Algoritmo

Simulation

September 17, 2024

Contents

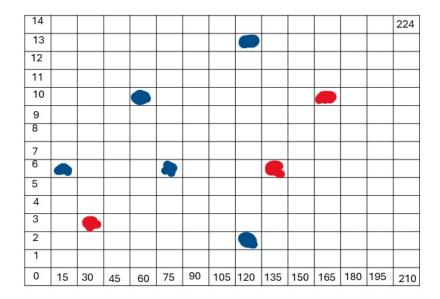
1	Idea del algoritmo.			
	1.1	Ideas	iniciales	1
	1.2	$Ajust\epsilon$	e	2
		1.2.1	Queremos:	3
		1.2.2	¿Y bajo qué criterio los ajusto?	3
		1.2.3	¿Qué obtengo?	3

1 Idea del algoritmo.

1.1 Ideas iniciales.

Tenemos el Lattice-Boltzmann.

Las manches azules, corresponden a las estaciones y las rojas a las casillas fuente. $(15\mathrm{x}15)$



Créditos a Andrés :P

Tenemos:

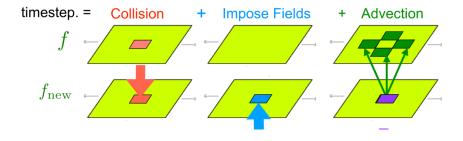
- Densidad en las estaciones.
- Densidad simulada.

Se inicializan, por ahora, pero estos son dos arreglos que entran al programa de Python.

density_grid_simulated = LB_density_with_sources(Lx, Ly, sources_1D, rho_sources, sigms_density_grid_obs = LB_density_with_sources(Lx, Ly, sources_1D, rho_sources_random, signs_sources_1D, rho_sources_random, signs_sources_random, sig

1.2 Ajuste.

 $\rho_S(ImposeFire)$ La densidad simulada depende de lo que ponga en Impose-Fire.



1.2.1 Queremos:

Optimizar los valores de ρ_F para las celdas fuente de incendio. Los ρ_F , ρ en las celdas fuente del incendio, son los parámetro que puedo variar mediante ImposeFire.

1.2.2 ¿Y bajo qué criterio los ajusto?

Se buscan ρ en las celdas fuente de incendio tal que se minimice el error cuadrático. El superíndice es el número de estación.

$$Error_C = (\rho_S^1 - \rho_O^1)^2 + (\rho_S^2 - \rho_O^2)^2 + (\rho_S^3 - rho_O^3)^2 + \dots + (\rho_S^n - \rho_O^n)^2$$

1.2.3 ¿Qué obtengo?

 ρ_F para las fuentes del incendio en un tiempo t
. Iterando, puedo graficar ρ_F v
st