Boundary Value Caching for Walk on Spheres

ABSTRACT

Grid-free Monte Carlo methods such as walk on spheres can be used to solve elliptic partial differential equations without mesh generation or global solves. However, such methods independently estimate the solution at every point, and hence do not take advantage of the high spatial regularity of solutions to elliptic problems. We propose a fast caching strategy which first estimates solution values and derivatives at randomly sampled points along the boundary of the domain (or a local region of interest). These cached values then provide cheap, output-sensitive evaluation of the solution (or its gradient) at interior points, via a boundary integral formulation. Unlike classic boundary integral methods, our caching scheme introduces zero statistical bias and does not require a dense global solve. Moreover we can handle imperfect geometry (e.g., with self-intersections) and detailed boundary/source terms without repairing or resampling the boundary representation. Overall, our scheme is similar in spirit to virtual point light methods from photorealistic rendering: it suppresses the typical salt-and- pepper noise characteristic of independent Monte Carlo estimates, while still retaining the many advantages of Monte Carlo solvers: progressive evaluation, trivial parallelization, geometric robustness, etc. We validate our approach using test problems from visual and geometric computing.

1 INTRODUCTION

The walk on spheres (WoS) method solves problems like the Laplace or Poisson equation by aggregating information from repeated random walks [Muller 1956; Sawhney and Crane 2020]. Like Monte Carlo ray tracing—and unlike conventional partial differential equation (PDE) solvers—it does not require a mesh of the problem domain, nor even a high-quality mesh of its boundary. This fact makes WoS valuable for problems in visual and geometric computing, as one can directly use imperfect assets from design or visualization to perform simulation and analysis (Figure 1). However, classic WoS methods estimate the PDE solution pointwise and do not share information between sample points, resulting in highly redundant computation. We propose a simple boundary value caching (BVC) scheme, well- suited for problems like visualization, where the solution must be evaluated densely in space. This scheme is enabled by the recent walk on stars (WoSt) method [Sawhney et al. 2023], which extends WoS to problems with mixed Neumann and Dirichlet boundary conditions. In particular, we consider PDEs of the form

onΩ $^{\partial u}$=h on$\partial\Omega$

∂n

$\Delta u - \sigma u = f$

u=g on $\partial\Omega_D$ (1)

where the boundary of the domain $\Omega \subset \mathbb{R}^n$ is split into a Dirichlet part $\partial\Omega_D$ and Neumann part $\partial\Omega_N$ with prescribed values g and derivatives h respectively. Here Δ is the negative-semidefinite

Lapla- cian, $\sigma \in \mathbb{R}_{\geq 0}$ is a constant, and f is a given source term. At interior points x, the solution to Equation (1) is given by the boundary inte- gral equation (BIE) [Costabel 1987; Hunter and Pullan 2001]

$$u(x) = \iint \frac{\partial G}{\partial n}(x,z)u(z) - G(x,z)\frac{\partial u}{\partial n}(z)\,\mathrm{d}z + G(x,y)f(y)\,\mathrm{d}y, \quad (2)$$

where G is the free-space Green's function for Equation (1), and n is the unit outward normal at the boundary.

To make use of the BIE, one must somehow determine the unknown boundary data: Dirichlet values u on the Neumann boundary $\partial\Omega_N$, and Neumann values $\partial u/\partial n$ on the Dirichlet boundary $\partial\Omega_D$.

Schemes such as the boundary element method (BEM) use a finite- dimensional space of functions on the boundary (e.g., basis functions associated with mesh nodes), and solve a dense, globally-coupled linear system for the best approximation to the true solution.

We take a completely different approach, and instead use random walks to compute the unknown boundary values. In particular, we use WoS(t) to obtain u along $\partial\Omega_N$ and $\partial u/\partial n$ along $\partial\Omega_D$ (Section 3.1). This approach avoids global solves, boundary remeshing, and approximation of the function space; unlike BEM, it also handles the source term f. Moreover, as random walks can be expensive (especially in problems with predominantly Neumann boundaries), we cache these boundary values at a collection of random sample points along $\partial\Omega$. We can then use a Monte Carlo estimate of Equation (2) to cheaply evaluate the solution at any interior point x, without taking any further random walks (Algorithm 1). This scheme is easy to parallelize, and can be computed progressively (e.g., for interactive preview). We can also focus computation on a region of interest by caching points only on the boundary of a small subdomain $R \subset \Omega$ (Section 3.2)—unlike BEM which must always perform a global solve involving the entire boundary $\partial\Omega$.

In practice, we obtain far smoother results across the domain with our method compared to directly using pointwise estimators like WoS or WoSt (Figures 1, 5 and 7). This behavior can be attributed to correlations in the solution estimates at interior evaluation points that use the same boundary and source samples. On the flip side however, error is now more global akin to traditional PDE solvers such as FEM and BEM (Figure 12). Unlike pointwise estimators, we also observe boundary artifacts (Figure 11) as samples are no longer generated in proportion to the singular functions G and $\partial G/\partial n$. We show how to mitigate such artifacts in Section 3.3.