each l. We known that y_l, z_l have the same distribution as x_l.

Notice that the covariance between any two of x_l, y_l, z_l is zero, i.e.

$$E(x_l y_l) = 0,\ E(y_l z_l) = 0, \text{ and } E(x_l z_l) = 0.$$

So

$$E(\sum_{i=1}^{n} x_l \sum_{i=1}^{n} y_l) = 0,\ E(\sum_{i=1}^{n} y_l \sum_{i=1}^{n} z_l) = 0, \text{ and } E(\sum_{i=1}^{n} x_l \sum_{i=1}^{n} z_l) = 0.$$

According to the central limit theorem, we have

$$\sum_{i=1}^{n} x_l \overset{D}{=} N\left(0, \frac{nh^2}{3}\right) \ as \ \ n \to \infty. \tag{A.10}$$

The same assertion holds for $\sum_{i=1}^{n} y_l$ and $\sum_{i=1}^{n} z_l$.

Since $\lambda = \dfrac{\Delta t}{2(\Delta x)^2} = \dfrac{1}{6}$, then $h^2 = 3k$ and, hence, $\dfrac{nh^2}{3} = nk = t$. Therefore $\sum_{i=1}^{n} x_l \sim N(0,t)$ as $n \to \infty$. So are $\sum_{i=1}^{n} y_l$ and $\sum_{i=1}^{n} z_l$.

Recall that the covariance between any pair of $\sum_{i=1}^{n} x_l$, $\sum_{i=1}^{n} y_l$, and $\sum_{i=1}^{n} z_l$ is zero, that $\sum_{i=1}^{n} x_l$, $\sum_{i=1}^{n} y_l$, and $\sum_{i=1}^{n} z_l$ become independent normal random variables as $n \to \infty$. Hence,

$$C_{i',j',k',n} = P\left\{ \sum_{l=1}^{n} \vec{\eta_l} = \begin{pmatrix} -n+2i'+i \\ -n+2j'+j \\ -n+2k'+k \end{pmatrix} h = \begin{pmatrix} \sum_{i=1}^{n} x_l \\ \sum_{i=1}^{n} y_l \\ \sum_{i=1}^{n} z_l \end{pmatrix} \right\} \xrightarrow{D} \frac{1}{(2\pi t)^{3/2}} e^{\frac{-\|\vec{x}-\vec{x_0}\|^2}{2t}}, \quad \text{(A.11)}$$

and

$$v_{i,j,k}^{n+1} = \sum_{i',j',k'} C_{i',j',k',n} \phi(\sum_{l=1}^{n} \vec{\eta_l}) \to \iiint_{\mathbb{R}^3} \frac{1}{(2\pi t)^{3/2}} e^{\frac{-\|\vec{x}-\vec{x_0}\|^2}{2t}} \phi(\vec{x}) d\vec{x}, \quad \text{(A.12)}$$

which coincides with the density function of the 3D standard BM.

In conclusion, when $\frac{\Delta t}{2(\Delta x)^2} = \frac{1}{6}$, i.e. $\Delta t = \frac{(\Delta x)^2}{3}$ or $\sqrt{dt} = \frac{dx}{\sqrt{3}}$, the central difference scheme converges to the standard BM in 3D. Generally, the result can be extended to d-dimensional Euclidean space and the result will be $\Delta t = \frac{(\Delta x)^2}{d}$.

Acknowledgement. The authors Y.J.Z. and W.C. acknowledge the support of the National Science Foundation (DMS-1315128) and the National Natural Science Foundation of China (No. 91330110) for the work in this paper. For the present work, the author E. H. was supported in part by a Simons Foundation Collaboration Grant for Mathematicians and by a research grant administered through the University of Science and Technology of China. The authors appreciate the constructive comments from a reviewer during the revision process.

REFERENCES

[1] A. Benchérif-Madani and E. Pardoux, *A probabilistic formula for a Poisson equation with Neumann boundary condition*, Stoch. Anal. Appl., 27:739–746, 2009.
[2] I. Binder and M. Braverman, *The rate of convergence of the walk of sphere algorithm*, Geometric and Functional Analysis, 22:558–587, 2012.
[3] J.L. Blanco, *nanoflann: a C++ header-only fork of FLANN a library for Nearest Neighbor*, (NN) wih KD-trees. https://github.com/jlblancoc/nanoflann, 2014.
[4] M. Bossy, E. Gobet, and D. Talay, *Symmetrized Euler scheme for an efficient approximation of reflected diffusions*, Journal of Applied Probability, 41(3):877–889, 2004.
[5] G.A. Brosamler, *A probabilistic solution of the Neumann problem*, Mathematica Scandinavica, 38:137–147, 1976.
[6] A.J. Chorin and O.H. Hald, *Stochastic Tools in Mathematics and Science*, Dordrecht: Springer, 1, 2009.
[7] K.L. Chung and R.J. Williams, *Introduction to Stochastic Integration*, Progress in Probability and Statistics, 4, 1983.
[8] K.L. Chung, *Green, Brown, and Probability*, Singapore: World scientific, 1995.
[9] C. Costantini, B. Pacchiarotti, and Flavio Sartoretto, *Numerical approximation for functionals of reflecting diffusion processes*, SIAM J. Appl. Math., 58(1):73–102, 1998.

[10] M. Csörgő and P. Révész, *Three Strong Approximations of The Local Time of a Wiener Process and Their Applications to Invariance*, North-Holland, 1984.
[11] F. Fogolari, A. Brigo, and H. Molinari, *The Boltzmann equation for biomolecular electrostatics: a tool for structural biology*, J. Mol. Recognit., 15:377–392, 2002.
[12] M. Freidlin, *Functional Integration and Partial Differential Equations*, Princeton University Press, 1985.
[13] A. Friedman, *Stochastic Differential Equations and Applications*, Dover Publication, 2006.
[14] J.A. Given, Chi-Ok Hwang, and M. Mascagni, *First-and last-passage Monte Carlo algorithms for the charge density distribution on a conducting surface*, Physical Review E, 66, 056704, 2002.
[15] (Elton) P. Hsu, *Reflecting Brownian Motion, Boundary Local Time and the Neumann Problem*, Dissertation Abstracts International Part B: Science and Engineering [DISS. ABST. INT. PT. B- SCI. ENG.], 45(6), 1984.
[16] G.J. Jiang and J.L. Knight, *A nonparametric approach to the estimation of diffusion processes, with an application to a short-term interest rate model*, Econometric Theory, 13(05):615–645, 1997.
[17] I. Karatzas and S.E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer-Verlag New York Inc., 1988.
[18] F.C. Klebaner, *Introduction to Stochastic Calculus with Applications*, Imperial College Press, 2001.
[19] P. Lévy, *Processus Stochastiques et Mouvement Brownien, Gauthier-Villars*, Paris, 1948.
[20] P.L. Lions and A.S. Sznitman, *Stochastic differential equations with reflecting boundary conditions*, Comm. Pure Appl. Math., 37:511–537, 1984.
[21] S. Maire and E. Tanré, *Monte Carlo approximations of the Neumann problem*, Monte Carlo Methods and Applications, 19(3):201–236, 2013.
[22] J.P. Morillon, *Numerical solutions of linear mixed boundary value problems using stochastic representations*, Int. J. Numer. Meth. Engng., 40:387–405, 1997.
[23] P. Mörters and Y. Peres, *Brownian Motion*, Cambridge University Press, 30, 2010.
[24] M.E. Müller, *Some continuous Monte Carlo methods for the Dirichlet problem*, The Annals of Mathematical Statistics, 27(3):569–589, 1956.
[25] K.K. Sabelfeld and N.A. Simonov, *Random Walks on Boundary for Solving PDEs*, Walter de Gruyter, 1994.
[26] Y. Saisho, *Stochastic differential equations for multidimensional domain with reflecting boundary*, Probab. Theory Related Fields, 74:455–477, 1987.
[27] J.E. Souza de Cursi, *Numerical methods for linear boundary value problems based on Feynman–Kac representations*, Mathematics and Computers in Simulation, 36(1):1–16, 1994.
[28] C. Yan, W. Cai, and X. Zeng, *A parallel method for solving Laplace equations with Dirichlet data using local boundary integral equations and random walks*, SIAM J. Sci. Comput., 35(4):B868–B889, 2013.