Grid-Free Monte Carlo for PDEs with Spatially Varying Coefficients - Supplemental

ROHAN SAWHNEY*, Carnegie Mellon University, USA
DARIO SEYB*, Dartmouth College, USA
WOJCIECH JAROSZ†, Dartmouth College, USA
KEENAN CRANE†, Carnegie Mellon University, USA

ACM Reference Format:
Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-Free Monte Carlo for PDEs with Spatially Varying Coefficients - Supplemental. *ACM Trans. Graph.* 41, 4, Article 53 (July 2022), 3 pages. https://doi.org/10.1145/3528223.3530134

1 GREEN'S FUNCTIONS AND POISSON KERNELS

Here we provide the Green's function $G^\sigma(x, y)$ and Poisson kernel $P^\sigma(x, z)$ for a constant coefficient screened Poisson equation on a ball $B(c)$ in 2D and 3D. These quantities are needed to estimate the integral expression in Eq. 26 we derive for variable coefficient PDEs in the paper. Expressions for $\nabla_x G^{\bar\sigma}(x, y)$ and $\nabla_x P^{\bar\sigma}(x, z)$ are provided as well to estimate the spatial derivative of Eq. 26. We also describe how to draw samples y inside $B(c)$ from a probability density p^B that is proportional to the Green's function. Derivations of G^σ_{2D} and G^σ_{3D} can be found in Duffy [2015].

1.1 Centered Expressions

Assume that the point x lies at the center of a ball $B(c)$ with radius R, and let $r := |y - x|$. Then the Green's function on $B(x)$ in two and three dimensions is given by:

$$G^\sigma_{2D}(x, y) = \frac{1}{2\pi}\underbrace{\left(K_0(r\sqrt{\sigma}) - \frac{K_0(R\sqrt{\sigma})}{I_0(R\sqrt{\sigma})} I_0(r\sqrt{\sigma})\right)}_{Q^\sigma_{2D}(r)}, \tag{1}$$

$$\begin{aligned}G^\sigma_{3D}(x, y) &= \frac{1}{4\pi}\sqrt{\frac{2\sqrt{\sigma}}{\pi r}}\left(K_{\frac12}(r\sqrt{\sigma}) - \frac{K_{\frac12}(R\sqrt{\sigma})}{I_{\frac12}(R\sqrt{\sigma})} I_{\frac12}(r\sqrt{\sigma})\right)\\ &= \frac{1}{4\pi}\underbrace{\left(\frac{e^{-r\sqrt{\sigma}}}{r} - \frac{e^{-R\sqrt{\sigma}}}{R}\left(\frac{\sinh(r\sqrt{\sigma})}{r\sqrt{\sigma}}\frac{R\sqrt{\sigma}}{\sinh(R\sqrt{\sigma})}\right)\right)}_{Q^\sigma_{3D}(r)},\end{aligned}$$

where $I_n, I_{n+\frac12}$ and $K_n, K_{n+\frac12}$ (for $n = 0, 1, 2, \ldots$) denote *modified Bessel functions* of the first and second kind (resp.). Routines to efficiently evaluate these functions are available in numerical libraries such as *Boost* [Schäling 2014] and *SciPy* [Virtanen et al. 2019].

To compute the probability density $p^B(x, y) := G^\sigma(x, y)/|G^\sigma(x)|$ associated with these Green's functions, we need to evaluate the integrated value of G^σ over all y on $B(x)$:

$$\begin{aligned}|G^\sigma_{2D}(x)| &:= \int_{B(x)} G^\sigma_{2D}(x, y)\,\mathrm{d}y &= \frac{1}{\sigma}\left(1 - \frac{1}{I_0(R\sqrt{\sigma})}\right),\\ |G^\sigma_{3D}(x)| &:= \int_{B(x)} G^\sigma_{3D}(x, y)\,\mathrm{d}y &= \frac{1}{\sigma}\left(1 - \frac{R\sqrt{\sigma}}{\sinh(R\sqrt{\sigma})}\right).\end{aligned} \tag{2}$$

The Poisson kernel is defined as the normal derivative of the Green's function along the boundary, i.e., for any point z on $\partial B(x)$, $P^\sigma(x, z) := \nabla_z G^\sigma(x, z) \cdot \vec{n}(z)$. In two and three dimensions it is given by:

$$\begin{aligned}P^\sigma_{2D}(x, z) &= \frac{1}{2\pi R}\left(\frac{1}{I_0(R\sqrt{\sigma})}\right),\\ P^\sigma_{3D}(x, z) &= \frac{1}{4\pi R^2}\left(\frac{R\sqrt{\sigma}}{\sinh(R\sqrt{\sigma})}\right).\end{aligned} \tag{3}$$

Notice that in both dimensions, the Poisson kernel equals $\frac{1-\sigma|G^\sigma(x)|}{|\partial B(x)|}$. Also note that for a ball $B(x)$ of radius R (in either 2D or 3D)

$$\sigma|G^\sigma(x)| + |P^\sigma(x)| = 1, \tag{4}$$

where $|P^\sigma(x)|$ denotes the integral of the Poisson kernel over $B(x)$. Intuitively, a random walker is either absorbed inside the ball, *or* it escapes through the boundary. We use these facts to develop the delta tracking variant of WoS, in Sec. 5.1 of the main paper.

1.2 Off-centered Expressions

The next-flight variant of WoS from Sec. 5.2 in the paper requires off-centered versions of the Green's function and Poisson kernel. In particular, assume x is an arbitrary point inside $B(c)$, and let $r_- := \min(|x - c|, |y - c|)$ and $r_+ := \max(|x - c|, |y - c|)$. Furthermore, let θ define the angle between the vectors $x - c$ and $y - c$ in 2D or 3D. Then the off-centered Green's function on $B(c)$ is given by the

*and † indicate equal contribution.

Authors' addresses: Rohan Sawhney, rohansawhney@cs.cmu.edu, Carnegie Mellon University, USA; Dario Seyb, dario.r.seyb.gr@dartmouth.edu, Dartmouth College, USA; Wojciech Jarosz, wojciech.k.jarosz@dartmouth.edu, Dartmouth College, USA; Keenan Crane, kmcrane@cs.cmu.edu, Carnegie Mellon University, USA.

0730-0301/2022/7-ART53
https://doi.org/10.1145/3528223.3530134

infinite series:

$$
\begin{aligned}
G^{\sigma}_{2D}(x,y) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \cos(n\theta)\, I_n(r_-\sqrt{\sigma}) \\
\left(K_n(r_+\sqrt{\sigma}) - \frac{K_n(R\sqrt{\sigma})}{I_n(R\sqrt{\sigma})} I_n(r_+\sqrt{\sigma}) \right),
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
G^{\sigma}_{3D}(x,y) = \frac{1}{4\pi} \sum_{n=0}^{\infty} (2n+1)\, P_n(\cos(\theta)) \left(\sqrt{\frac{\pi}{2r_-\sqrt{\sigma}}} I_{n+\frac{1}{2}}(r_-\sqrt{\sigma}) \right) \\
\sqrt{\frac{2\sqrt{\sigma}}{\pi r_+}} \left(K_{n+\frac{1}{2}}(r_+\sqrt{\sigma}) - \frac{K_{n+\frac{1}{2}}(R\sqrt{\sigma})}{I_{n+\frac{1}{2}}(R\sqrt{\sigma})} I_{n+\frac{1}{2}}(r_+\sqrt{\sigma}) \right),
\end{aligned}
$$

where P_n denotes the recursively defined Legendre polynomials. As usual, the Poisson kernel can be computed by evaluating $\nabla_z G^{\sigma}(x,z) \cdot \vec{n}(z)$ on $\partial B(c)$. We recover the expressions for G^{σ} and P^{σ} in Sec. 1.1 when x coincides with the ball center c.

In practice, we observe that 100 to 200 terms are required to accurately approximate these series. To avoid this computational burden, we provide approximations for these off-centered quantities. In particular, let $\vec{u} := x - c$, $\vec{v} := y - c$ and $\vec{w} := y - x$. Then in two and three dimensions we have:

$$
\begin{aligned}
G^{\sigma}_{2D}(x,y) &= \frac{1}{2\pi} \left(Q^{\sigma}_{2D}(|\vec{w}|) - Q^{\sigma}_{2D}\left(\frac{R^2 - \vec{u}\cdot\vec{v}}{R} \right) \right), \\
G^{\sigma}_{3D}(x,y) &= \frac{1}{4\pi} \left(Q^{\sigma}_{3D}(|\vec{w}|) - Q^{\sigma}_{3D}\left(\frac{R^2 - \vec{u}\cdot\vec{v}}{R} \right) \right), \\
P^{\sigma}_{2D}(x,y) &= \frac{1}{2\pi} \left(V^{\sigma}_{2D}(|\vec{w}|) \frac{|\vec{v}|^2 - \vec{u}\cdot\vec{v}}{|\vec{w}||\vec{v}|} + V^{\sigma}_{2D}\left(\frac{R^2 - \vec{u}\cdot\vec{v}}{R} \right) \frac{\vec{u}\cdot\vec{v}}{R|\vec{v}|} \right), \\
P^{\sigma}_{3D}(x,y) &= \frac{1}{4\pi} \left(V^{\sigma}_{3D}(|\vec{w}|) \frac{|\vec{v}|^2 - \vec{u}\cdot\vec{v}}{|\vec{w}||\vec{v}|} + V^{\sigma}_{3D}\left(\frac{R^2 - \vec{u}\cdot\vec{v}}{R} \right) \frac{\vec{u}\cdot\vec{v}}{R|\vec{v}|} \right),
\end{aligned} \tag{6}
$$

where

$$
\begin{aligned}
V^{\sigma}_{2D}(r) &:= \sqrt{\sigma} \left(K_1(r\sqrt{\sigma}) + \frac{K_0(R\sqrt{\sigma})}{I_0(R\sqrt{\sigma})} I_1(r\sqrt{\sigma}) \right), \\
V^{\sigma}_{3D}(r) &:= \sqrt{\sigma} \sqrt{\frac{2\sqrt{\sigma}}{\pi r}} \left(K_{\frac{3}{2}}(r\sqrt{\sigma}) + \frac{K_{\frac{1}{2}}(R\sqrt{\sigma})}{I_{\frac{1}{2}}(R\sqrt{\sigma})} I_{\frac{3}{2}}(r\sqrt{\sigma}) \right) \\
&= \frac{\sqrt{\sigma}}{r} \left(e^{-r\sqrt{\sigma}} \left(1 + \frac{1}{r\sqrt{\sigma}} \right) + \right. \\
&\qquad \left. \frac{e^{-R\sqrt{\sigma}}}{\sinh(R\sqrt{\sigma})} \left(\cosh(r\sqrt{\sigma}) - \frac{\sinh(r\sqrt{\sigma})}{r\sqrt{\sigma}} \right) \right).
\end{aligned} \tag{7}
$$

These expressions for G^{σ} and P^{σ} are exact when x lies at the center of $B(c)$, but begin to diverge slightly from the true values as x is moved closer to $\partial B(c)$ and the value of coefficient σ is decreased; see Fig. 1. In our experiments, we observe that these approximate expressions provide sufficiently accurate results with the next-flight variant of WoS with far less compute, especially when the value of σ if large.

1.3 Gradient Expressions

In Sec. 2 of this document, we provide an integral expression for the gradient of a PDE solution at a point x. To estimate the gradient at the center of $B(x)$, we need to evaluate the gradients of the Green's

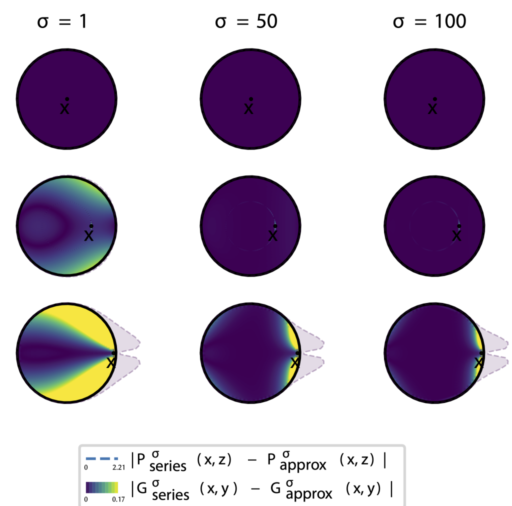

Fig. 1. *First Row*: The series and approximate expressions for the Green's function and Poisson kernel on a ball $B(c)$ from Sec. 1.2 match exactly when x lies at the center of the ball. *Remaining Rows*: The approximate expressions begin to diverge slightly as x is moved closer to $\partial B(c)$, and the value of σ is decreased.

function and Poisson kernel. In two and three dimensions they are given by:

$$
\begin{aligned}
\nabla_x G^{\sigma}_{2D}(x,y) &= \frac{(y-x)\sqrt{\sigma}}{2\pi r} \left(K_1(r\sqrt{\sigma}) - \frac{K_1(R\sqrt{\sigma})}{I_1(R\sqrt{\sigma})} I_1(r\sqrt{\sigma}) \right), \\
\nabla_x G^{\sigma}_{3D}(x,y) &= \frac{(y-x)\sqrt{\sigma}}{4\pi r^2} \left(e^{-r\sqrt{\sigma}} \left(1 + \frac{1}{r\sqrt{\sigma}} \right) - \right. \\
&\qquad \left. \left(\cosh(r\sqrt{\sigma}) - \frac{\sinh(r\sqrt{\sigma})}{r\sqrt{\sigma}} \right) \left(\frac{e^{-R\sqrt{\sigma}} \left(1 + \frac{1}{R\sqrt{\sigma}} \right)}{\cosh(R\sqrt{\sigma}) - \frac{\sinh(R\sqrt{\sigma})}{R\sqrt{\sigma}}} \right) \right), \\
\nabla_x P^{\sigma}_{2D}(x,z) &= \frac{(z-x)\sigma}{2\pi R} \left(\frac{1}{R\sqrt{\sigma}\, I_1(R\sqrt{\sigma})} \right), \\
\nabla_x P^{\sigma}_{3D}(x,z) &= \frac{(z-x)\sigma}{4\pi R^2} \left(\frac{1}{\cosh(R\sqrt{\sigma}) - \frac{\sinh(R\sqrt{\sigma})}{R\sqrt{\sigma}}} \right).
\end{aligned}
$$

1.4 Sampling

To sample from the probability density $p^B := G^{\sigma}(x,y)/|G^{\sigma}(x)|$ associated with the *centered* Green's functions G^{σ} in Sec. 1.1, we first pick a direction $\vec{y}$ uniformly on the unit sphere [Arvo 2001]. A radius r is then sampled from the distribution $2\pi r p^B$ in 2D, or $4\pi r^2 p^B$ in 3D, using rejection sampling. The extra factor in front of p^B accounts for the change of measure between polar and Cartesian coordinates. For rejection sampling, we bound the radial density by the following case dependent function:

$$
h(R,\sigma) := \begin{cases} \max(2.2 * \max(1/R, 1/\sigma), 0.6 * \max(\sqrt{R}, \sqrt{\sigma})) & R \le \sigma, \\ \max(2.2 * \min(1/R, 1/\sigma), 0.6 * \min(\sqrt{R}, \sqrt{\sigma})) & \text{otherwise.} \end{cases} \tag{8}
$$

The final sample point is given by $y = r\vec{y} + x$.

ALGORITHM 1: DeltaTrackingEstimate(x)

```
Input: A point x ∈ Ω.
Output: A single sample estimate û(x) of the solution to Eq. 1.
/* Compute distance and closest point to x on ∂Ω */
d, x̄ ← DistanceToBoundary(x);
/* Return boundary value g at x̄ if x ∈ ∂Ω_ε */
if d < ε then return g(x̄) ;
/* Estimate source contribution at random point y ∈ B(x) */
y ~ G^σ̄(x,y) / |G^σ̄(x)|;
Ŝ ← |G^σ̄(x)| / sqrt(α(x)α(y)) f(y);
/* Decide whether to sample volume or boundary term in Eq. 27 */
if μ ~ U ≤ σ̄|G^σ̄(x)| then
    /* Estimate solution at y ∈ B(x); adjust estimate by
       null-event contribution from Eq. 21 */
    return sqrt(α(y)/α(x)) (1 − σ'(y)/σ̄) DeltaTrackingEstimate(y) + Ŝ;
else
    /* Estimate solution at random point z ∈ ∂B(x) */
    z ~ 1/|∂B(x)|;
    return sqrt(α(z)/α(x)) DeltaTrackingEstimate(z) + Ŝ;
end
```

ALGORITHM 2: NextFlightEstimate(x)

```
Input: A point x ∈ Ω.
Output: A single sample estimate û(x) of the solution to Eq. 1.
/* Compute distance and closest point to x on ∂Ω */
d, x̄ ← DistanceToBoundary(x);
/* Return boundary value g at x̄ if x ∈ ∂Ω_ε */
if d < ε then return g(x̄) ;
/* Initialize series expressions from Eq. 29 */
T̂ ← 0;
Ŝ ← 0;
/* Initialize path throughput */
W ← 1;
/* Sample random exit point z ∈ ∂B(x) used across all
   entries in T̂ */
z ~ 1/|∂B(x)|;
/* Initialize temporary variable to track current
   sample point inside B(x) */
x_c ← x;
while True do
    /* Accumulate boundary contribution */
    T̂ += P^σ̄(x_c,z) / p^∂B(z) W;
    /* Use path throughput as Russian Roulette
       probability to terminate loop */
    P^RR ← min(1, W);
    if P^RR < μ ~ U then break ;
    W /= P^RR;
    /* Sample next random point x_n ∈ B(x) */
    x_n ~ 1/|B(x)|;
    /* Update path throughput */
    W *= G^σ̄(x_c,x_n)(σ̄ − σ'(x_n)) / p^B(x_n);
    /* Accumulate source contribution */
    S += f(x_n) / sqrt(α(x_n)(σ̄ − σ'(x_n))) W;
    /* Update current sample point inside B(x) */
    x_c ← x_n;
end
/* Estimate solution at z ∈ ∂B(x) */
return 1/sqrt(α(x)) (sqrt(α(z)) T̂ NextFlightEstimate(z) + Ŝ);
```

Generating samples from an *off-centered* Green's function $G^{\sigma}(x, y)$ in Sec. 1.2 is more challenging since we do not know of a closed-form expression for $|G^{\sigma}(x)|$. While using a uniform density $\frac{1}{|B(c)|}$ for p^B suffices for unbiased sampling, more sophisticated techniques to generate samples according to the profile of G^{σ} exist. We recommend the weighted reservoir version of resampled importance sampling provided in [Bitterli et al. 2020, Alg. 3].

2 SPATIAL GRADIENT

Applications often require computing not just the solution to a PDE, but the spatial gradient of the solution as well. Fortunately, estimating the gradient $\nabla_x u(x)$ of Eq. 26 from the paper at a point x adds virtually no cost on top of estimating the solution $u(x)$ itself. In particular, either of our WoS algorithms can be used to evaluate the following integral expression for $\nabla_x u(x)$ in a ball $B(x)$:

$$\nabla_x u(x) = \frac{1}{\sqrt{\alpha(x)}} \left(\int_{B(x)} f'(y, \sqrt{\alpha}\, u)\ \nabla_x G^{\bar{\sigma}}(x, y)\, \mathrm{d}y \ + \int_{\partial B(x)} \sqrt{\alpha(z)}\, u(z)\ \nabla_x P^{\bar{\sigma}}(x, z)\, \mathrm{d}z \right) - \frac{u(x)}{2\alpha(x)} \nabla_x \alpha(x). \quad (9)$$

The value of $\nabla_x u(x)$ only needs to be estimated in the first ball in any walk—the solution estimates it depends on can be computed recursively using the delta tracking or next-flight estimators; see Sec. 5 in the paper. Furthermore, the parameters $\bar{\sigma}$, p^B, $p^{\partial B}$, $\mathbb{P}^B$ and $\mathbb{P}^{\partial B}$ remain unchanged with either algorithm.

3 PSEUDO-CODE

Here we provide pseudo-code for the two variants of walk on spheres presented in Sec. 5 of the paper. To maintain consistency with the paper, we assume the drift coefficient $\vec{\omega}(x) = \vec{0}$ over the entire domain.

REFERENCES

J. Arvo. 2001. Stratified sampling of 2-manifolds. *SIGGRAPH Course Notes* 29, 2 (2001).

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal Reservoir Resampling for Real-Time Ray Tracing with Dynamic Direct Lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020).

Dean G Duffy. 2015. *Green's functions with applications.* Chapman and Hall/CRC.

B. Schäling. 2014. *The boost C++ libraries.* XML Press.

P. Virtanen, R. Gommers, and Contributors. 2019. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, Article arXiv:1907.10121 (Jul 2019), arXiv:1907.10121 pages. arXiv:1907.10121

Received January 2022