

# Estudio numérico de la ecuación Eikonal mediante los métodos de fast marching y fast sweeping

**Presenta:** Axel Álvarez, Axel Kolm y Manuel Torres V.  
**Profesor:** Axel Osses.  
**Auxiliar:** Emir Chacra.  
**Curso:** Análisis numérico de EDPs:  
Teoría y Laboratorio (MA4307).



Ingeniería Matemática  
FACULTAD DE CIENCIAS  
FÍSICAS Y MATEMÁTICAS  
UNIVERSIDAD DE CHILE

- 1 Introducción
- 2 Diferencias finitas
- 3 Fast sweeping
- 4 Fast marching
- 5 Bibliografía y discusión

# Motivación

Qué es la ecuación Eikonal?

Consideramos la ecuación

$$|\nabla f(x)|^2 = \frac{1}{v^2(x)} \quad x \in \Omega.$$

conocida como la *ecuación Eikonal*. La *ecuación Eikonal* es instrumental en muchas aplicaciones en varios

campos que van desde la visión por computadora hasta la geociencia. Esta ecuación se puede resolver con buenos resultados usando:

- 1 Fast sweeping.
- 2 Fast marching.

- 1 Considerando el caso en que  $v^2(x, y) = \frac{1}{4(x^2 + y^2)}$  se estudia la ecuación diferencial

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 = 4(x^2 + y^2), \quad \text{en } [0, 1]^2,$$

con *condiciones de borde Dirichlet*  $u = 0$  en  $x_0 = (0, 0)$ . La solución exacta es

$$u(x, y) = x^2 + y^2.$$

- 2 Considerando el caso en que  $\frac{1}{v^2(x, y)} = \frac{4(x^2 + y^2)}{(1 + x^2 + y^2)^2}$  se estudia la ecuación diferencial

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 = \frac{4(x^2 + y^2)}{(1 + x^2 + y^2)^2}, \quad \text{en } [0, 1]^2,$$

con *condiciones de borde Dirichlet*  $u = 0$  en  $x_0 = (0, 0)$ . La solución exacta es

$$u(x, y) = \ln(1 + x^2 + y^2).$$

- 1 Introducción
- 2 Diferencias finitas
- 3 Fast sweeping
- 4 Fast marching
- 5 Bibliografía y discusión

Recordar la ecuación de Eikonal

$$|\nabla T(x, y)| = \frac{1}{c^2(x, y)} = s^2(x, y) \quad (1)$$

En [5] proponen un esquema basado en diferencias finitas para resolver esta EDP. Supongamos que el sismo empieza en el punto A de la figura 1 con  $T_A = 0$ , luego se tiene lo siguiente

$$T_i = \frac{h}{2}(s_{B_i} + s_A)$$

donde  $T_i$  corresponde al valor de  $T$  en  $B_i$ .

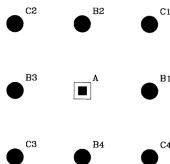


Figura 1: Grilla

Luego para conocer los otros 4 valores restantes se utiliza diferencias finitas con  $T_0, T_1, T_2, T_3$  los valores de  $T$  en  $A, B_1, B_2, C_1$  respectivamente.

$$\begin{aligned}\frac{\partial T}{\partial x} &= \frac{1}{2h}(T_0 + T_2 - T_1 - T_3) \\ \frac{\partial T}{\partial y} &= \frac{1}{2h}(T_0 + T_1 - T_2 - T_3)\end{aligned}$$

Con lo que sustituyendo esto en (1) se tiene lo siguiente

$$T_3 = T_0 + \sqrt{2(hs)^2 - (T_2 - T_1)^2} \quad (2)$$

Luego se procede iterativamente de manera similar aproximando el anillo fuera de lo recién calculado. Para ello, primero se calcula los puntos de los lados y por último las esquinas, por ejemplo, se empieza calculando el valor de  $T$  en el bloque derecho como se puede observar en la figura 2.

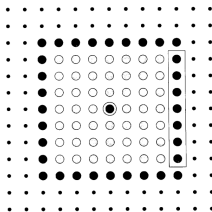


Figura 2: Grilla de la iteración. Los círculos blancos indican los puntos cuyos tiempos fueron calculados.



Para resolver el bloque hay que escoger desde donde se empieza a calcular el tiempo, el cual es el mínimo relativo (un punto se dice mínimo relativo si es que el punto adyacente, es mínimo local en el bloque adyacente del anillo recién resuelto). El tiempo en este punto se calcula usando diferencias finitas.

Luego hay que escoger la dirección en la cual se recorre el bloque, para ello se identifica el máximo relativo y se recorre en dirección a este punto, los tiempos de los puntos en el recorrido se calculan usando (2).

Posteriormente, se recorre la otra dirección y se calculan los tiempos usando (2), incluyendo al mínimo relativo. Notar que se calcula dos veces el valor del mínimo relativo, se escoge el menor tiempo para este punto.

Finalmente, se procede de la misma manera para los otros tres bloques restantes y se calculan los tiempos de las esquinas usando (2).

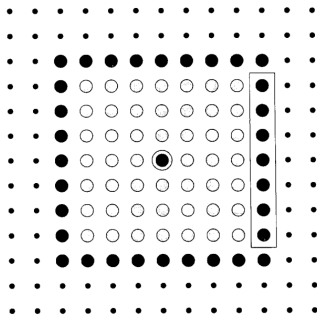


Figura 3: Grilla en la iteración.

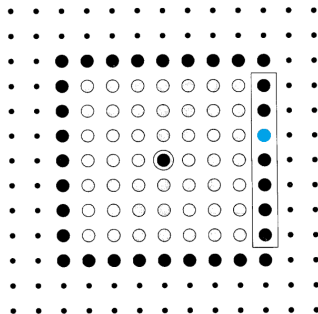


Figura 4: Grilla en la iteración.

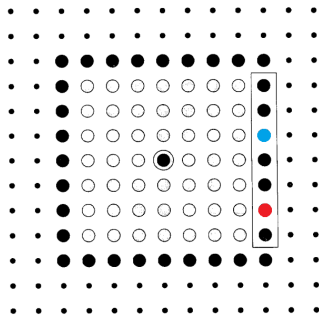


Figura 5: Grilla en la iteración.

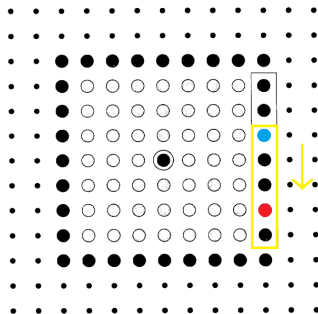


Figura 6: Grilla en la iteración.

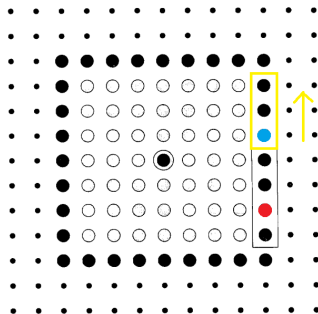


Figura 7: Grilla en la iteración.

Ahora se van a presentar los resultados de la implementación de este método.

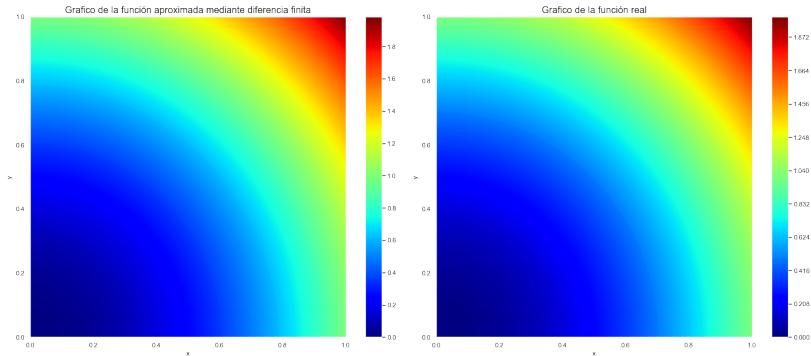


Figura 8: Simulación cuando  $s^2(x, y) = 4(x^2 + y^2)$

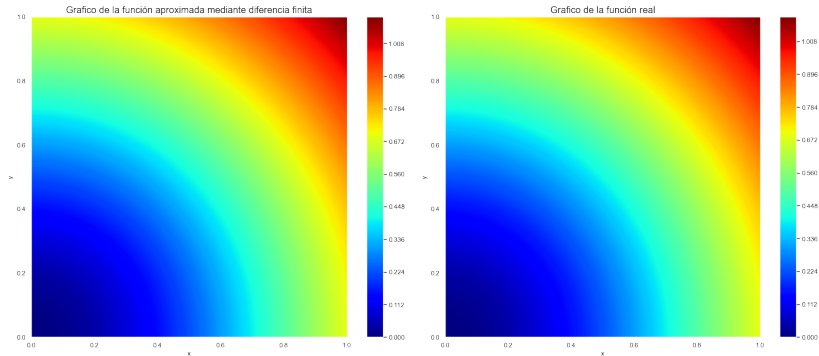


Figura 9: Simulación cuando  $s^2(x, y) = 4(x^2 + y^2)/(1 + x^2 + y^2)$



- 1 Introducción
- 2 Diferencias finitas
- 3 Fast sweeping
- 4 Fast marching
- 5 Bibliografía y discusión

# Fast sweeping

Idea principal

En este esquema se recorre el dominio realizando barridos como en la imagen

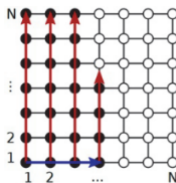


Figura 10: Grilla

Estos se realizan para cada permutación de sentidos de recorrido de ejes del plano cartesiano. Por lo tanto, si estamos en  $\mathbb{R}^n$  se realizan  $2^n$  barridos.

La forma de actualizar los valores de los puntos es dada por el siguiente esquema

$$\left((u_{ij}^h - u_{x\min}^h)^+\right)^2 + \left((u_{ij}^h - u_{y\min}^h)^+\right)^2 = f_{ij}^2 h^2$$
$$i = 2, \dots, I-1, \quad j = 2, \dots, J-1.$$

Se recorre en el siguiente orden para  $\mathbb{R}^2$ :

$$(1) \quad i = 1 : I, j = 1 : J,$$

$$(2) \quad i = I : 1, j = 1 : J,$$

$$(3) \quad i = I : 1, j = J : 1,$$

$$(4) \quad i = 1 : I, j = J : 1.$$

# Fast sweeping

## Algoritmo

- 1 Se inicializan los valores de la frontera según las condiciones iniciales
- 2 Se asigna valor infinito al resto del dominio.
- 3 Los puntos son recorridos según cada barrido .
- 4 Para cada punto  $x = (i, j)$ , calculamos  $u(x)$  según el esquema numérico ya descrito, luego sea  $u_p$  el valor previo asignado a  $x$ , si  $u_p > u(x)$  entonces se asigna este como su nuevo valor.

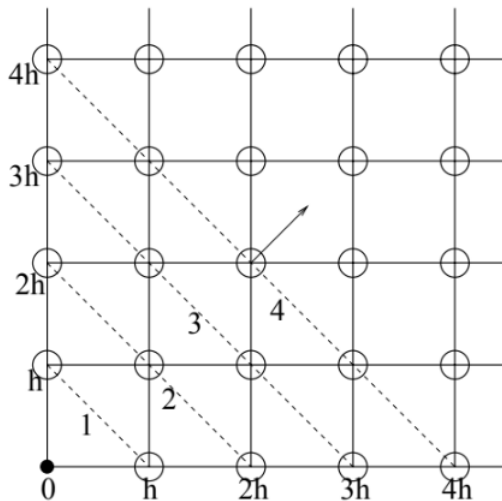
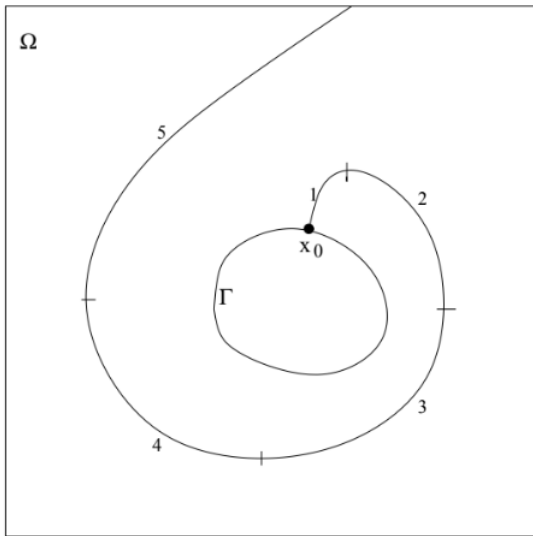


Figura 11: Dependencia temporal de la solución

En el caso heterogéneo, las características podrían no estar solo en un cuadrante



Curvas de nivel de la solución mediante: Fast sweeping method $[[0, 0]]$

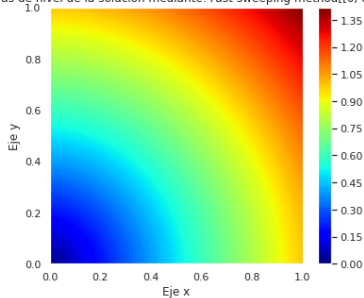


Figura 13: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

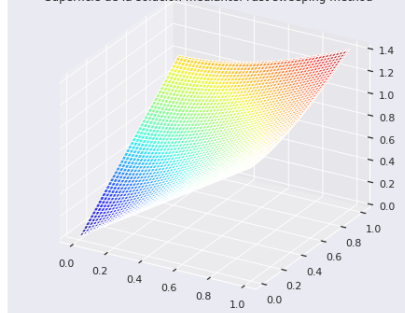


Figura 14: Grafo de la solución

# Fast Sweeping

Resultados para medio homogéneo

Curvas de nivel de la solución mediante: Fast sweeping method[0, 0], [99, 99]]

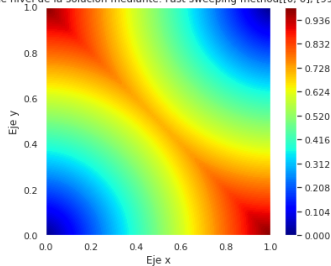


Figura 15: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

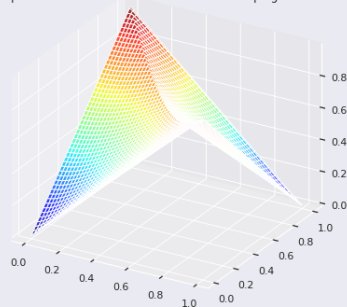


Figura 16: Grafo de la solución



# Fast Sweeping

Resultados para medio homogéneo

Curvas de nivel de la solución mediante: Fast sweeping method[[50.0, 50.0]]

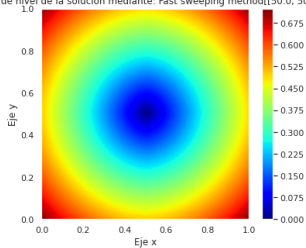


Figura 17: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

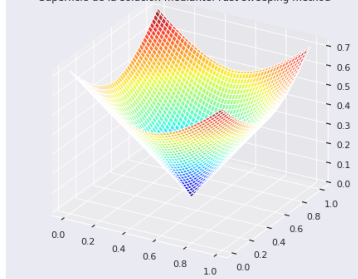


Figura 18: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[[0, 0]]

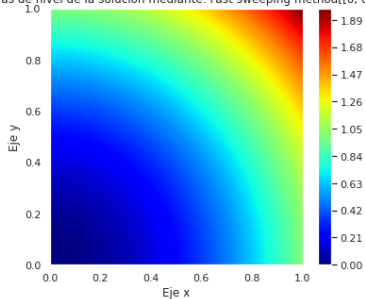


Figura 19: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

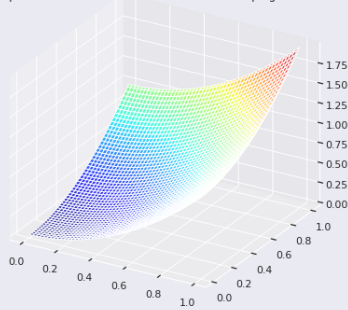


Figura 20: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[[0, 0], [99, 99]]

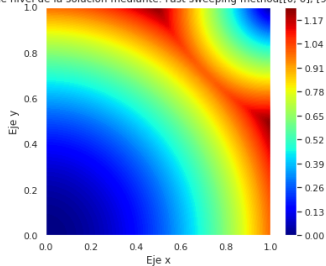


Figura 21: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

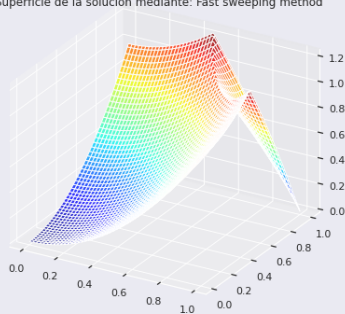


Figura 22: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[[50.0, 50.0]]

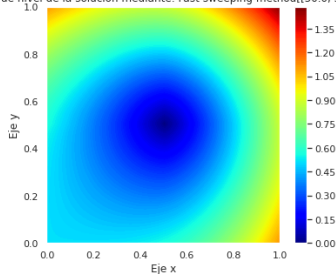


Figura 23: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

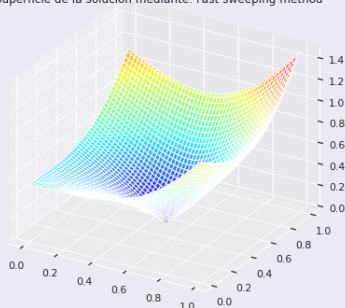


Figura 24: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)/(1 + x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[0, 0]]

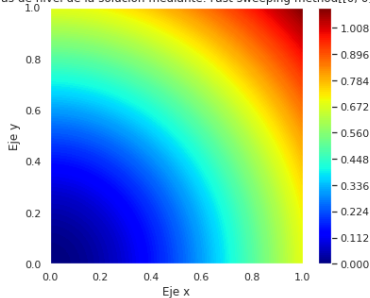


Figura 25: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

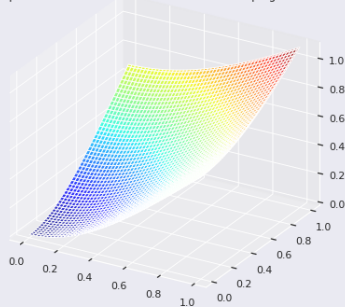


Figura 26: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)/(1 + x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[[0, 0], [99, 99]]

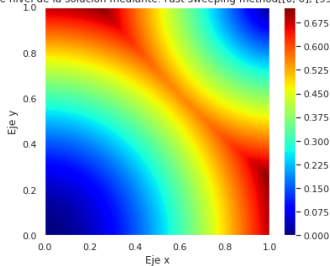


Figura 27: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

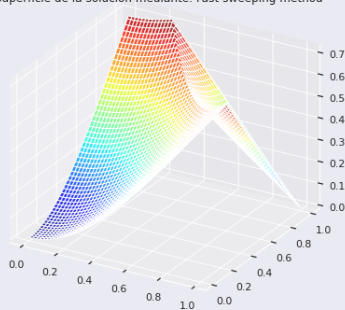


Figura 28: Grafo de la solución

# Fast Sweeping

Resultados para medio con velocidad  $s^2(x, y) = 4(x^2 + y^2)/(1 + x^2 + y^2)$

Curvas de nivel de la solución mediante: Fast sweeping method[[50.0, 50.0]]

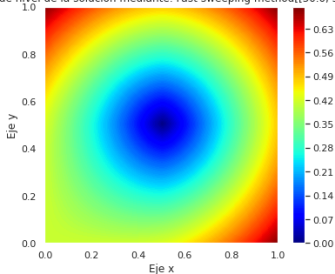


Figura 29: Curvas de nivel de la solución

Superficie de la solución mediante: Fast sweeping method

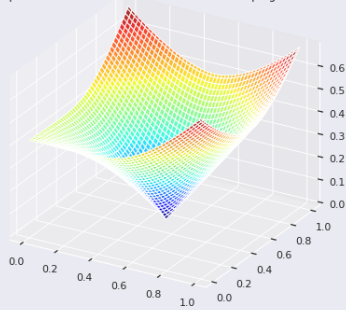


Figura 30: Grafo de la solución

- 1 Introducción
- 2 Diferencias finitas
- 3 Fast sweeping
- 4 Fast marching
- 5 Bibliografía y discusión



Consideramos la ecuación Eikonal evaluada en una cantidad finita de puntos

$$|(\nabla T)_{ijk}|^2 = \frac{1}{v_{ijk}^2},$$

donde denotamos:

$$f_{ijk} := f(i \cdot \Delta x_1, j \cdot \Delta x_2, k \cdot \Delta x_3),$$

Los autores White y et. al. [4] presentan la siguiente discretización para obtener el esquema a utilizar

$$|(\nabla T)_{ijk}| \approx \sum_{a=1,\dots,3} \max \left\{ \frac{1}{h_{\xi_a}} D_{ijk}^{-\xi_a}, -\frac{1}{h_{\xi_a}} D_{ijk}^{+\xi_a}, 0 \right\},$$

donde  $h_{\xi_a}$  representa el factor de escala a lo largo de  $\xi_a$  para el sistema de coordenadas usado, y  $D_{ijk}^{-\xi_a}$  y  $D_{ijk}^{+\xi_a}$ , representan las diferencias finitas *backward* y *forward* en  $ijk$  de  $\nabla T$  respectivamente, a lo largo de  $\xi_a$ . Entonces tomamos

$$\sum_{a=1,\dots,3} \max \left\{ \frac{1}{h_{\xi_a}} D_{ijk}^{-\xi_a}, -\frac{1}{h_{\xi_a}} D_{ijk}^{+\xi_a}, 0 \right\} - \frac{1}{v_{ijk}^2} \approx 0,$$

que es una *diferencia contra el viento estructurada* (la información se propaga en una dirección).

El algoritmo (Sethian [3], White [4]) es:

- 1 Para todo  $ijk$  en  $\Omega$ ,  $T_{ijk} = \infty$ . Guardar  $ijk$  en  $U$ .
- 2 Para todo  $ijk$  en  $\partial\Omega$ ,  $T_{ijk} = 0$ , guardar  $ijk$  en  $V$  (o mover de  $U$  a  $V$ ).
- 3 Mientras exista  $ijk$  en  $V$  hacer:
  - Sea  $\tilde{ijk}$  el índice que minimiza  $T$  en  $V$ . Trasferimos  $\tilde{ijk}$  de  $V$  hacia  $K$ .
  - Para todo  $lmn$  vecindad de  $\tilde{ijk}$ , hacer:
    - $T_{lmn}^*$  = solución de la ecuación  $(*)$ .
    - Si  $T_{lmn}^* > T_{lmn}$  entonces  $T_{lmn} \leftarrow T_{lmn}^*$ .
    - Si  $lmn \in U$  entonces mover  $lmn$  desde  $U$  hacia  $V$ .

$$(*) \quad \sum_{a=1, \dots, 3} \max \left\{ \frac{1}{h_{\xi_a}} D_{ijk}^{-\xi_a}, -\frac{1}{h_{\xi_a}} D_{ijk}^{+\xi_a}, 0 \right\} - \frac{1}{v_{ijk}^2} \approx 0.$$

- 1 Introducción
- 2 Diferencias finitas
- 3 Fast sweeping
- 4 Fast marching
- 5 Bibliografía y discusión

- El método de *fast marching* es más eficiente que el método de *fast sweeping* según Rickett [2].
  - Fast sweeping realiza al menos  $2^m$  iteraciones, donde  $m$  es la dimensión del dominio, y en cada iteración se realizan al menos  $n^m$  operaciones, con  $n$  el tamaño de la grilla.
  - Fast marching tiene cota  $O(n \log(n))$  donde  $n$  determina el refinamiento de la grilla (en el caso de 3 dimensiones)<sup>1</sup>.
- El método de *fast sweeping* permite múltiples emisores, mientras que el método de *fast marching* no lo permite (Rickett [2]).

---

<sup>1</sup>White [4] el 2020 implementa una versión con esta complejidad, en 2013 [1] habla de una implementación con complejidad  $O(n^2 \log(n))$ .



Capozzoli, A., A Comparison of Fast Marching, Fast Sweeping and Fast Iterative Methods for the Solution of the Eikonal Equation (2013).



Rickett, J., Fomel, S., A Second-Order Fast Marching Eikonal Solver (1999).



Sethian, J., A Fast Marching Level Set Method for Monotonically Advancing Fronts (1995).



White, M., Nakata, N., Fang, H., PyKonal: A Python Package for Solving the Eikonal Equation in Spherical and Cartesian Coordinates Using the Fast Marching Method (2020).



Vidale, J., Finite-difference calculation of travel times (1988).



Zhao, H., A Fast Sweeping Method for Eikonal Equations (2004).