

Borrador Informe

Alumno: Pendiente

Profesor: Pendiente

1. Teoría

Presentamos a continuación las ecuaciones de Laplace y de Poisson, junto con algunas propiedades que poseen sus soluciones, las cuales serán relevantes para la introducción del algoritmo de caminata en esferas [JF: Modificar esto si es que agregamos otros problemas, como el Screened Poisson.](#)

Dado un conjunto abierto $\Omega \subseteq \mathbb{R}^d$ y funciones $f : \Omega \rightarrow \mathbb{R}$ y $g : \partial\Omega \rightarrow \mathbb{R}$, se define

1. La ecuación de Laplace como

$$\begin{aligned}\Delta u &= 0 \text{ en } \Omega \\ u &= g \text{ en } \partial\Omega.\end{aligned}\tag{1}$$

2. La ecuación de Poisson como

$$\begin{aligned}-\Delta u &= f \text{ en } \Omega \\ u &= g \text{ en } \partial\Omega.\end{aligned}\tag{2}$$

La solución a la ecuación (1) es armónica en Ω , por ende, satisface la siguiente representación integral, la cual es conocida como *propiedad del valor medio*.

Proposición 1 (Propiedad del valor medio, (Axler et al., 2013)). Sean $x_0 \in \Omega$, $R > 0$ y $B(x_0, R) \subseteq \Omega$. Si u es una solución de (1), entonces

$$u(x_0) = \frac{1}{|\partial B(x_0, R)|} \int_{\partial B(x_0, R)} u(y) dy.$$

Dicho de otra manera, si $X \sim \mathcal{U}(\partial B(x_0, R))$ es una variable aleatoria con distribución uniforme sobre la esfera de centro x_0 y radio $R > 0$, entonces

$$u(x_0) = \mathbb{E}[u(X)].$$

La solución de la ecuación (2) admite una representación integral similar.

Proposición 2 (Zachmanoglou and Thoe (1986), Ejercicio 9.4, Capítulo 7). Sean $x_0 \in \Omega$, $R > 0$ y $B(x_0, R) \subseteq \Omega$. Si u es una solución de (2), entonces

$$u(x_0) = \frac{1}{|\partial B(x_0, R)|} \int_{\partial B(x_0, R)} u(y) dy + \int_{B(x_0, R)} f(y) G(x_0, y) dy,$$

donde $G(x_0, y)$ es la función de Green en la esfera $B(x_0, R)$.

Parafraseado en términos probabilísticos, si $X \sim \mathcal{U}(\partial B(x_0, R))$ e $Y \sim \mathcal{U}(B(x_0, R))$, entonces

$$u(x_0) = \mathbb{E}[u(X)] + |B(x_0, R)| \cdot \mathbb{E}[f(Y)G(x_0, Y)].$$

Para instancias de la función de Green de la esfera para los casos $d = 2$ y $d = 3$, ver Tabla 1.

	$2D$	$3D$
$G(x, y)$	$\frac{1}{2\pi} \ln(R / \ y - x\ _2)$	$\frac{1}{4\pi} \left(\frac{R - \ y - x\ _2}{R \ y - x\ _2} \right)$
$\nabla_x G(x, y)$	$\frac{y - x}{2\pi} \left(\frac{1}{\ y - x\ _2^2} - \frac{1}{R^2} \right)$	$\frac{y - x}{4\pi} \left(\frac{1}{\ y - x\ _2^3} - \frac{1}{R^3} \right)$

Tabla 1: Funciones de Green y sus gradientes (con respecto a x) de la bola Euclidiana de radio R para los casos $d = 2$ y $d = 3$.

2. Walk on Spheres

2.1. Derivación del algoritmo

La idea principal del algoritmo de *Caminata en Esferas* es reemplazar las representaciones integrales de las Proposiciones 1 y 2 por simulaciones de Montecarlo. Específicamente,

1. En el contexto de la ecuación de Laplace (1), muestrear $X_1 \sim \mathcal{U}(\partial B(x_0, R))$ y aproximar $u(x_0)$ mediante:

$$u(x_0) \approx u(X_1).$$

2. En el contexto de la ecuación de Poisson (2), muestrear $X_1 \sim \mathcal{U}(\partial B(x_0, R))$ e $Y_1 \sim \mathcal{U}(B(x_0, R))$ y aproximar $u(x_0)$ mediante:

$$u(x_0) \approx u(X_1) + |B(x_0, R)| \cdot f(Y_1)G(x_0, Y_1).$$

Observación: En la ecuación de Poisson, el término $f(Y_1)G(x_0, Y_1)$ es fácil de calcular, pues tanto f como G son funciones conocidas; el término $u(X_1)$, en cambio, presenta dificultades, ya que lo que buscamos es precisamente conocer u (naturalmente, esto también es válido para la ecuación de Laplace). **JF:** Por algún motivo overleaf deja de compilar si defino un comando para la observación.

Desde una perspectiva algorítmica, hay dos preguntas naturales respecto a la anterior propuesta de aproximación: ¿Cómo estimamos $u(X_1)$? y ¿Cómo elegimos el radio de la bola?

Debido a que ambas ecuaciones necesitan calcular $u(X_1)$ y la ecuación de Laplace es más sencilla, usaremos esa aproximación de modelo; es decir, estaremos considerando $u(x_0) \approx u(X_1)$.

Observemos que si $X_1 \in \partial\Omega$, entonces $u(X_1) = g(X_1)$, valor que conocemos por el *setting* del problema. Así que la dificultad se presenta cuando $X_1 \notin \partial\Omega$. Si $X_1 \notin \partial\Omega$, entonces podemos aplicar la aproximación estocástica, pero esta vez a $u(X_1)$; es decir, tomar una bola $B(X_2, R) \subset \Omega$, muestrear $X_2 \sim \mathcal{U}(\partial B(X_1, R_2))$ y estimar $u(X_1) \approx u(X_2)$. Repetiremos este proceso hasta que –de manera fortuita– podamos muestrear una variable aleatoria que se encuentre en $\partial\Omega$. Una elección sensata para el radio, $R > 0$,

de las bolas es que este sea maximal en el sentido de que $B(x_0, R)$ sea la mayor bola centrada en x_0 que está dentro Ω , pues así «aumentamos» nuestras chances de caer en $\partial\Omega$. Se obtiene así una primera versión (ingenua) del algoritmo (ver Algoritmo 1).

Algoritmo 1: Caminata en esferas (ingenuo)

Input : Punto inicial $x_0 \in \Omega$, función $g : \partial\Omega \rightarrow \mathbb{R}$
 $X_{-1} \leftarrow x_0$;
while $X_{-1} \notin \partial\Omega$ **do**
 Calcular la distancia, R , de X_{-1} a $\partial\Omega$;
 Muestrear $X \sim \mathcal{U}(\partial B(X_{-1}, R))$;
 $X_{-1} \leftarrow X$
end
Output: $g(X_{-1})$

El gran problema de este algoritmo es que de manera casi segura no sale de su bucle computacional. Más precisamente, si definimos

$$T^* := \text{Iteraciones que toma realiza el algoritmo 1 en entregar un resultado}, \quad (3)$$

entonces

$$\mathbb{P}(T^* = +\infty) = 1.$$

Esto es intuitivo si consideramos dominios poliedrales, donde una esfera contenida solo puede intersectar una cantidad finita de puntos en la frontera y que, por ende, tiene probabilidad 0 de caer en uno de ellos¹.

JF: Agregar dibujo con un rectángulo y una esfera tocando algunos puntos de la frontera.

Para subsanar este problema, se utiliza como criterio de parada la pertenencia a un cascarón, $\partial\Omega_\varepsilon$, de diámetro ε a la frontera –en vez de la pertenencia exacta a $\partial\Omega$. Formalmente, la pertenencia a

$$\partial\Omega_\varepsilon := \{x \in \bar{\Omega} : \text{dist}(x, \partial\Omega) < \varepsilon\}. \quad (4)$$

Si bien en la discusión anterior nos enfocamos en el problema de Laplace (1), esta continúa siendo válida para el problema de Poisson (2). Esto debido a que la detención del algoritmo de caminata en esferas para el problema de Poisson depende de exactamente la misma condición: ser capaces de estimar $u(X)$, que en última instancia depende de que $X \in \partial\Omega$ (o $\partial\Omega_\varepsilon$).

Se presenta a continuación el algoritmo de caminata en esferas (algoritmo 2) para el problema de Poisson. Para mitigar el efecto de la estocasticidad, se agrega un bucle extra, que permite realizar N estimaciones independientes de $u(x_0)$, para luego promediarlas.

¹Esto sigue siendo cierto si consideramos como Ω una bola abierta y a x_0 como cualquier punto distinto del centro.

Algoritmo 2: Caminata en esferas

Input : Punto inicial $x_0 \in \Omega$, parámetro $\varepsilon > 0$, cantidad de estimaciones $N \in \mathbb{N}$, funciones $g : \partial\Omega_\varepsilon \rightarrow \mathbb{R}$ y $f : \Omega \rightarrow \mathbb{R}$

for $n = 1, \dots, N$ **do**

$X_{-1} \leftarrow x_0$;

$u_n \leftarrow 0$;

while $X_{-1} \notin \partial\Omega_\varepsilon$ **do**

 Calcular la distancia, R , de X_{-1} a $\partial\Omega$;

 Muestrear $Y \sim \mathcal{U}(B(X, R))$;

$u_n \leftarrow u_n + |B(X_{-1}, R)| \cdot f(Y)G(X_{-1}, Y)$;

 Muestrear $X \sim \mathcal{U}(\partial B(X_{-1}, R))$;

$X_{-1} \leftarrow X$

end

$u_n \leftarrow u_n + g(X_{-1})$

end

Output: $S_N = \frac{1}{N} \sum_{n=1}^N u_n$

En caso de que queramos aproximar el problema de Laplace, simplemente ignoramos el muestreo de Y y asignamos $f \equiv 0$.

2.2. Garantías teóricas de la caminata en esferas

Presentamos a continuación garantías en esperanza del tiempo de ejecución y de convergencia del Algoritmo 2.

Teorema 3. (DeLaurentis and Romero, 1990) Sea $\Omega \subseteq \mathbb{R}^d$ un abierto tal que $\bar{\Omega}$ es una unión de convexos compactos. Sea $0 < \varepsilon < 1$, $x_0 \in \Omega \setminus \partial\Omega_\varepsilon$ el punto donde queremos estimar la solución de la ecuación y T^* definido como en (3). Luego

$$\mathbb{E}[T^*] = O(\ln(1/\varepsilon)).$$

Teorema 4. (DeLaurentis and Romero, 1990) Bajo las mismas hipótesis del Teorema 3 y asumiendo adicionalmente que $|\mathbb{E}[u(X_{T^*})] - \mathbb{E}[f(X_{T^*})]| = O(\varepsilon^{1/p})$, para algún $p > 0$, y que $\varepsilon = N^{-p/2}$, se tiene que

$$\mathbb{E}[(S_N - u(x_0))^2] = O(1/N).$$

En consecuencia, bajo las hipótesis de este teorema, si queremos obtener una aproximación con error del orden de $O(1/N)$, la cantidad de iteraciones esperadas es

$$\mathbb{E}[\text{RunTimeWoS}] = O(N \ln(N)).$$

JF: Creo que sería bueno agregar la cantidad de Queries que necesitamos (lo que depende del Bounding Volume Hierarchy).

JF: Aún queda añadir

- Describir más detalladamente las ecuaciones diferenciales, dar motivación (posiblemente física) de ellas.

- Detallar las condiciones que deben imponerse a la ecuación para obtener existencia y unicidad de soluciones (me parece que la unicidad para Poisson es relativamente sencilla).
- Hablar de la estabilidad del algoritmo. Me parece que no tiene mucho sentido hacer un estudio de estabilidad como el que vimos en clases, pero creo que hay que elaborar un poco en por qué esto es así.
- Resultados de los experimentos.
- Conclusiones
- Intro (creo que es mejor patearla para el final).

Referencias

- AXLER, S., BOURDON, P. and WADE, R. (2013). *Harmonic function theory*, vol. 137. Springer Science & Business Media.
- DELAURENTIS, J. M. and ROMERO, L. A. (1990). A monte carlo method for poisson's equation. *Journal of Computational Physics*, **90** 123–140.
- ZACHMANOGLU, E. C. and THOE, D. W. (1986). *Introduction to partial differential equations with applications*. Courier Corporation.