

TESIS MAESTRÍA EN FÍSICA

SIMULACIONES MASIVAMENTE PARALELAS DE MODELOS DE PROPAGACIÓN DE EPIDEMIAS E INCENDIOS

Renzo Zagarra Saez
Maestrando

Dr. Alejandro Kolton
Director

Miembros del Jurado

Dr. Ezequiel Ferrero

Dr. Eduardo Jagla

Dr. Guillermo Abramson

1 de Diciembre de 2022

Grupo de Teoría de la Materia Condensada – Centro Atómico
Bariloche

Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica
Argentina

A mis padres, Francisco y Liliana,
por su apoyo incondicional.

A mis hermanos, Franco y Lucas,
por hacer de cada momento una sonrisa.

A mi novia, Sol,
por acompañarme y entenderme como nadie.

Índice de contenidos

Índice de contenidos	v
Índice de figuras	vii
Índice de tablas	ix
Resumen	xi
Abstract	xiii
1. Estructura de la tesis	1
2. Introducción	3
2.1. interfaces fuera del equilibrio en la naturaleza	3
2.2. Conceptos de escala	7
2.2.1. Ancho de la interfaz	8
2.2.2. Factor de estructura de la interfaz	11
2.3. Universalidad: Poisson, KPZ y EW	13
3. Modelo SIR	17
3.1. Historia	17
3.2. Modelo SIR de campo medio	19
3.3. Modelo SIR de reacción-difusión homogéneo	23
3.3.1. Soluciones de onda	24
3.3.2. Soluciones en sistema de reacción-difusión-convección	27
3.4. Modelo SIR de reacción-difusión-convección heterogéneo	28
3.5. Problema y observables	29
4. Reseña numérica y computacional	35
4.1. Diferencias finitas	36
4.2. Implementaciones en <i>Python</i>	37
4.2.1. Implementación con <i>NumPy</i>	38
4.2.2. Implementación serial con <i>Numba</i>	39

4.2.3. Implementación paralela con <i>Numba</i>	41
4.2.4. Implementación con <i>CuPy</i>	42
4.3. Tamaño del sistema	45
5. Fenómenos críticos del frente de infección	47
5.1. Medio homogéneo (H)	48
5.2. Medios heterogéneos	50
5.2.1. DA vs H	50
5.2.2. Medios correlacionados	55
6. Geometría del frente de infección	61
6.1. Ancho del frente de infección	62
6.1.1. Diferencias por definición del campo de desplazamiento	62
6.1.2. Exponente de rugosidad y dinámico	64
6.2. Velocidad con frente inclinado	67
6.3. Factor de estructura	68
7. Conclusiones	75
Bibliografía	79
Agradecimientos	85

Índice de figuras

2.1.	Perfil de un frente de incendio sobre una hoja de papel	5
2.2.	Mecanismo de defensa de las <i>abejas gigantes</i>	5
2.3.	Crecimiento de bacterias <i>Bacillus subtilis</i>	6
2.4.	Evolución de la plaga en EE.UU.	7
2.5.	Esquema del modelo BD discreto.	8
2.6.	Ancho de la interfaz como función del tiempo para el modelo BD discreto.	9
2.7.	Ancho de la interfaz en el modelo BD discreto para distintos tamaños de sistema y colapso de curvas.	10
2.8.	Auto-afinidad estadística.	12
2.9.	Factor de estructura del modelo BD discreto.	13
2.10.	Término difusivo y no lineal de la ecuación de KPZ.	15
3.1.	Evolución de la peste negra en Europa.	18
3.2.	Solución numérica del modelo S-I-R	21
3.3.	Solución numérica del modelo S-I-R con distintos valores de R_0	22
3.4.	Representación del medio DA sobre una grilla de 20×20	31
3.5.	Paso de suavizado de la heterogeneidad suavizada	32
3.6.	Representación del medio S sobre una grilla de 20×20 con $n = 1$	32
3.7.	Representación del medio S sobre una grilla de 20×20 con distintos n	32
3.8.	Representación del medio DC sobre una grilla de 20×20 con $n = 1$	33
3.9.	Representación del medio DC sobre una grilla de 20×20 con distintos n	33
4.1.	Tiempo de simulaciones en función del tamaño del sistema en CPUs y GPUs.	46
5.1.	Representación del campo de infectados y evolución del centro de masa en medio H.	49
5.2.	Representación del campo de infectados y evolución del centro de masa en medio DA.	50
5.3.	Velocidad del frente infección en función del parámetro de desorden p	51

5.4. Amplitud máxima del frente de infección en función del parámetro de desorden p	53
5.5. Velocidad del frente de infección en función de β_m para los medios H, S, DA y DC	56
5.6. Medio DA en diferentes escalas	57
5.7. Medio DC en diferentes escalas	57
5.8. Amplitud máxima en función de β_m para los medios H, DA, S y DC	58
5.9. Susceptibles remanentes S_1 en función de β_m para los medios H, S, DA y DC	59
6.1. Anchos $\bar{\omega}_0$ y $\bar{\omega}_1$ del frente de infectados	63
6.2. Ancho del frente de infectados $\bar{\omega}_1$ para diferentes tamaños L_y del sistema con $L_x = 2^{16}$	65
6.3. Ancho de saturación ω_{sat} y tiempo de saturación t_x de la interfaz de infectados para diferentes L_y	65
6.4. Se muestra el colapso de las curvas para el ancho $\bar{\omega}_1$ al dividir por L_y^α y escalar el tiempo por L^z	66
6.5. Velocidad del frente de infección en función de m y β_m	68
6.6. Factor de estructura de la interfaz de infectados	69
6.7. Factor de estructura estacionario $\bar{S}_{est}(t)$ y de saturación $\bar{S}_{sat}(t)$ de la interfaz de infectados	71
6.8. Colapso del factor de estructura dado por $u_1(y, t)$ utilizando los exponentes de universalidad $\alpha = 0.42 \pm 0.02$ y $z = 1.64 \pm 0.12$	72

Índice de tablas

5.1.	Parámetros críticos p_c y α_c de los medios H y DA con diferentes γ/β	52
5.2.	Parámetros críticos p_c y α_I del medio DA con diferentes γ/β	54
5.3.	Parámetros críticos β_c y α_c de los medios DC, DA ,S y H.	56
5.4.	<i>Ranking</i> esquemático de heterogeneidades en función de los observables c , I_{max} y S_1 ordenado de mayor a menor.	60
6.1.	Exponentes de rugosidad y dinámico obtenidos de los ajustes $\omega_{sat}(L_y) \propto L_y^\alpha$ y $t_x \propto L_y^z$ respectivamente, para distintas definiciones del ancho de la interfaz.	66

Resumen

Se estudió la propagación de frentes de onda gobernados por ecuaciones de reacción-difusión en el marco del modelo SIR espacial. Dichos frentes de onda podrían utilizarse para caracterizar frentes de infección en una problemática epidemiológica o bien orientarse a una problemática completamente diferente como la de frentes de incendios. Se definió una metodología estadística para la caracterización de los frentes de onda a partir de la cual se obtuvieron resultados cuantitativos respecto de la velocidad, la amplitud media, las propiedades geométricas e incluso la nocividad de los frentes sobre diferentes medios homogéneos y heterogéneos. La heterogeneidad se introdujo en el modelo por medio de una tasa de transmisión espacialmente dependiente. En particular, se exploraron heterogeneidades desordenadas y correlacionadas a partir de lo cual pudo describirse cuantitativamente el efecto que tenía cada una de ellas sobre las características del frente de infección.

Se realizaron simulaciones numéricas masivas para resolver el sistema de ecuaciones de reacción-difusión involucrado en la dinámica. Estas se implementaron de manera eficiente utilizando computación acelerada a través de programación en paralelo sobre procesadores gráficos. De esta manera fue posible obtener resultados sobre sistemas a gran escala en tiempos razonables.

Encontramos una dependencia no trivial del umbral de propagación y la velocidad del frente con el desorden y observamos diferencias dependiendo del tipo de desorden involucrado. Estudiamos la dinámica de la rugosidad del frente de infección y determinamos que la misma cae en la clase de universalidad de KPZ (Kardar-Parisi-Zhang) de origen cinético.

Palabras clave: SISTEMAS COMPLEJOS, MEDIOS DESORDENADOS, ECUACIONES DE REACCIÓN-DIFUSIÓN, MODELO SIR, EPIDEMIOLOGÍA, AUTOAFINIDAD, LEYES DE ESCALEO, KPZ

Abstract

The propagation of wave fronts governed by reaction-diffusion equations was studied within the framework of the spatial SIR model. These wave fronts could be used to characterize infection fronts in an epidemiological problem or be oriented to a completely different problem such as forest fire fronts. A statistical methodology was defined for the characterization of the wave fronts from which quantitative results were obtained regarding the speed, the mean amplitude, the geometric properties and even the harmfulness of the fronts on different homogeneous and heterogeneous media. Heterogeneity was introduced into the model by means of a spatially dependent transmission rate. In particular, disordered and correlated heterogeneities were explored, from which it was possible to quantitatively describe the effect that each of them had on the characteristics of the wavefront.

Massive numerical simulations were performed to solve the system of reaction-diffusion equations involved in the dynamics. These were efficiently implemented using accelerated computing through parallel programming on graphics processors. In this way it was possible to obtain results on large-scale systems in reasonable times.

We find a non-trivial dependence of the propagation threshold and the front velocity with disorder and observe differences depending on the type of disorder involved. We study the dynamics of the roughness of the front and determine that it could be explained by the KPZ (Kardar-Parisi-Zhang) universality class of kinetic origin.

Keywords: DYNAMIC SYSTEMS, DISORDERED MEDIA, REACTION-DIFFUSION EQUATIONS, SIR MODEL, EPIDEMIOLOGY, SELF-AFFINITY, SCALING LAWS, KPZ

Capítulo 1

Estructura de la tesis

El modelado matemático de una dada fenomenología constituye una herramienta fundamental en el proceso de entendimiento cuantitativo de la misma. Más aún, aplicado correctamente sobre una problemática concreta, como lo son las epidemias o los incendios forestales, permite desarrollar estrategias de contención, mitigación y prevención.

Entre la gran diversidad de desafíos que se presentan al momento de describir la dinámica de enfermedades infecciosas sobre una dada población, o bien, la propagación de un incendio, se encuentra el desafío de representar correctamente el carácter heterogéneo de la población o la vegetación [1]. Esto incluye aspectos desde el ámbito comportamental de los individuos hasta la distribución espacial de los mismos. En tanto que para incendios, involucra la topografía e irregularidades del terreno, la diversidad de vegetación o su distribución espacial y hasta contribuciones climáticas.

El objetivo de este trabajo de maestría es dar un paso en el entendimiento del efecto de las heterogeneidades sobre la dinámica de un dado problema con frentes definidos que se propagan espacio-temporalmente. Adicionalmente, la intención es desarrollar en el camino herramientas estadísticas y computacionales que puedan ser utilizadas en problemas completamente diferentes donde la influencia de las características del medio sean de interés.

Para ello se consideró un problema particular, dado por un modelo espacio-temporal de tipo SIR (Susceptibles - Infectados - Recuperados), en donde la movilidad de los individuos es dictada por un término difusivo. Este modelo admite soluciones de onda, lo cual permite definir frentes de infección que se propagan en el espacio y el tiempo. Nos centraremos en describir y entender en profundidad las propiedades de estos frentes de infección, en particular su geometría, su universalidad y los fenómenos críticos asociados. Haremos foco en los efectos que tienen sobre la dinámica las heterogeneidades del medio de sustentación, introducidas en el modelo en la forma de una tasa de transmisión espacialmente heterogénea.

La tesis se divide en seis capítulos además del presente, los cuales se describen brevemente a continuación:

- **Capítulo 2:** se da una introducción general donde se presentan diversos contextos en la naturaleza donde observamos interfaces fuera del equilibrio y vemos cómo las características de las mismas dan información sobre la física subyacente. Luego, se desarrolla resumidamente la teoría asociada, donde definimos las propiedades de interés de las interfaces y los modelos paradigmáticos.
- **Capítulo 3:** se presenta una introducción al modelo SIR en su forma más elemental junto con algunos resultados teóricos conocidos. Luego, se describe precisamente el problema que se aborda en este trabajo, que consiste en un modelo SIR difusivo con medio heterogéneo. Se presenta a su vez, una descripción de las heterogeneidades exploradas y los observables que se utilizarán para caracterizar la dinámica del problema.
- **Capítulo 4:** se presenta una reseña numérica y computacional que abarca el trabajo de investigación desarrollado para resolver ecuaciones diferenciales parciales utilizando procesadores gráficos (GPU). Se muestran explícitamente implementaciones sobre CPU y GPU utilizando *Python*. Para la versión de GPU se utilizó la librería *CuPy*. Por último, se muestra el beneficio obtenido en términos de aceleración al usar GPUs para resolver el tipo de simulaciones realizadas en este proyecto.
- **Capítulo 5:** Mostramos los resultados obtenidos de las simulaciones en lo que respecta a la velocidad, la amplitud y la nocividad de los frentes de infección. Mostramos cómo cada una de estas cantidades se ve afectada por las diferentes heterogeneidades del medio. Observamos fenómenos críticos y determinamos los exponentes críticos asociados en cada caso.
- **Capítulo 6:** En este capítulo profundizamos en el aspecto geométrico y leyes de escala dinámicas del frente de infección. Estudiamos la evolución de la rugosidad del mismo. Determinamos el exponente de rugosidad α y el exponente dinámico z asociado a la interfaz del frente. Observamos también una dependencia cinética de la rugosidad del frente, propia de la clase de universalidad KPZ de origen cinético.
- **Capítulo 7:** se comentan las conclusiones del trabajo y posibles continuaciones y sugerencias a futuro.

Capítulo 2

Introducción

Este capítulo introductorio se divide en tres secciones:

- En la primera sección (2.1) se presenta la motivación y antecedentes enmarcados en el contexto de esta investigación, definiendo así también el objetivo de la misma.
- En la segunda sección (2.2) se describen brevemente conceptos generales de teoría fractal e invariantes de escalas, los cuales son fundamentales para el desarrollo de este trabajo.
- En la tercera sección (2.3) se presentan y se caracterizan tres de las clases de universalidad más relevantes, conocidas como KPZ (Kardar-Parisi-Zhang), EW (Edwards-Wilkinson) y Poisson.

Gran parte de esta introducción estará basada en el libro de Barabási, A.L. y Stanley, H.E. (1995), *Fractal concepts in surface growth* [2]. Para mayor detalle en los temas desarrollados en este capítulo se recomienda la lectura de este libro.

2.1. interfaces fuera del equilibrio en la naturaleza

Parafraseando a Barabási y Stanley (1995) [2], gran parte de nuestras vidas tiene lugar en la superficie/interfaz de algo. En efecto, es probable que usted se encuentre sentado sobre una silla en este momento, o más bien, sobre la superficie de la misma. Por otro lado, la humanidad entera vive en la superficie de la Tierra, que a su vez constituye una interfaz entre el espacio exterior y el interior de nuestro planeta, vivimos en una interfaz. Si se profundiza en esta idea, es posible llegar a pensar que conceptualizamos en función de interfaces. En otro sentido, un sentido más físico, es interesante notar el poder del concepto de interfaz como clasificador y descriptor de fenómenos naturales.

Puede parecer aburrido al principio, dado que observando brevemente a su alrededor encontrará que la mayoría de las cosas que nos rodean son superficies/interfaces que parecieran ser estáticas y no presentar variaciones en el tiempo, las cosas que no cambian suelen ser aburridas. Sin embargo, si se observa con mayor detalle, quizás en otra escala (temporal o espacial), podremos encontrar una riqueza dinámica muy extensa, asombrosa, y muchas veces universal.

A modo de ejemplo, considere nuevamente la superficie de la Tierra, si se observa desde el espacio, la misma parecería ser una superficie suave y estática, sin embargo, es claro que no es así. La superficie de la tierra es rugosa vista en la escala espacial correcta y está en constante cambio si la observamos en el intervalo de tiempo adecuado. Dicho de otra manera, la morfología de una interfaz o una estructura puede cambiar al observarla en diferentes escalas.

Por otro lado, es posible encontrar también ejemplos en la naturaleza en los que la morfología de una interfaz o estructura no cambia al observarla en diferentes escalas. En este caso, la interfaz o estructura es *auto-similar* o *fractal*. Más aún, existen ejemplos en los que la morfología no cambia, pero solo si el cambio de escala es distinto en diferentes direcciones. En este caso, se dice que el objeto tiene la propiedad de ser *auto-afín*. Precisaremos estos conceptos en la sección siguiente.

De momento vamos a conformarnos con observar que la naturaleza es rica en estructuras y fenómenos que pueden ser descritos en términos de interfaces fuera del equilibrio. Algunos ejemplos de ello, muy diferentes microscópicamente, son los que siguen.

Propagación de frentes de incendio: Este es un fenómeno muy habitual en la naturaleza, es posible observarlo sobre una hoja de papel o bien, tristemente, en un bosque. En ambos casos es fácil distinguir una interfaz entre el papel o el bosque quemado y el resto, dando lugar al conocido frente de incendio. Este es un ejemplo de una interfaz que se desplaza sobre un medio desordenado, tanto el papel como el bosque presentan una estructura irregular y heterogénea. En la figura 2.1 se muestra un el perfil del frente de incendio sobre una hoja de papel quemada. Nótese la forma de la interfaz, en principio de apariencia aleatoria [3–5].

Flujo de fluidos en medios porosos: Si alguna vez derramó café sobre una hoja de papel, probablemente haya notado que el café fluye por el papel, formando una interfaz entre la parte del papel que está mojada y la que no lo está. Este es otro ejemplo de una interfaz que se desplaza sobre un medio desordenado. Una vez que el café es absorbido por el papel, por el fenómeno de capilaridad, la interfaz se estabiliza dejando una estructura característica que con las herramientas adecuadas puede ser descrita cuantitativamente [6, 7].

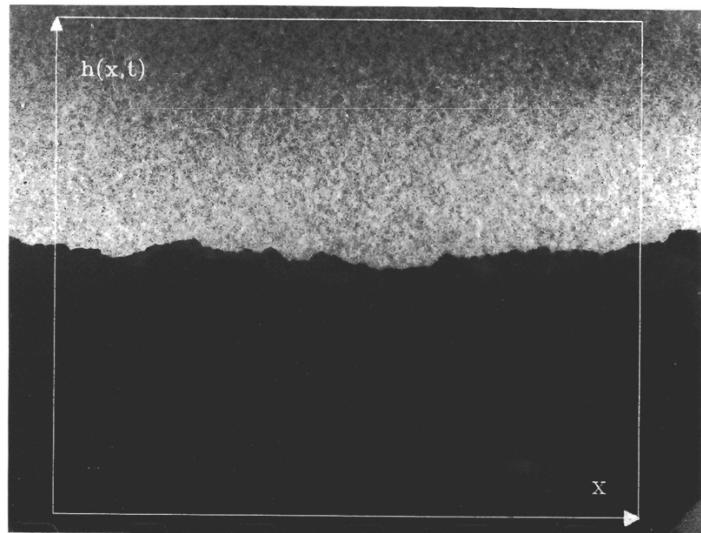


Figura 2.1: Perfil de un frente de incendio sobre una hoja de papel. Figura extraída de Ref. [3].

Mecanismo de defensa de las *abejas gigantes*: Las *abejas gigantes* (*Apis dorsata*), son una especie de abeja nativa del sudeste asiático, Indonesia y Australia. Estas abejas son conocidas por su agresividad y por defender su colmena a toda costa. Uno de sus variados mecanismos de defensa es muy peculiar, cuando la colmena se ve amenazada por insectos más grandes, como avispas, las abejas, que se encuentran alrededor de la colmena, colectivamente se coordinan para producir olas o frentes que espantan a los predadores¹ [8–10]. En la figura 2.2 se muestra una secuencia de imágenes de este comportamiento, similar a las olas humanas que se dan en los estadios de fútbol [11, 12].



Figura 2.2: Secuencia de imágenes que muestra a las *abejas gigantes* haciendo la “ola” como mecanismo de defensa. Figura extraída de Ref. [13].

¹Si se encuentra leyendo esto en una computadora puede mirar un video de este comportamiento [aquí](#).

Crecimiento de bacterias: Otro ejemplo de tipo biológico es el crecimiento de colonias de bacterias. En la figura 2.3 se muestra la morfología característica del crecimiento de una especie de bacteria denominada *Bacillus subtilis* sobre un medio de cultivo. Se observa que la estructura de la interfaz es sensible a la concentración de nutrientes y otros parámetros experimentales, de modo que la forma de la interfaz tiene información sobre el medio de cultivo [14, 15].

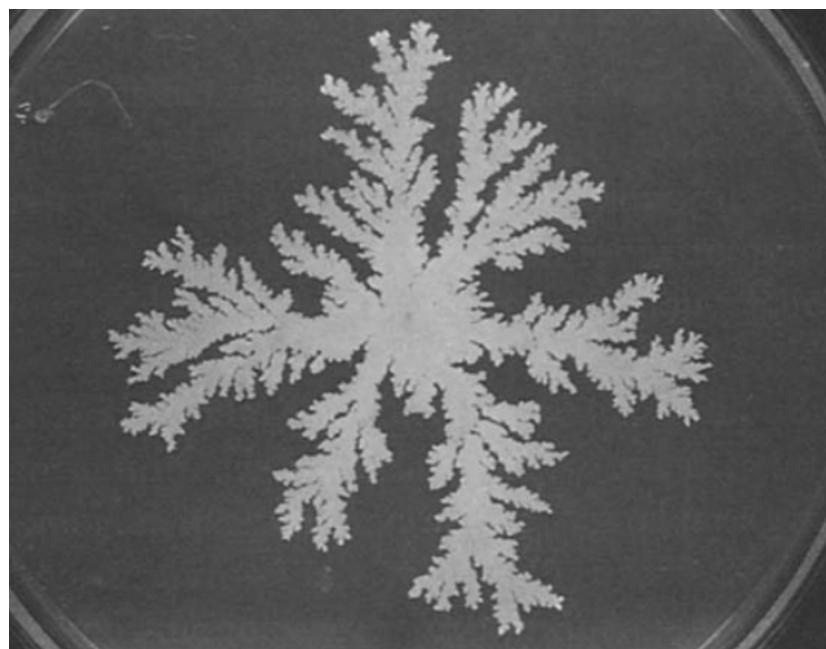


Figura 2.3: Crecimiento de bacterias *Bacillus subtilis*. Figura extraída de [14].

Frentes de infección: En epidemiología es posible definir frentes de infección, en donde la interfaz podría entenderse como la frontera entre la población infectada y la población no infectada. En el año 1900 la bacteria *Y. pestis* llegó sobre algún pasajero (humano o animal) proveniente de Oriente al puerto de San Francisco en Estados Unidos. Esta bacteria es la causante de la peste bubónica, una enfermedad mortal que dio lugar a la Peste Negra en Europa en el siglo XIV. Los notables avances en la medicina y la higiene desde aquel entonces han reducido el impacto de esta enfermedad. Sin embargo, la bacteria sigue presente en el territorio de Estados Unidos, dado que se mantiene endémica sobre la población de roedores, ardillas y conejos. Por esto, se presentan casos esporádicos aún en la actualidad. Observando el registro de casos desde 1900 hasta 2012 puede construirse la figura 2.4, en al que se ve cómo la peste progresó lentamente desde la costa oeste de Estados Unidos hacia la costa este, a unos 50 km por año, trasladada fundamentalmente por los animales pequeños. Esta situación constituye una preocupación real dado que no se sabe qué pasará cuando llegue a las grandes ciudades del este de Estados Unidos, donde se encuentra la mayor densidad de población, humana y de roedores [16–18].

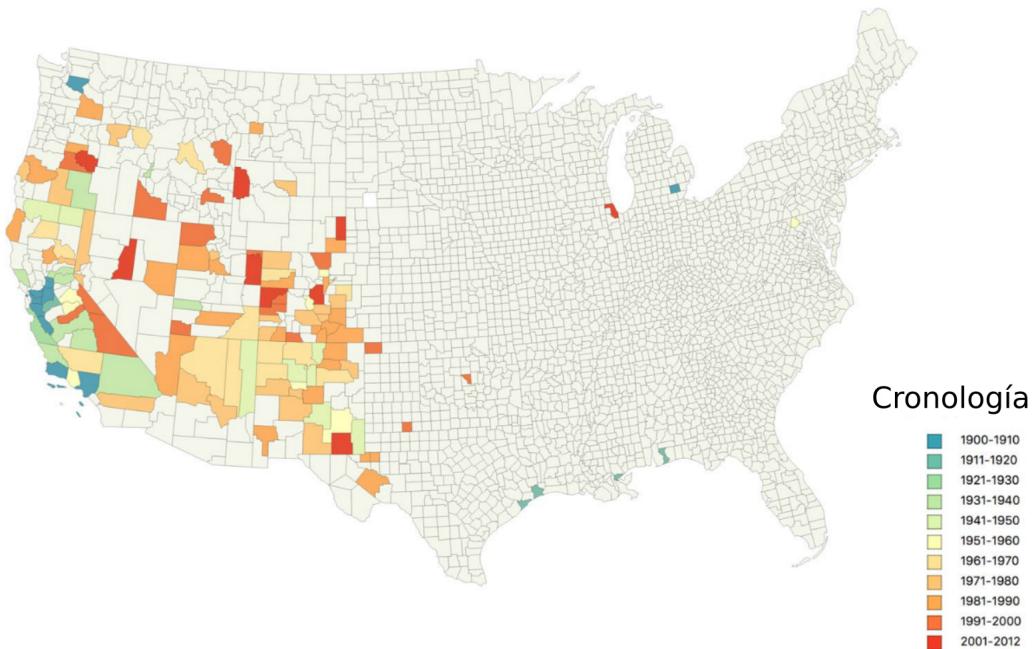


Figura 2.4: Evolución de la plaga en EE.UU., se muestran coloreados los condados donde se reportó por primera vez un caso y el color indica la cronología. Figura extraída de Ref. [17].

Discusión

En muchos de estos ejemplos notamos que la aleatoriedad, ya sea de la composición del papel que se quema, o bien, del movimiento de las bacterias en el medio de cultivo, constituye un aspecto crucial en la morfología de las interfaces en cuestión. Por otro lado, observamos que algunos otros casos corresponden con comportamientos o dinámicas subyacentes complejas, tal es el caso de las olas de abejas y los frentes infección epidemiológicos.

Lo interesante es que, sea el fenómeno que sea, la forma de la interfaz tiene información sobre la física subyacente. Más aún, a menudo es posible encontrar conexiones fundamentales entre problemas esencialmente diferentes. En la sección siguiente presentamos brevemente las herramientas más comúnmente utilizadas para estudiar y clasificar las interfaces y sus dinámicas.

2.2. Conceptos de escala

Para presentar las herramientas y conceptos fundamentales vamos a utilizar como ejemplo ilustrativo un modelo de crecimiento simple y conocido, llamado crecimiento balístico (diremos modelo BD, por *Ballistic Deposition*). La idea es entender las propiedades fundamentales y universales del crecimiento de interfaces, dejando los detalles del proceso de lado.

El proceso del modelo BD es el siguiente: una partícula es liberada en una posición aleatoria, con distribución uniforme, sobre una superficie plana a una distancia mayor que la altura máxima de la interfaz. La partícula cae en línea recta hasta entrar en contacto con la superficie, punto en el cual se fija. En la figura 2.5 se muestra esquemáticamente este proceso. Iterando con N partículas obtenemos el crecimiento balístico. Consideraremos por simplicidad, pero sin pérdida de generalidad, el caso unidimensional.

2.2.1. Ancho de la interfaz

Para comenzar a caracterizar la interfaz primero definimos la altura de la misma como $h(x, t)$, que da la altura en la posición x a tiempo t . A partir de esta podemos definir también la altura media de la interfaz $h_m(t)$ como,

$$h_m(t) = \langle h(x, t) \rangle_x = \frac{1}{L} \int_0^L h(x, t) dx, \quad (2.1)$$

donde L es la longitud de la superficie de apoyo, o bien, el tamaño del sistema en la dirección transversal al crecimiento. Si la cantidad de partículas que llegan a una determinada posición por unidad de tiempo es constante, entonces la altura media crece linealmente con el tiempo

$$h_m(t) \propto t. \quad (2.2)$$

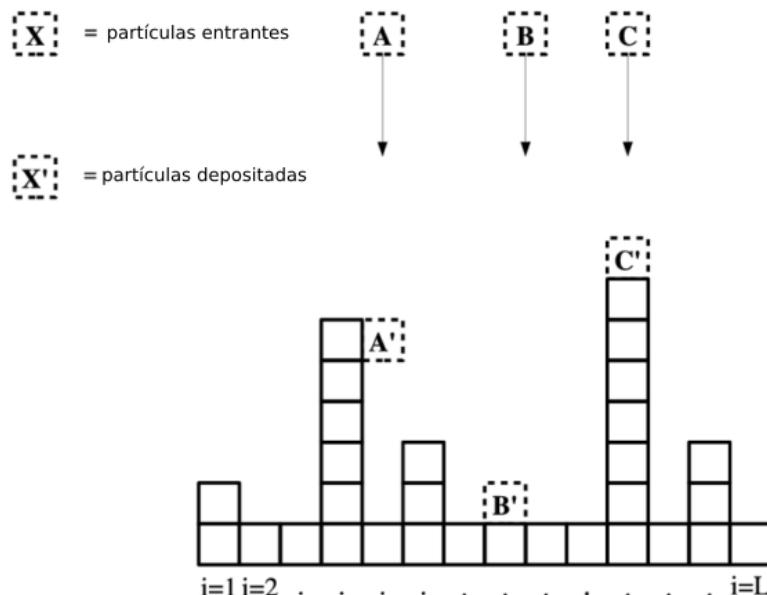


Figura 2.5: Esquema del modelo BD discreto: la partícula A cae hasta fijarse a la superficie al entrar en contacto de manera lateral con la misma, por su parte las partículas B y C, se fijan a la superficie de manera vertical. La posibilidad de fijación lateral es de gran relevancia en la morfología de la interfaz. Figura extraída de Ref. [19].

Una cantidad esencial para estudiar el crecimiento es el ancho medio de la interfaz, dado simplemente por la desviación estándar de la altura, es decir,

$$\omega(L, t) = \sqrt{\langle (h(x, t) - h_m(t))^2 \rangle_x} = \frac{1}{\sqrt{L}} \left(\int_0^L (h(x, t) - h_m(t))^2 dx \right)^{1/2}, \quad (2.3)$$

donde $\overline{\dots}$ significa promedio sobre distintas realizaciones. Si graficamos el ancho como función del tiempo para el modelo BD, obtenemos la figura 2.6, que a su vez es un resultado representativo de lo que resulta típicamente al observar esta cantidad.² Vemos que, inicialmente, el ancho crece siguiendo una ley de potencia,

$$\omega(L, t) \propto t^\beta, \quad (2.4)$$

donde β se conoce como el exponente de crecimiento, el cual caracteriza la dependencia temporal del proceso de rugosidad. Luego de un tiempo de saturación t_x , el ancho deja de crecer y se mantiene constante en un valor ω_{sat} .

Consideremos ahora la figura 2.7 (a), donde se muestra la evolución del ancho para distintos tamaños de sistema L . Vemos que, tanto el valor de saturación del ancho w_{sat} como t_x crecen con el tamaño del sistema. En particular, puede verse que se cumple

$$\omega(L, t \gg t_x) \propto L^\alpha, \quad (2.5)$$

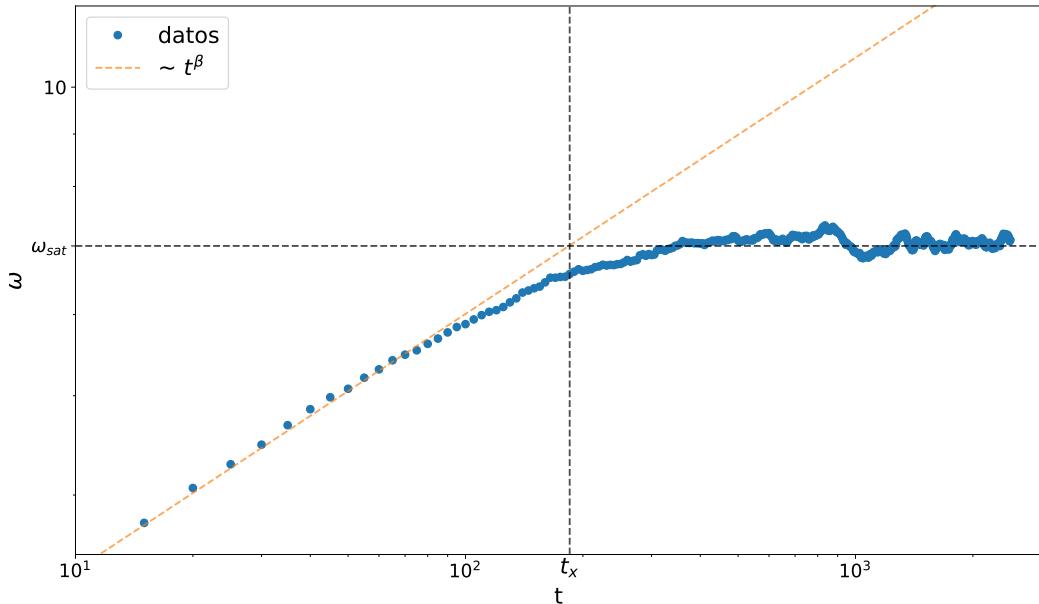


Figura 2.6: Ancho de la interfaz ω como función del tiempo para el modelo BD discreto con $L = 200$, $N = 50000$ partículas y 500 realizaciones. Se observa el régimen de crecimiento, donde $\omega(L, t \ll t_x) \propto t^\beta$ y el régimen de saturación $\omega(L, t \gg t_x) = \omega_{sat}$, que se alcanza a un tiempo de saturación t_x .

²Se utilizó $t = n/L$ donde n es el número de iteraciones realizadas del modelo BD.

donde α es el exponente de rugosidad. Mientras que el tiempo de correlación crece como,

$$t_x \propto L^z, \quad (2.6)$$

donde z es el exponente dinámico.

Teniendo estas leyes de potencia en cuenta, es posible colapsar las curvas de la figura 2.7a en una única curva (figura 2.7b), dividiendo $\omega(L, t)$ por $\omega_{sat} \propto L^\alpha$. De esta manera hacemos que todas las curvas saturen en el mismo valor. Luego, para hacer coincidir el tiempo de saturación, escaleamos el tiempo por $t_x \propto L^z$. Con este argumento obtenemos la relación de escaleo de *Family-Vicsek* [20],

$$\omega(L, t) \propto L^\alpha f\left(\frac{t}{L^z}\right), \quad (2.7)$$

donde $f(u)$ es una función universal que cumple,

$$f(u) = \begin{cases} u^\beta & \text{si } u \ll 1 \\ cte & \text{si } u \gg 1. \end{cases} \quad (2.8)$$

Hasta el momento hemos presentado tres exponentes que describen cuantitativamente diferentes aspectos del crecimiento de la interfaz, estos son, β , α y z . Es importante notar que estos exponentes no son independientes entre sí, consideremos el momento donde el ancho del frente alcanza su valor de saturación, es decir, $t \approx t_x$. En este momento, el ancho de la interfaz puede expresarse de dos formas equivalentes, considerando el régimen de crecimiento $\omega(L, t_x) \propto t_x^\beta$, mientras que en el régimen de

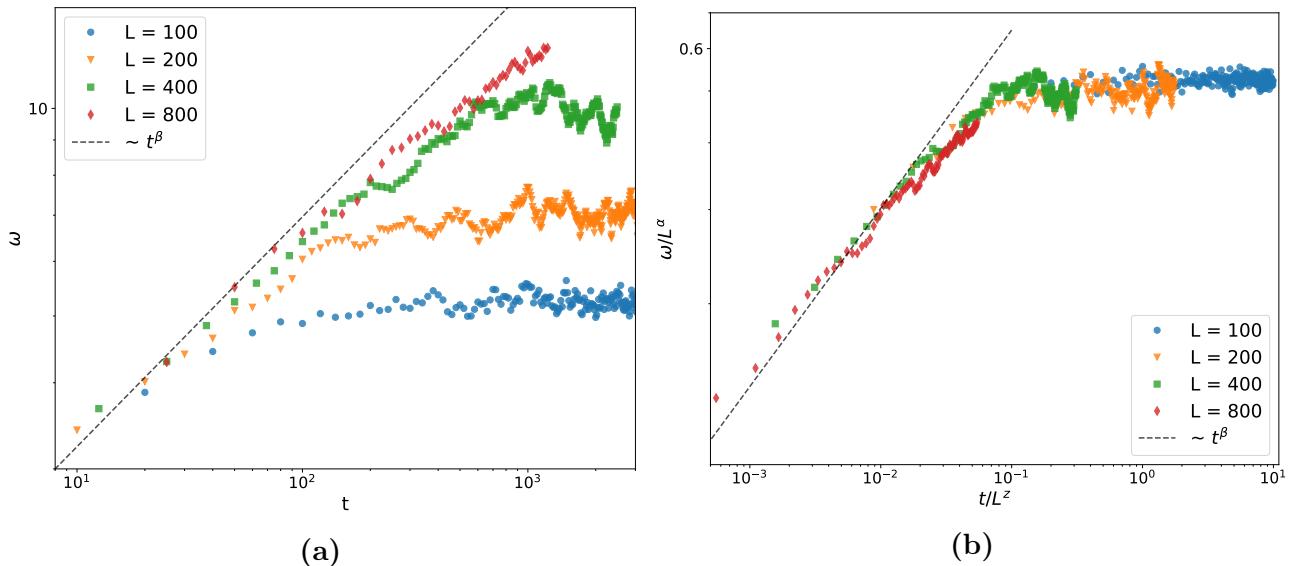


Figura 2.7: (a) Ancho de la interfaz como función del tiempo para el modelo BD discreto con distintos tamaños de sistema L y $N = 10^6$ partículas. (b) Colapso de las curvas que permite obtener los exponentes de rugosidad α y dinámico z .

saturación $\omega(L, t) \propto L^\alpha$, usando además que $t_x \propto L^z$, obtenemos que

$$z = \frac{\alpha}{\beta}. \quad (2.9)$$

Por otro lado, desde un punto de vista físico, la pregunta que debe hacerse es: ¿por qué el ancho satura en un valor finito? Más aún, ¿por qué el ancho satura en un valor que depende del tamaño del sistema? Esto se debe a que el crecimiento de la interfaz está correlacionado. En el caso del modelo BD esto se corresponde con el hecho de que las partículas pueden pegarse lateralmente al frente, de modo que la altura en la posición donde cae esa partícula será igual o mayor a la de sus vecinos. Esta correlación microscópica se propaga lateralmente sobre la interfaz a medida que pasa el tiempo, de modo que la altura de un punto puede impactar a una distancia cada vez mayor. De esta manera comienzan a crearse regiones en la interfaz con una longitud característica $\xi_{||}(t)$ denominada longitud de correlación dinámica. Dado que el sistema es de tamaño finito L , la longitud de correlación no puede ser mayor que L , es decir,

$$\xi_{||}(t) \propto L \quad \text{con} \quad t \gg t_x. \quad (2.10)$$

Además, dado que $t_x \propto L^z$, en el tiempo de saturación se cumple $\xi_{||} \propto t_x^{1/z}$, que de hecho también vale para $t \ll t_x$,

$$\xi_{||}(t) \propto t^{1/z}. \quad (2.11)$$

De modo que el ancho de la interfaz deja de crecer cuando la longitud de correlación alcanza el tamaño del sistema, fijando así la máxima magnitud de fluctuaciones que puede tener la interfaz. Así mismo, queda claro por qué el valor de saturación depende del tamaño del sistema, al ser más grande el sistema lleva más tiempo correlacionarlo por completo y por lo tanto el ancho de la interfaz alcanza un valor mayor antes de saturar.

2.2.2. Factor de estructura de la interfaz

Otro observable de utilidad para caracterizar el crecimiento de interfaces es el factor de estructura $S(q, t)$, definido en términos de $h(x, t)$ como

$$S(q, t) = \overline{\tilde{h}(q, t)\tilde{h}(-q, t)}, \quad (2.12)$$

donde

$$\tilde{h}(q, t) = \int h(x, t)e^{-iqx}dx. \quad (2.13)$$

es la transformada de Fourier sobre el espacio del campo de desplazamiento o alturas $h(x, t)$ que definen la interfaz. En este caso, si la condición inicial de la interfaz es

plana, el factor de estructura sigue las siguientes leyes de potencia según sea $\xi_{||}(t)$,

$$S(q, t) \propto \begin{cases} t^{(1+2\alpha)/z} & \text{si } q \ll \xi_{||}(t)^{-1} \\ q^{-(1+2\alpha)} & \text{si } q \gg \xi_{||}(t)^{-1}. \end{cases} \quad (2.14)$$

En donde la dependencia temporal se da en todas las escalas por arriba de $\xi_{||}(t)$, donde el sistema se encuentra descorrelacionado, y por ende se tiene un espectro de ruido blanco, independiente de q , cuya magnitud crece con el tiempo. Por otro lado, cuando se observan las escalas por debajo de $\xi_{||}(t)$ encontramos una ley de potencia que describe una estructura auto-afín.

Que un objeto sea auto-afín implica que es invariante ante transformaciones anisotrópicas de la forma

$$x \rightarrow x' = bx \quad (2.15)$$

$$h \rightarrow h' = b^\alpha h. \quad (2.16)$$

En la forma más estricta, se dice que es auto-afín si tras esta transformación el objeto queda exactamente igual que antes. Sin embargo, existe una noción más débil de autoafinidad, en donde se dice que el objeto es estadísticamente auto-afín si luego de aplicar la transformación anisotrópica, el objeto conserva las mismas propiedades estadísticas que antes. Esta noción de auto-afinidad es la que comúnmente se utiliza al hablar de objetos auto-afines en la naturaleza ya que usualmente las mediciones experimentales están lejos de ser perfectas, además el proceso de generación de estas estructuras típicamente tiene un componente estocástico que imposibilita que sean exactamente auto-afines. En la figura 2.8 se muestra esquemáticamente el concepto de estructura estadísticamente auto-afín.

Por último, cuando la longitud de correlación $\xi_{||}(t)$ alcanza el tamaño del sistema

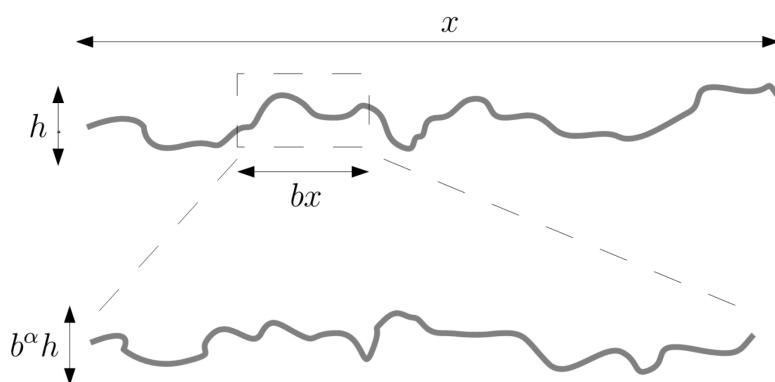


Figura 2.8: Auto-afinidad estadística: se muestra cómo el perfil de la interfaz permanece relativamente invariante ante una transformación anisotrópica.

L , similar a lo que pasa con el ancho, el factor de estructura deja de evolucionar y el sistema queda completamente correlacionado, con $S(q, t \gg t_x) \propto q^{-(1+2\alpha)}$.

En la figura 2.9a se muestra el factor de estructura $S(q, t)$ para el modelo BD. Vemos cómo el mismo sigue la ley de potencias $S(q, t) \propto q^{-(1+2\alpha)}$ para $q \gg \xi_{||}(t)^{-1}$, mientras que para $q \ll \xi_{||}(t)^{-1}$ se comporta como $S(q, t) \propto t^{(1+2\alpha)/z}$, correspondiente a los *plateaus*. De manera similar que con el ancho es posible escalar el factor de estructura para hacer que estas curvas colapsen en una sola, tal como se muestra en la figura 2.9b.

El factor de estructura ha mostrado ser más confiable a la hora de estimar el exponente de rugosidad α que a través del ancho de la interfaz, u otros observables en el espacio real [21], fundamentalmente porque regiones de escalamiento diferentes aparecen claramente separadas en el espacio recíproco.

2.3. Universalidad: Poisson, KPZ y EW

En la sección anterior se vio cómo el crecimiento de una interfaz puede ser caracterizado por un par de exponentes, por ejemplo (α, z) . Estos exponentes definen lo que se conoce como la clase de *universalidad* de la interfaz. Se dice universalidad dado que resulta que una gran diversidad de modelos que describen el crecimiento de una interfaz obedecen fundamentalmente las mismas relaciones de escala, independiente de los detalles del modelo.

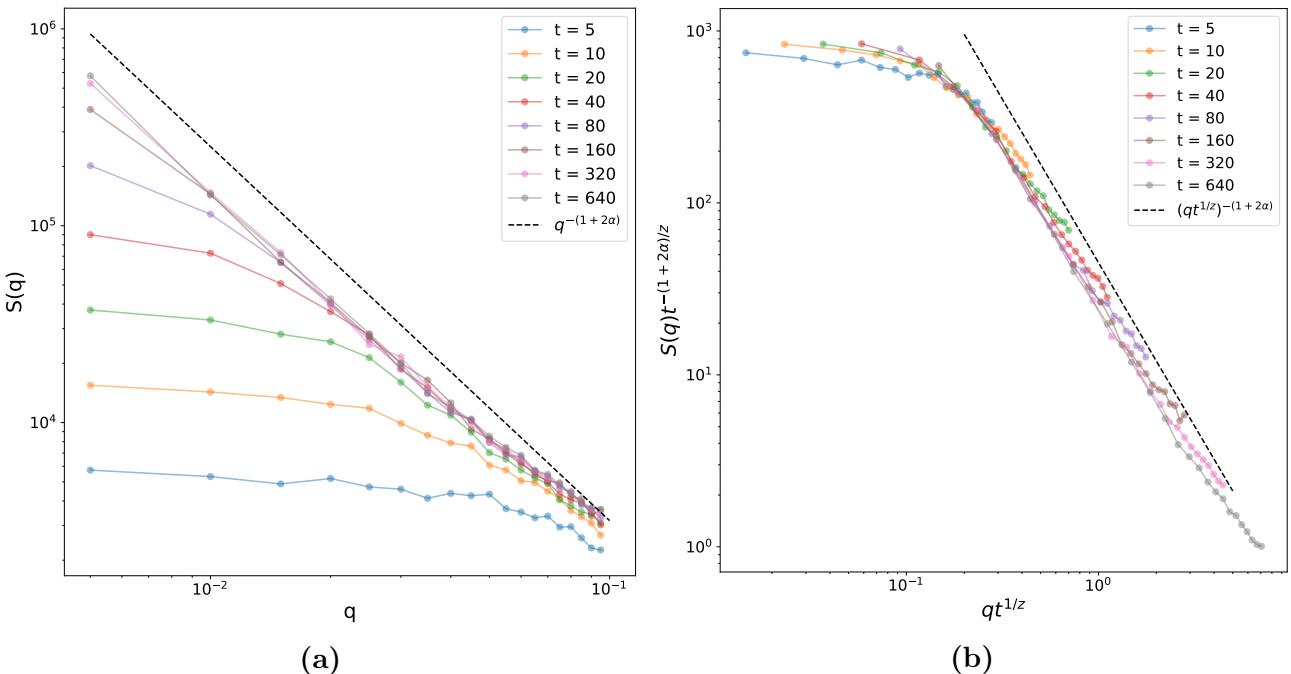


Figura 2.9: (a) Factor de estructura para el modelo BD discreto con $L = 200$, $N = 50000$ partículas y 500 realizaciones. (b) Colapso de las curvas que permite obtener los exponentes de rugosidad α y dinámico z .

Las tres clases de universalidad más conocidas son: Poisson, Edwards-Wilkinson (EW) y Kardar-Parisi-Zhang (KPZ) [22, 23]. En la primera, la altura de la interfaz en cada punto es independiente de la altura de los demás. En la segunda, las fluctuaciones son moderadas por términos difusivo que tienden a suavizar la interfaz. En la última, puntos cercanos de la interfaz se ayudan mutuamente para crecer, resultando en un efecto de amplificación no lineal de las fluctuaciones que compite con un término difusivo. Esta última descripción debiera recordarle el modelo BD, en donde los puntos alocados en cierta vecindad se ayudan mutuamente a crecer. En efecto, el modelo BD pertenece a la clase de universalidad KPZ en una dimensión.

Para estudiar las diferentes universalidades que pueden encontrarse, lo que se hace es proponer o bien derivar una ecuación diferencial estocástica que describa la evolución del campo de desplazamiento $h(x, t)$. Dicha ecuación para KPZ fue propuesta precisamente por Kardar, Parisi y Zhang [23] y es la siguiente

$$\frac{\partial h}{\partial t} = \nu \nabla^2 h + \frac{\lambda}{2} (\nabla h)^2 + F + \eta(\mathbf{x}, t), \quad [\text{KPZ}] \quad (2.17)$$

donde el primer término de la derecha describe una relajación de la interfaz ejercida por una tensión superficial ν (o difusión si vemos a h como una concentración), el término no lineal $(\nabla h)^2$ promueve el crecimiento de la interfaz en la dirección normal local de la misma, permitiendo así el crecimiento lateral. F es una constante que podría ser el valor medio de la cantidad de partículas por unidad de tiempo que llega a la interfaz, es decir, $F = v_0 \simeq \partial_t h_m$. Mientras que el último término es un ruido que satisface

$$\overline{\eta(\mathbf{x}, t)} = 0, \quad \overline{\eta(\mathbf{x}, t)\eta(\mathbf{x}', t')} = 2\kappa\delta^d(\mathbf{x} - \mathbf{x}')\delta(t - t'), \quad (2.18)$$

donde $\overline{\dots}$ denota promedio sobre realizaciones de ruido. La ecuación no lineal 2.17 puede resolverse de manera exacta en una dimensión $d = 1$, lo que permite obtener los siguientes exponentes de escaleo³

$$\alpha = \frac{1}{2}, \quad z = \frac{3}{2} \quad \text{y} \quad \beta = \frac{1}{3}. \quad (2.19)$$

El modelo BD es un modelo simple que ayuda a entender a nivel físico a qué se deben las características de la interfaz. Sin embargo, lo interesante es que es posible encontrar modelos más complejos que también pertenecen a esta clase de universalidad, más aún, resultados experimentales de fenómenos complejos muestran la dinámica de KPZ [24, 25].

En las figuras 2.10a y 2.10b puede observar una representación gráfica del efecto que tienen sobre $h(x, t)$ el término difusivo o elástico $\nabla^2 h$ y el término no lineal $(\nabla h)^2$

³Los exponentes con $d > 1$, no pueden determinarse analíticamente, sin embargo, se han obtenido numéricamente y pueden encontrarse reportados [2].

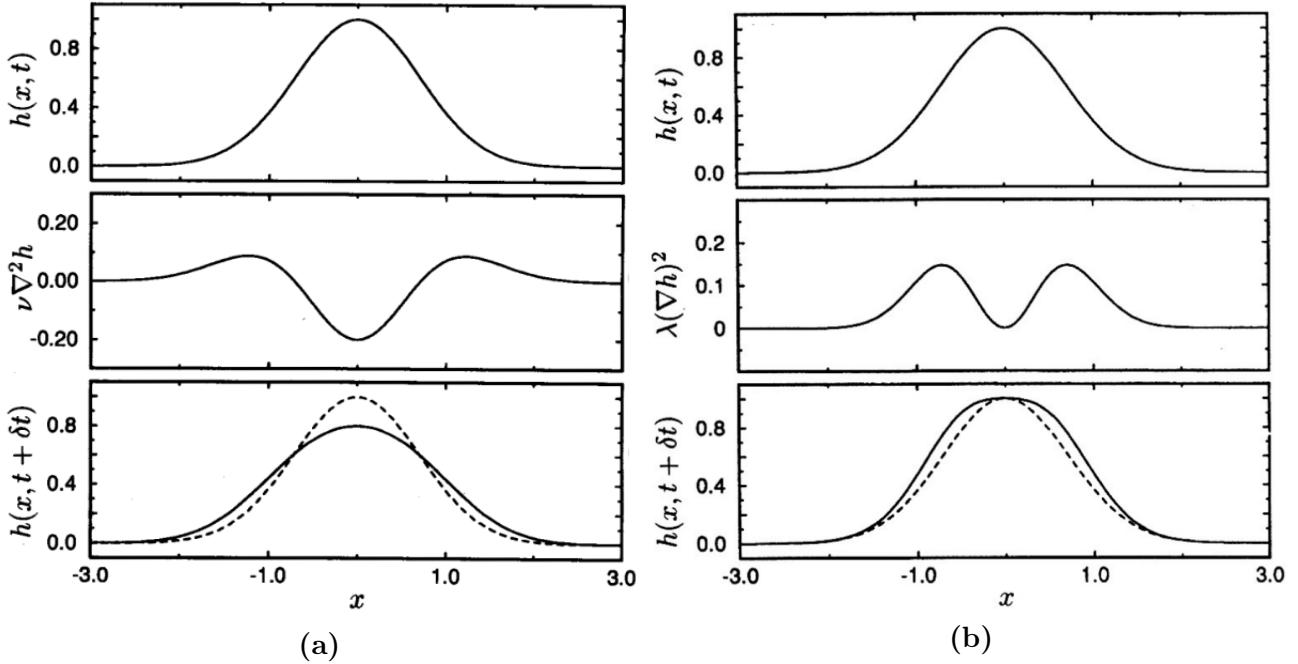


Figura 2.10: Efecto sobre la interfaz $h(x, t)$ de (a) el término difusivo y (b) el término no lineal. Figuras extraídas de Ref. [2].

respectivamente, que aparecen en la ecuación 2.17.

De momento hemos hablado solo de KPZ, pero ¿qué queda para decir de las universalidades de EW y Poisson? Afortunadamente, KPZ es el más complejo de los tres dado que es no lineal, en lo que respecta a EW, la ecuación que describe su evolución es simplemente 2.17 pero con $\lambda = 0$, es decir,

$$\frac{\partial h}{\partial t} = \nu \nabla^2 h + F + \eta(\mathbf{x}, t). \quad [\text{EW}] \quad (2.20)$$

que puede resolverse en cualquier dimensión obteniendo los exponentes de escaleo,

$$\alpha = \frac{2-d}{2}, \quad z = 2 \quad \text{y} \quad \beta = \frac{2-d}{4}. \quad (2.21)$$

Nótese que si $d > 2$ entonces $\alpha < 0$, lo que significa que la interfaz se contrae, se hace plana, cualquier perturbación que genere un ancho no nulo es suprimida por la tensión superficial.

Por otro lado, un proceso de Poisson es, como podría esperar, uno en donde la interfaz crece de manera completamente aleatoria, la ecuación correspondiente es la siguiente,

$$\frac{\partial h}{\partial t} = F + \eta(\mathbf{x}, t). \quad [\text{Poisson}] \quad (2.22)$$

En este caso, únicamente tenemos un exponente, el exponente de crecimiento

$$\beta = \frac{1}{2}, \quad [\text{Poisson}] \quad (2.23)$$

dado que no hay un fenómeno de saturación para el ancho de la interfaz en un sistema finito, el ancho crece indeterminadamente, debido a la ausencia de correlación.

Ejemplos de modelo comunes que responden a Poisson y EW, son el crecimiento aleatorio y el crecimiento aleatorio con difusión, respectivamente. El primero es lo que se imagina, se crece la interfaz de manera aleatoria, las partículas caen aleatoriamente en un lugar y ahí quedan. La diferencia de este modelo con BD es que las partículas no pueden pegarse lateralmente a la interfaz. Por otro lado, en el crecimiento aleatorio con difusión, las partículas caen aleatoriamente y una vez que entran en contacto con la interfaz tienen permitido difundir cierta distancia hasta encontrar la altura mínima.

Un comentario final que será de importancia más adelante en el desarrollo de este trabajo, es sobre la velocidad de propagación de la interfaz para el caso de KPZ. Si uno integra espacialmente la ecuación 2.17, promedia sobre desorden y define la velocidad media de propagación de la interfaz como

$$v = \dot{h}_m(t) = \frac{d}{dt} \left(\frac{1}{L} \int_0^L d^d \mathbf{x} h(\mathbf{x}, t) \right), \quad (2.24)$$

resulta que la velocidad de propagación depende únicamente del término no lineal de 2.17,

$$v = v_0 + \frac{\lambda}{2L} \int_0^L d^d \mathbf{x} \overline{(\nabla h)^2}, \quad (2.25)$$

con $v_0 = F$.

Capítulo 3

Modelo SIR

3.1. Historia

Desde tiempos remotos las epidemias y pandemias han sido origen de sufrimiento para la humanidad. Cada individuo, pueblo, ciudad o civilización amenazada por una epidemia ha tenido que enfrentarlas de una u otra manera. La historia de la humanidad está llena de acontecimientos epidemiológicos que han causado grandes pérdidas de vidas humanas y cambios en la forma de vida de las sociedades. La peste negra, la gripe española, la viruela, la tuberculosis, la malaria y la fiebre amarilla, entre otras, han sido algunas de las epidemias más devastadoras de la historia.

A continuación haremos un breve repaso histórico de dos de las epidemias mencionadas arriba, la peste negra y la gripe española. Destacaremos algunos detalles de cada una de ellas para ilustrar la complejidad dinámica de las epidemias y la cantidad de detalles que debe tenerse en cuenta a la hora de modelarlas. Además, dados los recientes acontecimientos epidemiológicos, creo que resulta de interés poner en observación algo de la extensa historia epidemiológica de la humanidad.

Peste Negra

En la Edad Media, la Peste Negra, fue una epidemia de peste bubónica que se extendió por Europa entre 1347 y 1350, causando la muerte de entre un tercio y la mitad de la población europea [26]. La enfermedad era producida por la bacteria *Yersinia pestis*, que se transmitía por la picadura de pulgas infectadas. De hecho, el mecanismo de transmisión de la enfermedad fue un misterio durante mucho tiempo. En 1855, una epidemia de peste bubónica, ocasionada por la misma bacteria, comenzó en el sur de China. Fue entonces cuando pudieron obtenerse resultados de laboratorio y recién en 1897, M. Ogata, y en 1898, P. L. Simonds, de manera independiente llegaron a la hipótesis de que la enfermedad se transmitía por medio de la picadura de pulgas infectadas que vivían fundamentalmente en ratas. Lo cual fue confirmado unos años

más tarde.

En resumen, el mecanismo de transmisión era el siguiente: algunas de las pulgas sufrían un bloqueo estomacal luego de alimentarse de una rata infectada, como consecuencia, estas pulgas comenzaban a picar con mayor frecuencia de lo habitual dado que la sangre que ingerían no les llegaba a los intestinos para ser digerida, es decir, tenían hambre constantemente. El punto crucial de la cadena de transmisión, reside en la existencia de dos poblaciones distintas de ratas, una resistente y otra que no lo es. La especie resistente era la responsable de mantener la enfermedad endémica albergando las pulgas infectadas. Mientras que la especie no resistente generaba una escasez de alimento para las pulgas, las cuales se veían forzadas a cambiar de huésped, en este caso, hacia el humano. Finalmente, la enfermedad se transmitía cuando las pulgas con bloqueo estomacal vomitaban sangre infectada al picar a una persona [28].

La plaga se introdujo a Europa por Italia en diciembre de 1347 y se extendió rápidamente, a un ritmo de 300 - 600 km por año [29]. Llegó por un barco genovés proveniente de Caffa, en el Mar Negro. La ciudad de Caffa tenía un puerto crucial para la ruta comercial entre Europa y Asia, y estaba bajo control genovés. Resulta que en 1347 dicha ciudad se encontraba sitiada por las tropas de la Horda Dorada, un estado Mongol establecido en Europa oriental y Asia central tras la desintegración del Imperio Mongol. Dicho ejército estaba sufriendo las consecuencias de la peste que habían traído desde Oriente. Los mongoles catapultaron cadáveres por sobre las murallas de la ciudad



Figura 3.1: Evolución de la peste negra en Europa. Figura extraída de Ref. [27].

sitiada, con el fin de infectar a los genoveses, dando lugar al primer registro de guerra bacteriológica de la historia. Fue así cómo la huída de los genoveses desencadenó la peor epidemia que haya sufrido Europa [16].

En la figura 3.1 se muestra la propagación espacio-temporal de la peste negra en Europa. Las rutas comerciales fueron las principales vías de propagación de la plaga.

Gripe Española

La gripe española, se extendió por todo el mundo entre 1918 y 1920, se estima que causó la muerte de unas 40 millones de personas y entre un 25-30 % de la población mundial padeció la enfermedad [30]. Se originó en Kansas, Estados Unidos, y se extendió por todo el mundo en menos de un año. Curiosamente, se denominó gripe española simplemente porque España no participó en la Primera Guerra Mundial que acontecía simultáneamente, por lo tanto, no se censuraron los datos de la enfermedad. La gripe española fue una pandemia de gripe de tipo A, causada por el virus H1N1. Fue la primera de tres pandemias causadas por este virus, la última de las cuales ocurrió en 2009.

A diferencia de otros virus de tipo A, que típicamente tienen mayor mortalidad sobre niños y ancianos, este virus impactó fuertemente sobre la población joven-adulta (20-40 años) [31]. El mecanismo de transmisión es como el de la gripe común, es decir, aéreo, por medio de gotitas de saliva que se expulsan al toser, estornudar o respirar.

Es interesante notar también en este caso cómo el contexto histórico influyó en la propagación de la enfermedad. En 1918, la Primera Guerra Mundial estaba en su apogeo y muchos países se encontraban en medio de una crisis económica. Se piensa que la elevada tasa de transmisión de la enfermedad, con un número de reproducción básico entre 2 y 3 [32], se debió a la enorme cantidad de tropas movilizadas por todo el mundo. Además, esto mismo promovió mutaciones en el virus.

3.2. Modelo SIR de campo medio

Naturalmente, motivados por la sección anterior, estamos interesados en modelar la propagación de una enfermedad en una población. De esta manera podríamos entender mejor el comportamiento de la enfermedad y predecir su evolución. En este sentido, el modelo SIR constituye la primera aproximación a este problema.

El modelo SIR describe la dinámica de tres grupos característicos de un sistema, denominados comúnmente como **Susceptibles**, **Infectados** y **Recuperados**, de ahí su nombre. Es el modelo más simple para describir la propagación de una enfermedad infecciosa, fue propuesto originalmente hace casi un siglo, en 1927, por *Kermack* y *McKendrick* [33].

El mismo está formulado de la siguiente manera: sean $S(t)$, $I(t)$ y $R(t)$ la fracción de susceptibles, infectados y recuperados de una población dada a tiempo t respectivamente, entonces, la dinámica de estos grupos está descrita por el siguiente sistema de ecuaciones diferenciales ordinarias

$$\frac{dS}{dt} = -\beta SI, \quad (3.1)$$

$$\frac{dI}{dt} = \beta SI - \gamma I, \quad (3.2)$$

$$\frac{dR}{dt} = \gamma I. \quad (3.3)$$

Donde β corresponde a una tasa de transmisión mientras que γ corresponde a una tasa de recuperación.

Usualmente, la ecuación para R (3.3) no se escribe, ya que puede ser reemplazada por la más simple $S+I+R = 1$. Puede verse por inspección que el sistema de ecuaciones (3.1 - 3.3) cumple $\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0$.

Para entender por qué este simple sistema de ecuaciones (3.1 - 3.3) podría describir la dinámica del problema observamos que el mismo está compuesto esencialmente por dos términos, el término de transmisión βSI y el de recuperación γI . Cualitativamente, resulta razonable que la magnitud de sujetos infectados por unidad de tiempo aumente con el producto de la cantidad de infectados y susceptibles, de ahí el término de transmisión. Por otro lado, la cantidad de infectados que se recuperan por unidad de tiempo es entendible que sea proporcional a la misma cantidad de infectados y de ahí el término de recuperación. Por supuesto, es posible justificar esto de una manera más cuantitativa y precisa (para ver una derivación de estas ecuaciones consúltese Ref. [34]).

A pesar de su simplicidad, este modelo (3.1 - 3.3) no puede resolverse explícitamente. Es decir, no puede hallarse una expresión analítica exacta para $I(t)$ y $S(t)$ que nos permita anticipar la cantidad de infectados que habrá a tiempo t dadas las condiciones iniciales $I(0) = I_0$ y $S(0) = S_0$ ¹. Por ello es necesario recurrir a métodos numéricos para resolverlo. En la figura 3.2 se puede ver la evolución temporal de las variables del modelo resuelto numéricamente usando los parámetros $\beta = 5/\text{semana}$ y $\gamma = 1/\text{semana}$ con condiciones iniciales $I_0 = 0.01$ y $S_0 = 0.99$. Se observa cómo la fracción de susceptibles decrece mientras la de infectados aumenta hasta llegar a un pico donde aproximadamente la mitad de la población está infectada. Luego los infectados comienzan a recuperarse y casi toda la población termina en la clase R , de modo que

¹En realidad, para ser precisos, en 2014 se encontró una solución analítica, en términos de una integral que debe resolverse numéricamente [35].

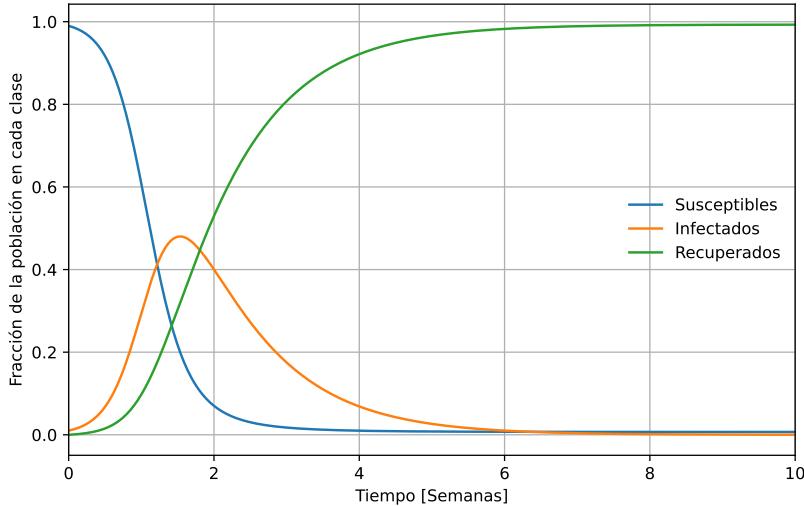


Figura 3.2: Evolución temporal de las variables del modelo resuelto numéricamente con $\beta = 5/\text{semana}$, $\gamma = 1/\text{semana}$, $I_0 = 0.01$ y $S_0 = 0.99$.

la mayoría de la población atravesó la enfermedad.

Es claro que la figura 3.2 no muestra toda la riqueza del sistema, ya que simplemente muestra la solución para un solo conjunto de parámetros y condiciones iniciales. Es decir, es de esperar que la dinámica difiera si por ejemplo la tasa de transmisión β es menor. En particular, es de interés saber qué conjunto de parámetros β y γ favorecen o dificultan el progreso de la infección. Esto puede determinarse de manera sencilla pidiendo que $\frac{dI}{dt} < 0$ al momento del brote de la infección. De esto resulta que si $S_0 < S_c = \gamma/\beta$ para cualquier $I_0 > 0$ entonces la infección no progresa. Este es un resultado conocido, obtenido por Kermack y McKendrick (1927).

El cociente γ/β es la tasa de recuperación relativa, sin embargo, es más habitual referirse a su recíproco, $R_0 = \beta/\gamma$ conocido en epidemiología comúnmente como el número de reproducción básico, el cual describe la media de personas infectadas por un individuo infectado durante su estadío infectivo. Dado que usualmente $S_0 \approx 1$, la condición para que la infección perezca se lee ahora en función de R_0 simplemente como $R_0 < 1$. Lo cual resulta natural: si un infectado infecta en promedio a menos de una persona en lo que cursa la enfermedad entonces la infección no se propaga.

En la figura 3.3 se puede ver la evolución temporal de las variables del modelo resuelto numéricamente con distintos números de reproducción básico R_0 . Se observa que en la medida que $R_0 \rightarrow 1$ la magnitud del máximo de infección va decreciendo, mientras que cuando $R_0 = 1$ la cantidad de infectados solo decrece en el tiempo hasta llegar a cero. Lo mismo sucede si $R_0 < 1$.

Es importante no confundir el número de reproducción básico R_0 con el número de reproducción efectivo $R_e(t)$. El primero considera que todos los individuos de la población son susceptibles, mientras que el segundo no y varía con el tiempo, se define como $R_e(t) = \beta S(t)/\gamma$. Este probablemente sea el indicador epidemiológico más usado.

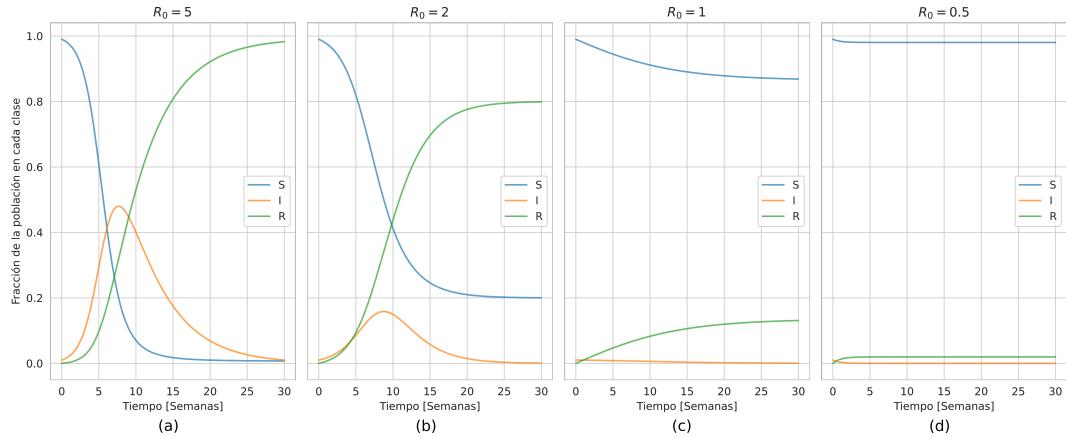


Figura 3.3: Evolución temporal de las variables del modelo resuelto numéricamente con (a) $R_0 = 5$, (b) $R_0 = 2$, (c) $R_0 = 1$ y (d) $R_0 = 0.5$.

De manera similar que con R_0 , se puede ver que si $R_e(t) < 1$ entonces la tasa de infección $\frac{dI}{dt}(t)$ es decreciente.

Otra cantidad de interés es la fracción de la población que no se contagió nunca. Es decir, cuánto vale el límite $\lim_{t \rightarrow \infty} S(t) \equiv S(\infty)$. De la figura 3.3 se puede ver que cuando $R_0 = 5$ la fracción de la población que no se contagia es muy cercana a cero, pero distinta a cero en fin. Mientras que cuando $R_0 = 2$, pareciera que $S(\infty) \rightarrow 0.2$. Para responder esta pregunta de manera más general podemos tomar la ecuación 3.1, dividirla por 3.3 y luego integrar para obtener una expresión de S en función de R . De esto resulta que,

$$S(t) = S_0 e^{-R_0 R(t)},$$

tomando el límite correspondiente, usando $\lim_{t \rightarrow \infty} R(t) \equiv R(\infty)$ y que $R(\infty) = 1 - S(\infty)$, resulta la siguiente ecuación trascendental para $S(\infty)$

$$\begin{aligned} S(\infty) &= S_0 e^{-R_0(1-S(\infty))}, \\ 0 &= S(\infty) - S_0 e^{-R_0(1-S(\infty))}. \end{aligned} \quad (3.4)$$

Puede verse que si $S(\infty) = S_0$, entonces el miembro derecho de 3.4 es positivo, mientras que si $S(\infty) = 0$, es negativo. De modo que la ecuación 3.4 tiene una solución para $S(\infty)$ entre 0 y 1. Es decir, sin importar qué tan grande sea el número de reproducción básico R_0 , o bien, qué tan contagiosa sea la enfermedad, siempre existe una fracción de la población que no se contagia.

Es importante señalar brevemente las virtudes y fundamentalmente las hipótesis bajo las que se presenta este modelo. La ventaja más notable es la simplicidad y el carácter didáctico del mismo, que como vimos, permite definir y asimilar conceptos generales asociados a la problemática de manera sencilla. Esta simplicidad, sin embargo, viene acompañada de hipótesis que en ocasiones resultan restrictivas y poco realistas

en lo que respecta a una dinámica tan compleja como la de una epidemia. Por ejemplo, se ignoran efectos de demografía los cuales pueden tener un impacto apreciable sobre la dinámica a escalas temporales extensas propias de una endemia. Además, se trata de un modelo de campo medio, donde se asume que cada sujeto de la población interactúa con todos los demás, es decir, desprecia heterogeneidades que puedan surgir de la edad, el espacio, la movilidad o aspectos de comportamiento. Adicionalmente, supone que individuos que pasaron por la enfermedad adquieren inmunidad para toda la vida y que un sujeto inmediatamente infectado puede infectar a otro. Por supuesto, esto no desmerece en nada al modelo, el cual sigue siendo extremadamente útil como primera aproximación al modelado de este tipo de sistemas complejos, simplemente es importante recordar las hipótesis sobre las que se trabaja para evitar posibles confusiones.

Por último, es de interés mencionar que si bien el enfoque ha estado hasta ahora centrado en una descripción epidemiológica, es posible extender este mismo modelo de manera sencilla a otras problemáticas. En particular, puede asociarse rápidamente el modelo SIR con la dinámica de un incendio en un bosque, donde los «susceptibles» son los árboles que pueden incendiarse, los «infectados» son los árboles en llamas y los «recuperados» los árboles que ya han sido quemados y no pueden volver a incendiarse [4, 5, 24, 36]. O bien, puede asociarse con la dinámica de una reacción química, donde los «susceptibles» son los reactantes, los «infectados» los productos intermedios y los «recuperados» los productos finales [37–39].

3.3. Modelo SIR de reacción-difusión homogéneo

De manera general las ecuaciones de reacción-difusión sobre un medio isotrópico son aquellas que pueden escribirse como

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{f} + \nabla \cdot (D \nabla \mathbf{u}), \quad (3.5)$$

donde \mathbf{u} es un campo vectorial que depende de la posición \mathbf{x} y el tiempo t , \mathbf{f} es el término de reacción, que es función de \mathbf{u} , \mathbf{x} y t , mientras que $\nabla \cdot (D \nabla \mathbf{u})$ es el término de difusión, donde D es la matriz de difusión que puede ser función de \mathbf{x} . Este tipo de ecuaciones han sido ampliamente estudiadas [29, 34, 40, 41] y se denominan así porque originalmente se utilizaron para estudiar la dinámica de reactivos químicos [42].

En lo que respecta al modelo SIR de reacción-difusión que nos interesa a nosotros, este puede escribirse de manera sencilla agregando el término difusivo a las ecuaciones

(3.1 - 3.3), dejando de lado la ecuación para R , esto es

$$\partial_t S = -\beta SI + D_S \nabla^2 S, \quad (3.6)$$

$$\partial_t I = \beta SI - \gamma I + D_I \nabla^2 I. \quad (3.7)$$

Donde ahora S , I y R son funciones de la posición (x, y) en un espacio bidimensional además del tiempo t , de modo que ahora se cumple $S(x, y, t) + I(x, y, t) + R(x, y, t) = 1$ para todo (x, y, t) . En este nuevo modelo (6.1 - 6.2) los términos difusivos dan lugar a una transmisión local de la infección. Es decir, abandonamos el modelo de campo medio que teníamos en la sección 3.2 donde todos los individuos podían interactuar entre sí a un instante dado.

Hemos supuesto que los coeficientes de difusión D_S y D_I son independientes de la posición y que la matriz D es diagonal dejando de lado la posibilidad de difusión cruzada. Adicionalmente, tanto β como γ son independientes de la posición, dando lugar a un medio totalmente homogéneo, es decir que todos los puntos del espacio son equivalentes en términos de transmisión. Esta es una característica crítica a remarcar, ya que en la sección 3.4 presentamos el correspondiente modelo heterogéneo, donde el medio puede adquirir un carácter desordenado, que es el foco de estudio de este trabajo.

A continuación se muestran algunos resultados interesantes asociados a este modelo que serán de interés a la hora de compararlo con su versión heterogénea.

3.3.1. Soluciones de onda

Dada una fracción de infectados inicial $I(x, y, 0)$ y una fracción de susceptibles distribuida homogéneamente $S(x, y, 0) = S_0$, se quiere saber cómo es la evolución espacio-temporal de la fracción de infectados $I(x, y, t)$. En la problemática de incendios la idea sería la misma, pero cambiando infectados por, digamos, incendiados.

Nuevamente, no es posible resolver las ecuaciones 6.1 y 6.2 de manera exacta, sin embargo, es posible estudiar qué condiciones deben satisfacerse para que cierto tipo de soluciones puedan existir. En particular, nos interesa estudiar bajo qué condiciones podría existir una solución de onda y qué características tendría.

Para ello proponemos una solución de onda plana (onda solitaria o frente) donde

$$S(x, y, t) = S(z), \quad I(x, y, t) = I(z), \quad z = x - ct, \quad (3.8)$$

que representa una onda de infección viajando en la dirección x positiva con una velocidad $c > 0$. Reemplazando 3.8 en 6.1 y 6.2, resulta el siguiente sistema de ecuaciones no lineales,

$$D_S S'' + cS' - \beta IS = 0, \quad (3.9)$$

$$D_I I'' + cI' + \beta I(S - \gamma/\beta) = 0, \quad (3.10)$$

donde las primas indican derivadas respecto de z . Como es habitual, este sistema tampoco puede resolverse explícitamente, sin embargo, imponiendo las siguientes condiciones para las soluciones²

$$I(\pm\infty) = 0, \quad S(\infty) = S_0, \quad S(-\infty) = S_1,$$

donde S_1 sería la fracción de susceptibles que deja la onda por detrás, es posible linearizar la ecuación 3.10 para z donde $S(z) \approx S_0$, es decir, sobre el perfil frontal de la onda. De lo cual resulta,

$$D_I I'' + cI' + \beta I(S_0 - \gamma/\beta) = 0, \quad (3.11)$$

que tiene una solución $I(z) \propto e^{-\lambda z}$, con λ satisfaciendo,

$$D_I \lambda^2 - c\lambda + \beta(S_0 - \gamma/\beta) = 0,$$

es decir,

$$\lambda = \frac{c}{2D_I} \pm \sqrt{\left(\frac{c}{2D_I}\right)^2 - \frac{\beta}{D_I}(S_0 - \gamma/\beta)}.$$

Debemos imponer además que $\lambda \in \mathbb{R}$ con $\lambda > 0$, de otra manera la solución no sería autoconsistente. Al imponer que $\lambda > 0$ resulta $\gamma/\beta < S_0$, que es la misma condición que habíamos obtenido en la sección 3.2 para que la infección progrese, mientras que ahora es una condición necesaria para la existencia de soluciones onda, las cuales darían lugar a la propagación de la infección. Más interesante quizás, es la condición $\lambda \in \mathbb{R}$, de la cual resulta que

$$0 \leq \left(\frac{c}{2D_I}\right)^2 - \frac{\beta}{D_I}(S_0 - \gamma/\beta),$$

$$c \geq 2\sqrt{D_I(\beta S_0 - \gamma)} \equiv c_0, \quad (3.12)$$

dando así una velocidad mínima c_0 para la existencia de la onda. Veremos en el capítulo

²Se utiliza la siguiente notación por simplicidad, dada una función $f(z)$,

$$\lim_{z \rightarrow \pm\infty} f(z) \equiv f(\pm\infty)$$

5 que la velocidad de propagación es de hecho la mínima encontrada aquí $c = c_0$. Tomando esto por cierto, el perfil frontal de la onda de infección queda dado por

$$I_f(z) \propto \exp\left[-\frac{c_0}{2D_I}z\right]. \quad (3.13)$$

Haciendo una cuenta equivalente para el perfil posterior de la onda donde $S(z) \approx S_1$, hay que resolver la ecuación análoga a 3.11, $D_I I'' + cI' + \beta I(S_1 - \gamma/\beta) = 0$, de esto resulta

$$I_p(z) \propto \exp\left[\left(-\frac{c_0}{2D_I} + \sqrt{(c_0/2D_I)^2 - \frac{\beta}{D_I}(S_1 - \gamma/\beta)}\right)z\right], \quad (3.14)$$

$$I_p(z) \propto \exp\left[\left(-\frac{c_0}{2D_I} + \sqrt{\frac{\beta}{D_I}(S_0 - S_1)}\right)z\right]. \quad (3.15)$$

Donde en el último paso se reemplazó la expresión para c_0 (ecuación 3.12) dentro de la raíz. De la ecuación 3.14 se ve que es necesario que $\gamma/\beta > S_1$, dado que debe satisfacerse $I_p(-\infty) = 0$, de modo que en resumen se tiene la siguiente relación

$$S_0 > \gamma/\beta > S_1 > 0,$$

es decir, que la fracción de susceptibles que quedan tras el paso de la onda no es suficiente para activar una onda de retroceso ya que $S_1 < \gamma/\beta$.

En resumen, se obtuvieron expresiones analíticas aproximadas del perfil frontal y posterior que tendría una solución de onda, ecuaciones 3.13 y 3.14 respectivamente, las cuales dan a entender que el perfil completo de la onda es asimétrico. Se determinó a su vez la velocidad de propagación del frente $c_0 = 2\sqrt{D_I(\beta S_0 - \gamma)}$ y se estableció la jerarquía $S_0 > \gamma/\beta > S_1 > 0$ para la existencia del mismo.

Por último, quería comentar brevemente acerca de la estabilidad de estas soluciones. Nótese que en el desarrollo hemos reducido la dimensionalidad del problema a una dimensión z y hemos mostrado el tipo de soluciones admisibles en esta situación. Sin embargo, estas soluciones solo tienen sentido si pensamos en una condición inicial plana para el campo de infectados. Sin embargo, es de interés generalizar un poco más y entender qué sucede si la condición no es plana o si presenta algunas perturbaciones iniciales. En este sentido, decimos que el frente es estable si bajo perturbaciones iniciales el frente tiende a uno plano, mientras que es inestable si no. Puede verse que esta estabilidad está directamente asociada al cociente entre los coeficientes de difusión de infectados y susceptibles $\frac{D_S}{D_I}$. Si $\frac{D_S}{D_I} < 1$ entonces el frente es estable, en otro caso es inestable. Nosotros ignoraremos por completo la difusión en susceptibles, es decir,

utilizaremos $D_S = 0$, por lo cuál siempre estaremos con frentes estables en este sentido. Se decidió así por simplicidad y porque no constituía un objetivo principal investigar los efectos con distintos parámetros de difusión. Sin embargo, parecía importante mencionarlo dado que no es un efecto trivial y puede incorporar nuevos fenómenos al problema. Para una discusión más detallada acerca de esto puede referirse a Ref. [39].

3.3.2. Soluciones en sistema de reacción-difusión-convección

Es posible agregar un término de convección al modelo presentado en la sección ???. Esto es,

$$\partial_t S = -\beta SI + D_S \nabla^2 S \quad (3.16)$$

$$\partial_t I = \beta SI + D_I \nabla^2 I - \mathbf{v} \cdot \nabla I \quad (3.17)$$

donde \mathbf{v} es un campo vectorial. En términos epidemiológicos \mathbf{v} podría interpretarse como un campo de migración de individuos infectados de un lugar a otro, indicando la dirección de migración en cada punto, en tanto que su magnitud sería la velocidad de migración. Pensando en incendios es más sencillo motivar este término de convección, pues se podría interpretar como el campo de viento.

Ahora bien, se puede ver que este modelo también tiene soluciones de onda cuando \mathbf{v} es constante. Sin perder generalidad podemos tomar $\mathbf{v} = (v, 0)$ con $v > 0$, de modo que la ecuación de evolución para I queda

$$\partial_t I = \beta SI + D_I \nabla^2 I - v \partial_x I. \quad (3.18)$$

Pasando las ecuaciones a un sistema de coordenadas que se mueve a velocidad v en x -positivo, es decir, $x' = x - vt$, se obtiene

$$\partial_t I = \beta SI + D_I \nabla^2 I, \quad (3.19)$$

es decir, recuperamos el sistema de reacción-difusión que estudiamos antes. Tomando el desarrollo de la sección anterior se obtiene en este caso que la velocidad de propagación del frente de infección es

$$c = c_0 + v,$$

mientras que si tomamos viento en contra $\mathbf{v} = (-v, 0)$ se tiene,

$$c = c_0 - v.$$

Lo interesante del caso con viento en contra es que debemos pedir que $c > 0$, dado que buscamos una solución de onda viajando en la dirección x -positivo, de lo cual surge un

nuevo valor crítico para el umbral de propagación. Es decir,

$$\begin{aligned} c &> 0 \\ 2\sqrt{D_I\beta(S_0 - \gamma/\beta)} - v &> 0 \\ \sqrt{D_I\beta(S_0 - \gamma/\beta)} &> v/2 \\ \beta &> \frac{1}{S_0} \left(\frac{v^2}{4D_I} + \gamma \right) \equiv \beta_c \end{aligned} \quad (3.20)$$

donde β_c corresponde al nuevo umbral de propagación dados (γ, D_I, v, S_0) con viento en contra. En contraste, el valor umbral de β con viento a favor, que corresponde al mismo que en el caso sin viento, es $\beta_c = \frac{\gamma}{S_0}$. En resumen, dado el conjunto de parámetros (γ, D_I, v, S_0) , la mínima tasa de transmisión necesaria para la existencia de una solución de onda viajando en la dirección x -positivo es

$$\beta_c = \begin{cases} \frac{\gamma}{S_0}, & \text{si } v \geq 0 \\ \frac{1}{S_0} \left(\frac{v^2}{4D_I} + \gamma \right), & \text{si } v < 0. \end{cases} \quad (3.21)$$

3.4. Modelo SIR de reacción-difusión-convección heterogéneo

En esta sección presentamos el modelo completo que estudiamos con profundidad en este trabajo junto a algunas descripciones estadísticas del mismo que se usarán posteriormente.

Esencialmente las ecuaciones son las mismas que 3.16 y 3.17 con la salvedad de que ahora introducimos heterogeneidad en el medio poniendo una tasa de transmisión β_r dependiente de la posición. Por completitud escribimos las ecuaciones nuevamente aquí,

$$\frac{\partial S}{\partial t} = -\beta_r SI + D_S \nabla^2 S, \quad (3.22)$$

$$\frac{\partial I}{\partial t} = \beta_r SI - \gamma I + D_I \nabla^2 I - \mathbf{v} \cdot \nabla I. \quad (3.23)$$

Una tasa de transmisión β_r de este tipo podría utilizarse, por ejemplo, para estudiar el efecto de la vacunación sobre la población. Esto es entendible dado que se espera que los lugares donde hay población vacunada la tasa de transmisión sea menor ya que hay menos individuos susceptibles para infectarse. Otra alternativa apuntaría a describir lugares donde la tasa de transmisión es más baja por un efecto de densidad poblacional o lugares donde se toman medidas de prevención. En cuanto a la propagación de incendios

podría interpretarse de manera similar como una variación de la densidad o el tipo de vegetación en el espacio, lo cual facilitaría o no la transmisión de las llamas.

3.5. Problema y observables

Vamos a estudiar el frente de infección/incendio en $(x, y, t) \in \Omega \times [0, \infty)$, con $\Omega = [0, L_x] \times [0, L_y]$. Sea la región donde comienza la infección/incendio $\Omega_0 = (0, \delta x) \times (0, L_y)$ donde $\delta x \in (0, L_x)$, entonces la condición inicial con la que trabajamos es

$$I(x, y, 0) = \begin{cases} I_0 & \text{si } (x, y) \in \Omega_0 \\ 0 & \text{si } (x, y) \notin \Omega_0 \end{cases}; \quad S(x, y, 0) = \begin{cases} 1 - I_0 & \text{si } (x, y) \in \Omega_0 \\ S_0 & \text{si } (x, y) \notin \Omega_0. \end{cases}$$

Mientras que las condiciones de contorno son de Dirichlet en la dirección x

$$I(0, y, t) = I(L_x, y, t) = S(0, y, t) = S(L_x, y, t) = 0,$$

y periódicas en la dirección y ,

$$I(x, 0, t) = I(x, L_y, t) \quad S(x, 0, t) = S(x, L_y, t).$$

Estas condiciones iniciales y de contorno dan lugar a un único frente de onda propagándose en la dirección x , lo cual resulta conveniente para realizar un análisis estadístico a partir de ciertos observables, los cuales se definen a continuación.

Para caracterizar las fluctuaciones espaciales y temporales del frente de infección/incendio, definimos primero el campo de desplazamiento del frente $u(y, t)$ como

$$\max_{x \in (0, L)} \{I(x, y, t)\} = I(u(y, t), y, t). \quad (3.24)$$

Es decir, $u(y, t)$ es la posición en x que maximiza la fracción de infectados/incendiados para un dado y en el instante t . Se define a su vez el centro de masa de $u(y, t)$,

$$u_{cm}(t) \equiv \langle u(y, t) \rangle_y, \quad (3.25)$$

donde $\langle \dots \rangle_y$ indica el promedio sobre la coordenada y . Tomamos la amplitud de $I(x, y, t)$ como el valor medio de $I(x, y, t)$ sobre $u(y, t)$, es decir,

$$I_{max}(t) = \langle I(u(y, t), y, t) \rangle_y. \quad (3.26)$$

Por otro lado, la velocidad del frente se define como

$$c(t) \equiv \dot{u}_{cm}(t). \quad (3.27)$$

Mientras que las fluctuaciones del frente de onda pueden ser caracterizadas definiendo el ancho del mismo $\omega(t)$ como la desviación estándar de $u(y, t)$,

$$w(t)^2 \equiv \langle [u(y, t) - u_{cm}(t)]^2 \rangle_y, \quad (3.28)$$

o bien utilizando el factor de estructura,

$$S(q, t) \equiv |\tilde{u}(q, t)|^2, \quad (3.29)$$

donde $\tilde{u}(q, t)$ es la transformada de Fourier sobre el espacio de $u(y, t)$.

En capítulos posteriores utilizaremos los observables definidos aquí como principales herramientas para caracterizar y comparar de manera cuantitativa los efectos que tienen distintas heterogeneidades, introducidas mediante β_r , sobre la dinámica del problema. A continuación describimos los distintos tipos de heterogeneidades estudiadas.

Tipos de heterogeneidades

Heterogeneidad dicotómica-aleatoria (DA)

Consiste en una tasa de transmisión β_r espacialmente dicotómica con una distribución de probabilidad dada por

$$f(\beta_r) = p\delta(\beta_r) + (1-p)\delta(\beta_r - \beta), \quad (3.30)$$

donde $0 \leq p \leq 1$. Esto es equivalente a decir, dada una posición $\mathbf{r} = (x, y)$, la tasa de transmisión en \mathbf{r} es 0 con probabilidad p o es β con probabilidad $1 - p$. Nótese que cuando $p = 0$ se recupera el caso homogéneo, donde la tasa de transmisión es β en todo el espacio. Con la distribución de probabilidad 3.30, este β_r cumple

$$\overline{\beta_r} = (1-p)\beta, \quad (3.31)$$

$$\overline{\beta_r \beta_{r'}} - \overline{\beta_r} \overline{\beta_{r'}} = \beta^2 p(1-p)\delta(\mathbf{r} - \mathbf{r}'), \quad (3.32)$$

donde las barras indican valor medio sobre desorden. Puede verse que no se tiene correlación espacial, por tanto se estaría representando una estrategia de vacunación aleatoria sobre la población. Llamaremos a la heterogeneidad definida así como «heterogeneidad dicotómica-aleatoria» o DA simplemente.

Para ganar cierta intuición sobre lo que implica este tipo de heterogeneidad sobre

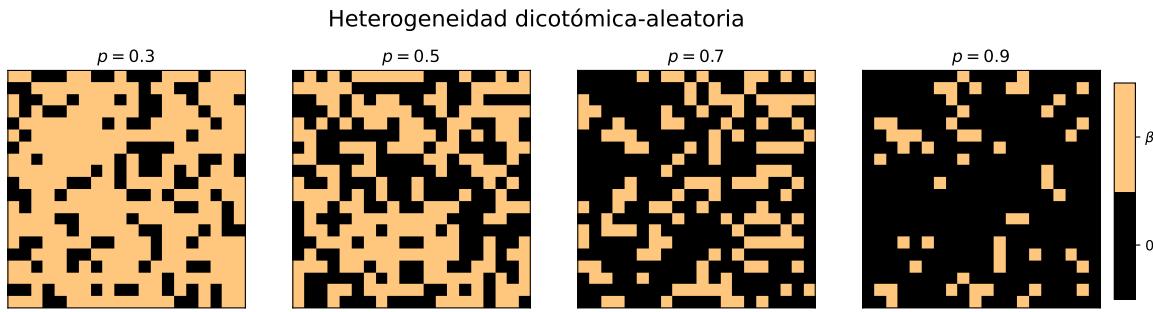


Figura 3.4: Representación de β_r sobre una grilla de 20×20 para $p = 0.3$, $p = 0.5$, $p = 0.7$ y $p = 0.9$.

el espacio, en la figura 3.4 pueden verse distintas realizaciones de β_r para diferentes valores de p representadas sobre una grilla de 20×20 .

Heterogeneidad suavizada (S)

En este caso se propone una heterogeneidad suavizada, donde, en primera instancia, se genera un β_r usando la distribución de probabilidad 3.30 y luego se aplica una regla de suavizado n veces. En un esquema discreto donde el espacio de $L \times L$ se descompone en $N \times N$ cuadrantes de L/N por L/N , la regla de suavizado puede describirse de manera recursiva como,

$$\beta_r^{(0)} = \beta_r^{DA}, \quad (3.33)$$

$$\beta_{i,j}^{(n)} = \frac{1}{5}(\beta_{i-1,j}^{(n-1)} + \beta_{i+1,j}^{(n-1)} + \beta_{i,j-1}^{(n-1)} + \beta_{i,j+1}^{(n-1)} + \beta_{i,j}^{(n-1)}), \quad (3.34)$$

donde $\beta_{i,j}^{(n)}$ es el valor del nuevo $\beta_r^{(n)}$ en el cuadrante $\mathbf{r} = (i, j)$ con $i, j = 1, 2, \dots, N$.³

La particularidad de esta tasa de transmisión es que ahora presenta una correlación espacial local. Esto implica que disminuirán los efectos de cambios abruptos en la tasa de transmisión sobre la propagación del frente de onda. Por otro lado, el valor medio de la tasa de transmisión del medio S con n pasos $\overline{\beta_r^{(n)}}$ es igual al valor medio de la tasa de transmisión del medio DA que le da origen $\overline{\beta_r^{DA}}$. Esto puede verse por inducción, $\overline{\beta_r^{(1)}} = \overline{\beta_r^{(0)}} = \overline{\beta_r^{DA}}$, luego, asumiendo que $\overline{\beta_r^{(n-1)}} = \overline{\beta_r^{DA}}$ resulta que $\overline{\beta_r^{(n)}} = \overline{\beta_r^{DA}}$.

En la figura 3.5 puede verse esquemáticamente el efecto de un paso de suavizado sobre una grilla de 10×10 , mientras que en la figura 3.6 se muestran realizaciones de $\beta_r^{(1)}$ para diferentes p sobre una grilla de 20×20 , compare con la figura 3.4.

Un detalle más a notar sobre este tipo de heterogeneidad es que para $n \rightarrow \infty$ recuperamos el caso homogéneo, con una tasa de transmisión constante de valor $\overline{\beta_r} = (1-p)\beta$. Esto puede verse tomando el límite en la ecuación 3.34 para $n \rightarrow \infty$, poniendo

³Se consideran condiciones periódicas sobre la grilla, de modo que por ejemplo, $i = N + 1$ se entiende como $i = 1$.

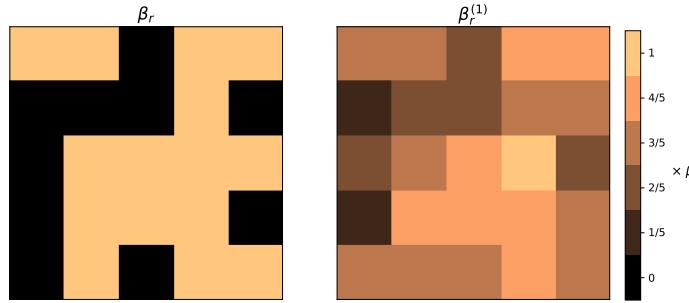


Figura 3.5: A la izquierda, se muestra una realización de β_r^{DA} con $p = 0.5$. A la derecha, se muestra el efecto de un paso de suavizado sobre el β_r^{DA} original generando $\beta_r^{(1)}$.

$\lim_{n \rightarrow \infty} \beta_{i,j}^{(n)} \equiv \beta_{i,j}^{(\infty)}$, se tiene

$$4\beta_{i,j}^{(\infty)} = \beta_{i-1,j}^{(\infty)} + \beta_{i+1,j}^{(\infty)} + \beta_{i,j-1}^{(\infty)} + \beta_{i,j+1}^{(\infty)},$$

para todo (i, j) , de modo que la única posibilidad es que $\beta_{i,j}^{(\infty)}$ sea el mismo para todo (i, j) e igual a $\bar{\beta}_r$, ya que debe conservar el valor medio. En la figura 3.7 se muestra el efecto al realizar varios pasos de suavizado.

Heterogeneidad dicotómica-correlacionada (DC)

Esta consiste, de alguna manera, en una combinación de las dos anteriores. Se genera un $\beta_r^{(n)}$ de la misma manera que para la versión suavizada. Una vez obtenido

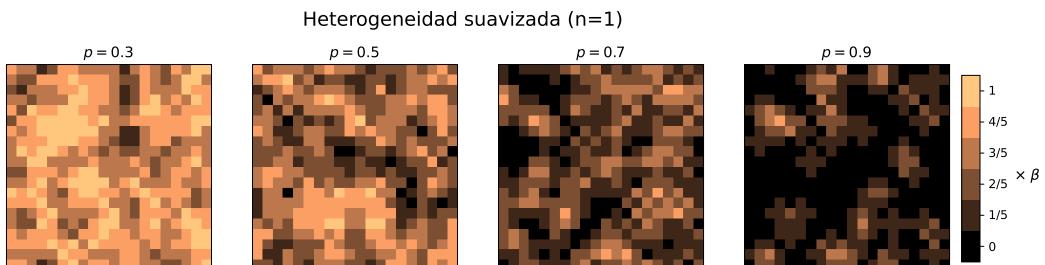


Figura 3.6: Representación de $\beta_r^{(1)}$ sobre una grilla de 20×20 para $p = 0.3$, $p = 0.5$, $p = 0.7$ y $p = 0.9$.

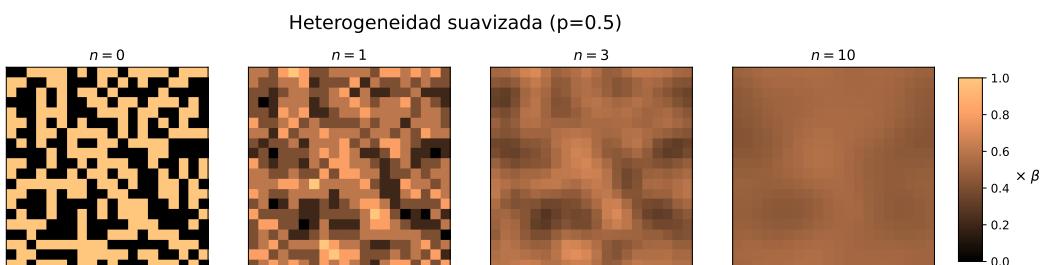


Figura 3.7: Se muestran los $\beta_r^{(n)}$ con $n = 0, 1, 3, 10$ sobre una grilla de 20×20 .

este $\beta_{\mathbf{r}}^{(n)}$ definimos un nuevo $\tilde{\beta}_{\mathbf{r}}^{(n)}$ de la siguiente manera,

$$\begin{aligned}\tilde{\beta}_{i,j}^{(n)} &= \beta, & \text{si } \beta_{i,j}^{(n)} \geq \bar{\beta}_{\mathbf{r}} = (1-p)\beta, \\ \tilde{\beta}_{i,j}^{(n)} &= 0, & \text{si } \beta_{i,j}^{(n)} < \bar{\beta}_{\mathbf{r}} = (1-p)\beta.\end{aligned}$$

Este caso presenta correlación local en una versión dicotómica, pero naturalmente no conserva el valor medio. En la figura 3.8 se muestran realizaciones para distintos valores de p con $n = 1$. Puede observarse a simple vista que la densidad de puntos negros no aumenta con p como pasaba en los casos anteriores, esto quiere decir que p no puede interpretarse como la fracción de lugares donde β se anula para este caso. Esta diferencia sustancial debe tenerse en cuenta a la hora de comparar resultados entre heterogeneidades. Para ello utilizaremos el valor medio espacial de la tasa de transmisión $\langle \beta_{\mathbf{r}} \rangle_{\mathbf{r}} = \beta_m$ en lugar de p , ya que da cuenta de esta variación del comportamiento.

En la figura 3.9 se muestra el efecto de los pasos de suavizado para este caso con $p = 0.5$. En este caso, al hacer el límite $n \rightarrow \infty$ se recupera nuevamente el caso homogéneo pero con un valor medio de $\tilde{\beta}_{\mathbf{r}}^{(\infty)}$ igual a β .

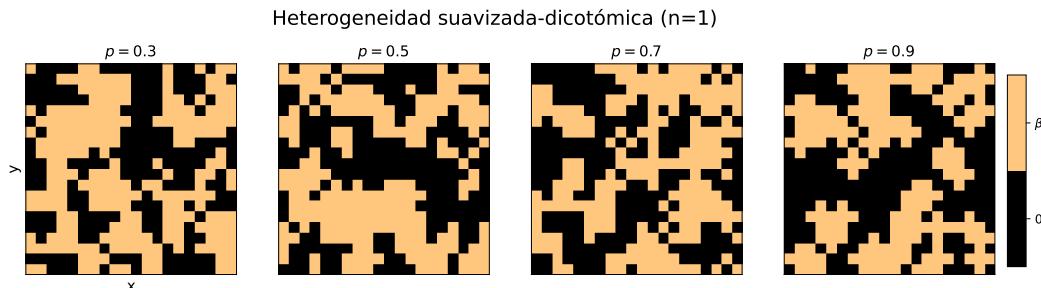


Figura 3.8: Representación de $\tilde{\beta}_{\mathbf{r}}^{(1)}$ sobre una grilla de 20×20 para $p = 0.3$, $p = 0.5$, $p = 0.7$ y $p = 0.9$.

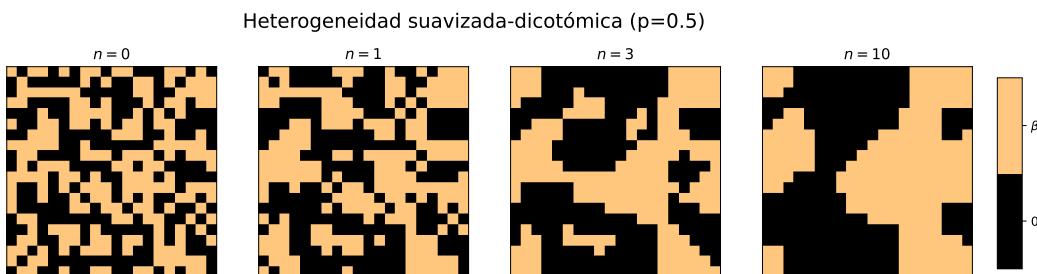


Figura 3.9: Se muestran los $\tilde{\beta}_{\mathbf{r}}^{(n)}$ con $n = 0, 1, 3, 10$ sobre una grilla de 20×20 .

Capítulo 4

Reseña numérica y computacional

La idea de este capítulo es dejar documentación que muestre con claridad el trabajo de investigación y elaboración técnico, a nivel computacional, realizado para llevar a cabo este proyecto. Adicionalmente, está en mi intención ser lo más genérico, claro y acotado posible, para que sea de utilidad a cualquier otra persona que esté interesada en resolver ecuaciones diferenciales parciales haciendo uso de programación en paralelo. En particular, usando *CUDA* [43] a través de las bondades que ofrece *Python* [44] mediante la librería *CuPy* [45].

Este capítulo cuenta con un material complementario en formato de *Jupyter Notebook* en *Google Colab* al cual puede acceder desde [aquí](#). *Google Colab* da acceso gratuito a GPUs, lo cual está fantástico para aprender a usarlas, aunque evidentemente es con tiempo limitado. En este *Jupyter Notebook* esencialmente encontrará todo el código presentado aquí y un poco más, para que pueda interactuar y hacer las modificaciones que quiera.

A continuación dejamos constancia del *software* y *hardware* utilizado en la ejecución de los códigos de este capítulo.

- Python: 3.9.7
- CUDA Version: 11.6
- CPU: AMD Ryzen 9 5900HX
- GPU: NVIDIA GeForce RTX 3060 Laptop
- Memoria de GPU: 6GB
- CUDA cores: 3840
- RAM: 32GB

4.1. Diferencias finitas

El objetivo es resolver numéricamente ecuaciones diferenciales parciales de la manera más simple y eficiente posible. Para la física este tipo de ecuaciones son de gran interés, ya que se usan ampliamente para modelar todo tipo de fenómenos.

Para reducir la complejidad del problema, acotamos la discusión mostrando en detalle el proceso de resolución de un sistema de dos ecuaciones de reacción-difusión con dos dimensiones espaciales (x, y) en cierta región $\Omega \in \mathbb{R}^2$. Esto es conveniente porque este problema corresponde con el tipo de sistemas usados en este trabajo. Explícitamente, queremos resolver,

$$\begin{aligned}\partial_t u &= f_u(u, v) + D_u \nabla^2 u \\ \partial_t v &= f_v(u, v) + D_v \nabla^2 v,\end{aligned}\tag{4.1}$$

donde u y v son las variables dinámicas que dependen de las variables (x, y, t) , f_u y f_v son funciones suaves, mientras que D_u y D_v son los coeficientes de difusión de u y v respectivamente. Estas ecuaciones deben resolverse teniendo en cuenta determinadas condiciones iniciales y de contorno para el problema en cuestión, podemos expresarlas genéricamente de la siguiente manera,

$$\begin{aligned}u(x, y, t = 0) &= g_u(x, y) \quad \forall x, y \in \Omega \quad | \quad u(x, y, t) = h_u(x, y, t) \quad \forall t \in \mathbb{R}; \forall x, y \in \delta\Omega \\ v(x, y, t = 0) &= g_v(x, y) \quad \forall x, y \in \Omega \quad | \quad v(x, y, t) = h_v(x, y, t) \quad \forall t \in \mathbb{R}; \forall x, y \in \delta\Omega.\end{aligned}$$

Donde las funciones g_u y g_v denotan las condiciones iniciales del sistema, las funciones h_u y h_v las condiciones de contorno y $\delta\Omega$ es el borde de Ω .

De esta manera queda completamente definido el problema y procedemos al armado del esquema numérico necesario para resolverlo. Por simplicidad consideramos que Ω es una región rectangular del plano donde $x, y \in [0, L_x] \times [0, L_y]$. Segmentamos este espacio en $N_x \times N_y$ cuadrantes de dimensiones $d_x = L_x/N_x$ y $d_y = L_y/N_y$ y denominamos u_{ij} y v_{ij} a los valores de las funciones u y v a tiempo t en el cuadrante (i, j) correspondiente a la región $[jd_x, (j+1)d_x) \times [id_y, (i+1)d_y)$, con $i = 0, 1, \dots, N_y - 1$ y $j = 0, 1, \dots, N_x - 1$.

A continuación, la idea consiste en reducir el sistema de ecuaciones diferenciales parciales 4.1 en un sistema de ecuaciones diferenciales ordinarias, donde cada ecuación describe la evolución de las variables dinámicas en un cuadrante distinto. Para ello necesitamos llevar los laplacianos de las ecuaciones al nuevo esquema espacial discretizado. Lo hacemos usando la siguiente aproximación por diferencias finitas para los laplacianos,

$$\begin{aligned} (\nabla^2 u)_{ij} &= \frac{1}{d^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}) \\ (\nabla^2 v)_{ij} &= \frac{1}{d^2}(v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{ij}) \end{aligned} \quad (4.2)$$

donde usamos que $d = d_x = d_y$. Utilizando la notación dada para la discretización espacial, tenemos el siguiente sistema de ecuaciones diferenciales ordinarias,

$$\begin{aligned} \frac{du_{ij}}{dt} &= f_u(u_{ij}, v_{ij}) + D_u(\nabla^2 u)_{ij} \\ \frac{dv_{ij}}{dt} &= f_v(u_{ij}, v_{ij}) + D_v(\nabla^2 v)_{ij}. \end{aligned} \quad (4.3)$$

Finalmente, discretizamos el espacio temporal con un intervalo dt y aproximamos la derivada temporal a primer orden. Usando $n \in \mathbb{N}$ como índice temporal, notamos u_{ij}^n y v_{ij}^n como el valor de las funciones u y v en el instante $t = n * dt$ sobre el cuadrante (i, j) . De esta manera obtenemos el siguiente esquema explícito de Euler para la resolución numérica del sistema 4.1,

$$\begin{aligned} u_{ij}^{n+1} &= u_{ij}^n + dt (f_u(u_{ij}^n, v_{ij}^n) + D_u(\nabla^2 u)_{ij}^n) \\ v_{ij}^{n+1} &= v_{ij}^n + dt (f_v(u_{ij}^n, v_{ij}^n) + D_v(\nabla^2 v)_{ij}^n), \end{aligned} \quad (4.4)$$

a partir del cual podemos iterar para obtener la solución. En la siguiente sección se describe cómo hacer esto usando *Python* con diferentes niveles de eficiencia.

4.2. Implementaciones en *Python*

En esta sección se describen brevemente 4 implementaciones distintas para la resolución del esquema numérico 4.4 hallado en la sección anterior. Comenzamos con una implementación usando la librería *NumPy* [46] en un esquema serial, luego vamos optimizando las implementaciones hasta llegar a la más eficiente obtenida, correspondiente a la utilización de *CUDA* a través de la librería *CuPy*.

La razón para hacer esto reside nuevamente en la idea de dejar documentación que pueda llegar a ser de utilidad para alguien más lidiando con problemas numéricos donde la eficiencia de la resolución es relevante. Además, antes de llegar a la versión final, donde usamos *CUDA* y consecuentemente es necesario un procesador gráfico (GPU) compatible con *CUDA*, se muestran implementaciones que son notablemente eficientes sin necesidad de una GPU, y que pueden ser usadas por una computadora más modesta.

4.2.1. Implementación con *NumPy*

NumPy es una de las librerías fundamentales para hacer computación científica en *Python*. Ofrece los invaluables *NumPy Arrays*, que están diseñados específicamente para ser lo más eficiente posible en la manipulación numérica sin perder la simplicidad y elegancia de *Python*. Utilizando *NumPy* se obtienen mejores resultados que usando *Python* nativo porque integra código de *C, C++ y Fortran* dentro de *Python*, adicionalmente la mayoría de los métodos implementados para la operación con *NumPy arrays* están paralelizados para correr en varios núcleos de la CPU a la vez.

El código 4.1 muestra la implementación propuesta usando *NumPy*. Nótese que hay más líneas de comentarios que de código. Esencialmente, primero definimos una función que llamamos *laplacian*, que toma un *array* 2D y devuelve su laplaciano, para ello usamos la función *roll* de *NumPy* que desplaza un *array* sobre un eje dado cuanto queramos y con condiciones periódicas. Luego, sumamos y restamos estas matrices desplazadas según 4.2 y obtenemos el laplaciano. Finalmente, definimos la función *cpu_numpy_solver*, que toma por argumentos las condiciones iniciales del problema, las funciones de reacción *f_u* y *f_v*, los coeficientes de difusión, el intervalo de integración temporal *dt* y espacial *d* y la cantidad de iteraciones *it*.

```

1 import numpy as np
2
3 def laplacian(X):
4     """
5         Take the Laplacian of a 2D array with periodic boundary conditions.
6     """
7     return np.roll(X,1,axis = 0) + np.roll(X,-1,axis = 0) + np.roll(X,1,axis = 1) + np.roll(X
8         ,-1,axis = 1) - 4*X
9
10 def cpu_numpy_solver(u, v, fu, fv, Ds, dt = .01, d = 1, it = 1000):
11     """
12         Solve a 2D reaction-diffusion equation for two dynamical variables with periodic boundary
13         conditions using NumPy.
14
15         u: Intial conditions for the first dynamical variable. 2d NumPy array.
16         v: Intial conditions for the second dynamical variable. 2d NumPy array.
17         fu: Reaction term for the first dynamical variable. Function.
18         fv: Reaction term for the second dynamical variable. Function.
19         Ds: Diffusion coefficients for the first and second dynamical variables. List or array of
20             length 2.
21         dt: Time step. Default value is 0.01.
22         d: Space step. Default value is 1.
23         it: Number of iterations. Default value is 1000.
24     """
25
26     Du,Dv = Ds
27     for _ in range(it):
28         u = u + dt*(fu(u,v) + Du*laplacian(u)/d**2)
29         v = v + dt*(fv(u,v) + Dv*laplacian(v)/d**2)
30
31     return u,v

```

Código 4.1: Implementación con *NumPy*.

```

1 def fu(u,v,beta):
2     return -beta*u*v
3
4 def fv(u,v,beta,gamma):
5     return beta*u*v - gamma*v
6
7 L = 1024
8 beta = 1
9 gamma = .2
10 u = np.ones((L,L))
11 v = np.zeros((L,L))
12 u[:,0] = 0
13 v[:,0] = 1
14 Ds = [0,1]
15
16 uf,vf = cpu_numpy_solver(u,v,fu,fv,Ds)

```

Código 4.2: Ejemplo de uso de la implementación con *NumPy*.

En el código 4.2 se muestra un ejemplo de uso utilizando las siguientes funciones de reacción,

$$\begin{aligned} f_u(u, v) &= -\beta uv, \\ f_v(u, v) &= \beta uv - \gamma v, \end{aligned} \tag{4.5}$$

donde β y γ son constantes.

Finalmente, a continuación se muestra la velocidad de la resolución con un sistema de 1024×1024 utilizando todos los parámetros y funciones del ejemplo 4.2. Para una justa comparación con las demás implementaciones, utilizaremos exactamente los mismos parámetros y funciones.

```

1 %timeit cpu_numpy_solver(u,v,fu,fv,Ds)
2 1min +- 1.53 s per loop (mean +- std. dev. of 7 runs, 1 loop each)

```

4.2.2. Implementación serial con *Numba*

Otra de las librerías fundamentales para cálculo numérico en *Python* es *Numba* [47]. *Numba* toma código nativo de *Python* y genera código de máquina optimizado usando *LLVM compiler infrastructure* [48]. También es capaz de procesar *NumPy* con una gran cantidad de sus métodos.¹

Lo mejor de todo esto es que *Numba* lo hace de manera completamente autónoma, solo hay que decirle que lo haga. Para ello utilizamos el decorador `@njit`, que indica a *Numba* que la función en cuestión debe ser procesada. El código 4.3 muestra cómo sería la implementación en este caso. Se define la función `cpu_numba_solver` que toma los mismos argumentos que la función `cpu_numpy_solver` vista anteriormente. Dentro

¹Para ver más detalles consultar la [documentación de Numba](#).

de la función iteramos temporalmente y para calcular los laplacianos y términos de reacción en cada cuadrante recorremos las matrices con un ciclo *for*. Usualmente es posible tomar las funciones o implementaciones realizadas con *Numpy* y decorarlas con *@njit* para obtener el resultado deseado. Sin embargo, en este caso no podemos hacerlo con las funciones del código 4.1 porque *Numba* no soporta la función *roll* de *NumPy*. Por lo cual estamos forzados a calcular el laplaciano de una manera más explícita para que *Numba* lo entienda.

```

1 from numba import njit
2 import numpy as np
3
4 @njit()
5 def cpu_numba_solver(u, v, fu, fv, Ds, dt=.01, d = 1, it = 1000):
6     """
7         Solve a 2D reaction-diffusion equation for two dynamical variables with periodic boundary
8             conditions using Numba
9             in a serial implementation.
10
11    u: Intial conditions for the first dynamical variable. 2d NumPy array of shape (Ly,Lx).
12    v: Intial conditions for the second dynamical variable. 2d NumPy array of shape (Ly,Lx).
13    fu: Reaction term for the first dynamical variable. Function.
14    fv: Reaction term for the second dynamical variable. Function.
15    Ds: Diffusion coefficients for the first and second dynamical variables. List or array of
16        length 2.
17    dt: Time step. Default value is 0.01.
18    d: Space step. Default value is 1.
19    it: Number of iterations. Default value is 1000.
20    """
21
22    Ly, Lx = u.shape
23    u = u.reshape(Ly*Lx)
24    v = v.reshape(Ly*Lx)
25    Fu = np.zeros_like(u)
26    Fv = np.zeros_like(v)
27    Du, Dv = Ds
28    for _ in range(it):
29        for i in range(Lx*Ly):
30            x = i % Lx
31            y = i // Lx
32            Lu = (u[(x+1)%Lx + Lx*y] + u[(x-1+Lx)%Lx+Lx*y] + u[x + Lx*((y+1)%Ly)] + u[x + Lx*((y-1+Ly)%Ly)] - 4*u[i])/d**2
33            Lv = (v[(x+1)%Lx + Lx*y] + v[(x-1+Lx)%Lx+Lx*y] + v[x + Lx*((y+1)%Ly)] + v[x + Lx*((y-1+Ly)%Ly)] - 4*v[i])/d**2
34            Fu[i] = fu(u[i],v[i]) + Du*Lu
35            Fv[i] = fv(u[i],v[i]) + Dv*Lv
36            u = u + dt*Fu
37            v = v + dt*Fv
38
39    return u.reshape(Ly,Lx), v.reshape(Ly,Lx)

```

Código 4.3: Implementación serial con *Numba*.

En cuanto al ejemplo de uso, sería idéntico al mostrado en 4.2, con la única salvedad de que las funciones f_u y f_v también deben estar decoradas con *@njit*. A continuación se muestra el tiempo de resolución para sistemas de 1024×1024 en esta versión. Resulta cerca de 3 veces más rápido que 4.1.

```

1 %timeit cpu_numba_solver(u,v,fu,fv,Ds)
2 18.1 s +- 564 ms per loop (mean +- std. dev. of 7 runs, 1 loop each)

```

4.2.3. Implementación paralela con *Numba*

En esta ocasión la idea es utilizar *Numba* aprovechando que una parte del algoritmo se puede realizar en paralelo. Esto simplemente quiere decir que hay un conjunto de operaciones que puede realizarse en simultáneo en lugar de una por una. Este es el caso para el ciclo *for* que se utiliza para calcular los laplacianos y términos de reacción en cada cuadrante. Es decir, no es necesario calcular el valor correspondiente al cuadrante (i, j) para luego calcular el del cuadrante (i', j') , se pueden calcular simultáneamente, en paralelo.

Lo mejor del caso, nuevamente, es que podemos indicarle de una manera muy sencilla a *Numba* que determinado *for* es paralelizable y listo, *Numba* se encargará de darle las instrucciones en paralelo al procesador. Para ello, todo lo que tenemos que hacer es importar la función *prange* de *Numba* que sirve para indicarle a *Numba* que el ciclo *for* es paralelizable y pasar la opción *parallel = True* al decorador *@njit*. Por completitud mostramos el código de la función *cpu_numba_parallel_solver* en 4.4, pero las únicas diferencias con la versión serial 4.3 son las indicadas aquí.

```

1 from numba import njit, prange
2 import numpy as np
3
4 @njit(parallel = True)
5 def cpu_numba_parallel_solver(u, v, fu, fv, Ds, dt=.01, d = 1, it = 1000):
6     """
7         Solve a 2D reaction-diffusion equation for two dynamical variables with periodic boundary
8             conditions using Numba
9             with a parallel implementation.
10
11        u: Intial conditions for the first dynamical variable. 2d NumPy array of shape (Ly,Lx).
12        v: Intial conditions for the second dynamical variable. 2d NumPy array of shape (Ly,Lx).
13        fu: Reaction term for the first dynamical variable. Function.
14        fv: Reaction term for the second dynamical variable. Function.
15        Ds: Diffusion coefficients for the first and second dynamical variables. List or array of
16            length 2.
17        dt: Time step. Default value is 0.01.
18        d: Space step. Default value is 1.
19        it: Number of iterations. Default value is 1000.
20        """
21
22        Ly, Lx = u.shape
23        u = u.reshape(Ly*Lx)
24        v = v.reshape(Ly*Lx)
25        Fu = np.zeros_like(u)
26        Fv = np.zeros_like(v)
27        Du, Dv = Ds
28
29        for _ in range(it):
30            for i in prange(Lx*Ly):
31                x = i % Lx

```

```

30     y = i // Lx
31     L_u = (u[(x+1)%Lx + Lx*y] + u[(x-1+Lx)%Lx+Lx*y] + u[x + Lx*((y+1)%Ly)] + u[x + Lx*((y
32         -1+Ly)%Ly]) - 4*u[i])/d**2
33     L_v = (v[(x+1)%Lx + Lx*y] + v[(x-1+Lx)%Lx+Lx*y] + v[x + Lx*((y+1)%Ly)] + v[x + Lx*((y
34         -1+Ly)%Ly]) - 4*v[i])/d**2
35     Fu[i] = fu(u[i],v[i]) + Du*L_u
36     Fv[i] = fv(u[i],v[i]) + Dv*L_v
37     u = u + dt*Fu
38     v = v + dt*Fv
39
40     return u.reshape(Ly,Lx),v.reshape(Ly,Lx)

```

Código 4.4: Implementación paralela con *Numba*.

A continuación mostramos el rendimiento obtenido con esta nueva versión, observamos que es cerca de 3 veces más rápido que la versión serial 4.3 y 10 veces más rápido que la versión de *NumPy* 4.1. Esto muestra una primera aproximación al poder de la programación en paralelo y cómo es posible implementarla sin la necesidad de una GPU.

```

1 %timeit cpu_numba_parallel_solver(u,v,fu,fv,Ds)
2 6.31 s +- 439 ms per loop (mean +- std. dev. of 7 runs, 1 loop each)

```

4.2.4. Implementación con *CuPy*

CuPy es una librería de código abierto desarrollada para facilitar el acceso a las GPU compatibles con *CUDA*² en *Python*. *Cupy* ofrece prácticamente todos los métodos de *NumPy* y se encarga de lidiar de forma autónoma y eficiente con el problema de pasar las instrucciones a la GPU. Esta característica por sí sola ya es sorprendente, dado que ofrece la posibilidad de explotar las capacidades de la GPU sin saber nada de CUDA. Ello quiere decir, que en muchos casos basta escribir el código tal como lo haríamos con *NumPy* pero usando *CuPy*, y eso bastaría para tener una mejora decisiva en la eficiencia. De hecho, hagamos la prueba, si tomamos el código 4.1 y lo único que hacemos es cambiar *NumPy* por *CuPy*, el resultado obtenido es el siguiente.

```

1 %timeit gpu_simple_cupy_solver(u,v,fu,fv,Ds)
2 3.12 s +- 3.51 ms per loop (mean +- std. dev. of 7 runs, 1 loop each)

```

Esto es cerca de 19 veces más rápido que la versión de *NumPy* y además la más rápida hasta ahora, lograda con un esfuerzo mínimo. Cabe recordar que el tamaño del sistema que estamos resolviendo, es de 1024×1024 en todos los casos, este es un terreno favorable para el uso de GPU, dado que es un sistema lo suficientemente grande como para que la capacidad de paralelización masiva de la GPU marque la diferencia. Esto es de carácter elemental en lo que respecta al uso de procesadores gráficos para cálculo numérico, sin embargo no está de más recordarlo. No siempre es preferible usar la GPU, hay que ponderar el carácter del algoritmo a utilizar y la magnitud de

²Está en fase experimental la posibilidad de usar GPUs con *ROCm*.

operaciones susceptibles de ser paralelizadas. Si corremos los mismos códigos con un sistema de 100×100 , las cosas cambian rotundamente.

```

1 #Sistemas de 100x100
2 #NumPy
3 %timeit cpu_numpy_solver(u,v,fu,fv,Ds)
4 163 ms +- 502 us per loop (mean +- std. dev. of 7 runs, 10 loops each)
5 #CuPy
6 %timeit gpu_simple_cupy_solver(u,v,fu,fv,Ds)
7 1.07 s +- 3.91 ms per loop (mean +- std. dev. of 7 runs, 1 loop each)
```

Ahora bien, volviendo a sistemas grandes, podría decirse que una mejora en un factor 19 es sorprendente, sin embargo esto es así solo si comparamos con la implementación de *NumPy*, pero si miramos la mejora respecto de la implementación con *Numba* en paralelo, el factor de mejora es cerca de 2. En este caso alguien podría decir, con razón, que la utilización de la *GPU* no está justificada, dado que la mejora en eficiencia no vale el costo que implica el acceso a la GPU.

Para sortear esta razonable objeción, solo debemos profundizar un poco más en las herramientas que nos ofrece *CuPy*. Como decíamos al principio, el hecho de que *CuPy* ofrezca la posibilidad de acceder a la computación en GPU de una manera extremadamente sencilla y con buenos resultados, es sorprendente. Sin embargo, como es habitual, esa sencillez no viene gratis, paga un peaje a la potencia de cómputo real que puede ofrecer una GPU.

En lo que sigue se muestra cómo es posible sacar más provecho de la GPU, sin salirnos del entorno que ofrece *CuPy*. La razón por la que estamos desperdiando potencia de cómputo al resolver el problema reemplazando *CuPy* por *NumPy* es porque estamos lanzando demasiados *kernels* de manera de innecesaria. Es radicalmente más eficiente si conseguimos juntar todas las operaciones necesarias en una menor cantidad de *kernels*. Por cierto, se le dice *kernel* a una función que se ejecuta en la GPU, comúnmente esta función representan operaciones elementales que se realizan en paralelo en la GPU. Para entender esto, vamos a ver un ejemplo sencillo antes de atacar el problema original.

Supongamos que queremos multiplicar las matrices A y B , elemento a elemento, y luego sumar el resultado a la matriz $2C$. Utilizando *CuPy arrays* la función en cuestión y el resultado obtenido es el siguiente.

```

1 import cupy as cp
2 def mul_add(A,B,C):
3     return A*B + 2*C
4
5 L = 1024
6 A = cp.ones((L,L))
7 B = cp.ones((L,L))
8 C = cp.ones((L,L))
9 mul_add(A,B,C)
10 %timeit mul_add(A,B,C)
11 283 us +- 1.26 us per loop (mean +- std. dev. of 7 runs, 1,000 loops each)
```

Ahora bien, podemos hacerlo mejor, el problema con la función anterior es que está utilizando tres *kernels* en lugar de uno, dos para multiplicar y otro para sumar. Sin embargo sería mejor si pudiéramos usar un solo *kernel* que hiciera todo a la vez. Para ello, *CuPy* ofrece la posibilidad de escribir *kernels* personalizados, una de las maneras de hacerlo, es utilizando la función *cupy.ElementwiseKernel*. A continuación se muestra cómo quedaría la función y su resultado utilizando esta alternativa.

```

1 import cupy as cp
2 mul_add_kernel = cp.ElementwiseKernel(
3     'float64 A, float64 B, float64 C', 'float64 out',
4     'out = A*B + 2*C', 'mul_add')
5
6 L = 1024
7 A = cp.ones((L,L))
8 B = cp.ones((L,L))
9 C = cp.ones((L,L))
10 mul_add_kernel(A,B,C)
11 %timeit mul_add_kernel(A,B,C)
12 135 us +- 66.2 ns per loop (mean +- std. dev. of 7 runs, 10,000 loops each)
```

Vemos que la velocidad aumentó en un factor de 2, aún en un ejemplo tan sencillo como este, si aplicamos esta misma idea a algoritmos más complejos tenemos la posibilidad de mejorar la eficiencia sorprendentemente. No voy a entrar en los detalles de utilización de la función *cupy.ElementwiseKernel*, para más detalles recomiendo la siguiente [documentación](#).

Ahora procedemos finalmente con la propuesta para la resolución del problema original usando esta nueva idea de fusionar *kernels* (Código 4.5). Para ello lo que haremos será concentrar en un único *kernel* los cálculos necesarios para evaluar las funciones de reacción y los laplacianos. Nuevamente, no me detendré a explicar los detalles de la implementación, pero espero que el código sea lo suficientemente claro como para motivar el interés del lector.

```

1 import cupy as cp
2
3 forces = cp.ElementwiseKernel(
4     'raw float64 u, raw float64 v, float64 beta, float64 gamma, float64 Du, float64 Dv,
5      uint32 Lx, uint32 Ly',
6     'float64 Fu, float64 Fv',
7     '',
8     int x = i % Lx;
9     int y = (int) i / Lx;
10    Fu = -beta*u[i]*v[i] + Du*(u[(x+1)%Lx+Lx*y] + u[(x-1+Lx)%Lx+Lx*y] + u[x+Lx*(y+1)%Ly] +
11      u[x+Lx*((y-1+Ly)%Ly)] - 4*u[i]);
12    Fv = beta*u[i]*v[i] - gamma*v[i] + Dv*(v[(x+1)%Lx + Lx*y] + v[(x-1+Lx)%Lx+Lx*y] + v[x +
13      Lx*((y+1)%Ly)] + v[x + Lx*((y-1+Ly)%Ly)] - 4*v[i]);
14    '',
15    'forces')
16 euler = cp.ElementwiseKernel(
17     'float64 Fu, float64 Fv, float64 dt', 'float64 u, float64 v',
```

```

18 """
19 u = u + dt*Fu;
20 v = v + dt*Fv;
21 """",
22 'euler')
23
24 def gpu_cupy_solver(u, v, Ds, beta = 1., gamma = .2, dt = .01, d = 1, it = 1000):
25 """
26 Solve a 2D reaction-diffusion equation for two dynamical variables with periodic boundary
27 conditions using CuPy.
28
29 u: Intial conditions for the first dynamical variable. 2d CuPy array of shape (Ly,Lx).
30 v: Intial conditions for the second dynamical variable. 2d CuPy array of shape (Ly,Lx).
31 Ds: Diffusion coefficients for the first and second dynamical variables. List or array of
32 length 2.
33 dt: Time step. Default value is 0.01.
34 d: Space step. Default value is 1.
35 it: Number of iterations. Default value is 1000.
36 """
37 Ly, Lx = u.shape
38 Du, Dv = Ds
39 Fu = cp.zeros_like(u)
40 Fv = cp.zeros_like(v)
41
42 for _ in range(it):
43     forces(u,v,beta,gamma,Du,Dv,Lx,Ly,Fu,Fv)
44     euler(Fu,Fv,dt,u,v)
45
46 return u,v

```

Código 4.5: Implementación con *CuPy*.

Este es el tipo de implementación utilizada en este proyecto de tesis, el resultado obtenido en esta ocasión es el siguiente,

```

1 %timeit gpu_cupy_solver(u,v,Ds)
2 511 ms +- 12.3 ms per loop (mean +- std. dev. of 7 runs, 1 loop each)

```

esto es, mejor en un factor 6 que la versión estándar de *CuPy*, 12 veces mejor que la implementación con *Numba* en paralelo (la mejor sin usar GPU), y 117 veces más rápido que la versión con *NumPy*. Ahora sí, vemos que la diferencia entre usar una GPU y no usarla es, por lo menos, en un factor de 12, la diferencia entre una simulación de un 1 día y una de 12 días.

4.3. Tamaño del sistema

Por último, quisiera terminar este capítulo con algunos comentarios. Por un lado, recordar que todas las comparaciones de velocidad en las diferentes implementaciones se realizaron con los mismos parámetros, y fundamentalmente con sistemas relativamente grandes, 1024×1024 , donde la GPU se ve favorecida sobre la CPU. Diferente es el caso para sistemas chicos, por completitud en este sentido, en la figura 4.1 mostramos los resultados de velocidad de las distintas implementaciones para distintos tamaños

de sistemas. En esta misma figura se muestran también los tiempos de ejecución sobre una GPU *GeForce RTX 3080* del *cluster* de GPUs de la Gerencia de Física del Centro Atómico Bariloche, donde se realizaron la mayoría de las simulaciones de este trabajo.

Por otro lado, la mayoría de los resultados presentados en este proyecto fueron obtenidos con sistemas de $2^{16} \times 2^{11}$, en un sistema de dos ecuaciones de reacción-difusión como el discutido aquí, esto implica cuatro matrices con 2^{27} elementos cada una, considerando que además utilizamos un formato en coma flotante de doble precisión, cada matriz ocupa alrededor de *1GB* de memoria. El tiempo que llevaría resolver sistemas de este tamaño sin GPU sería inviable.

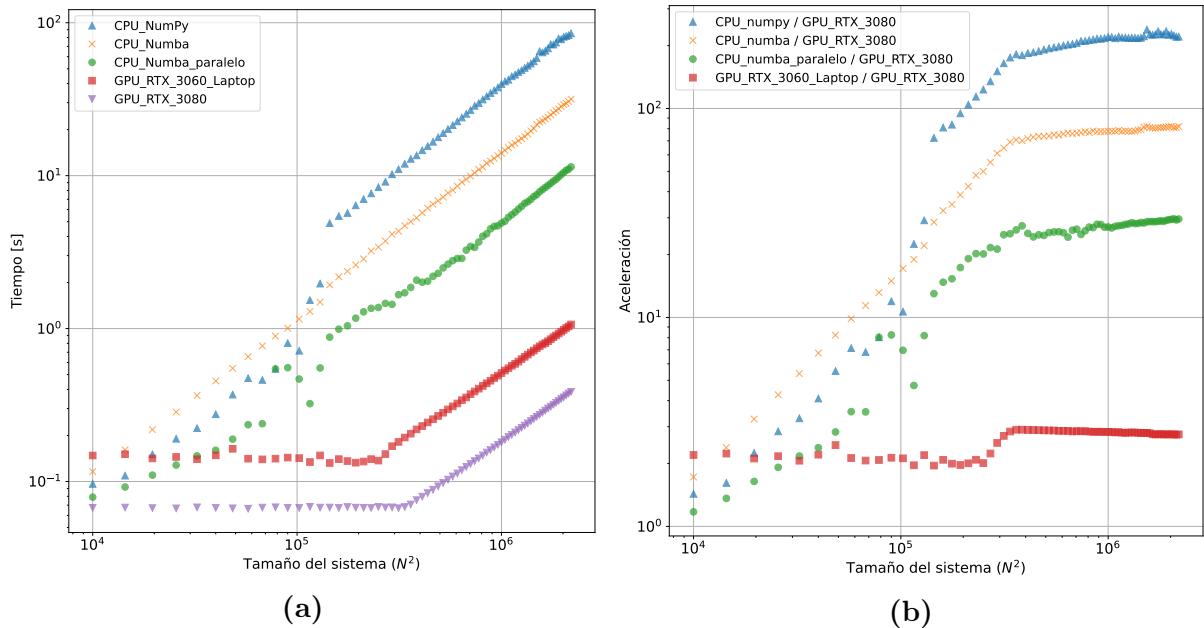


Figura 4.1: (a) Tiempo de simulación en función del tamaño del sistema $N \times N$ para las distintas implementaciones. (b) Aceleración obtenida al usar GPU en función del tamaño del sistema $N \times N$ para las distintas implementaciones.

Finalmente, es posible que quienes estén más interiorizados con *CUDA* tal vez estén sorprendidos por el carácter superficial con el que usamos la GPU aquí. Es decir, cuando escribimos el *kernel* no lo escribimos en *CUDA C* o *CUDA C++*, es algo similar, pero es distinto, lo cual puede confundir un poco. Además, en ningún momento indicamos cantidad de hilos por bloque ni cantidad de bloques en la grilla. Esto es, nuevamente, porque *CuPy* intenta simplificar todo lo posible el uso de la GPU, para que sea accesible a usuarios sin conocimientos de *CUDA*. Sin embargo, *CuPy* ofrece también todo el acceso al detalle, tal como podría hacerse con *CUDA C* por ejemplo, y admite la escritura de *kernels* directamente en *CUDA C* a través de la función `cupy.RawKernel`. En el material complementario de este capítulo, dejo la implementación usando esta función que da acceso completo a *CUDA*. No la agrego acá porque entiendo que no aporta demasiado, adicionalmente, no pude obtener ninguna mejora notable respecto al *kernel* escrito con `cupy.ElementwiseKernel`.

Capítulo 5

Fenómenos críticos del frente de infección

En este capítulo resolvemos numéricamente el modelo SIR espacial heterogéneo propuesto en el capítulo 3. Es decir, el sistema

$$\partial_t S = -\beta_{\mathbf{r}} S I + D_S \nabla^2 S, \quad (5.1)$$

$$\partial_t I = \beta_{\mathbf{r}} S I - \gamma I + D_I \nabla^2 I, \quad (5.2)$$

con $(x, y) \in [0, L_x] \times [0, L_y]$ y condiciones iniciales

$$I(x, y, 0) = \begin{cases} I_0 & \text{si } (x, y) \in \Omega_0 \\ 0 & \text{si } (x, y) \notin \Omega_0 \end{cases}; \quad S(x, y, 0) = \begin{cases} 1 - I_0 & \text{si } (x, y) \in \Omega_0 \\ S_0 & \text{si } (x, y) \notin \Omega_0. \end{cases}$$

donde $\Omega_0 = (0, \delta x) \times [0, Ly]$, es decir, la condición inicial consiste en un frente de infección plano. En cuanto a las condiciones de contorno tomamos condiciones periódicas en la dirección y y condiciones de Dirichlet en la dirección x dadas por

$$I(0, y, t) = I(L_x, y, t) = S(0, y, t) = S(L_x, y, t) = 0.$$

Todas las simulaciones realizadas en este capítulo se realizaron sobre sistemas relativamente chicos de 1024×1024^1 . Se decidió utilizar este tamaño de sistema dado que en este capítulo nos centramos en estudiar características macroscópicas del frente, las cuales presentan pequeñas fluctuaciones para diferentes realizaciones y transitorios cortos, de modo que un sistema de este tamaño resultó suficiente. Para referencia es importante decir que en este capítulo utilizamos un paso temporal de 0.05 para integrar las ecuaciones, $Lx = Ly = 1024$, $\delta x = 1$ y $S_0 = I_0 = 1$. Los demás parámetros del modelo se especificarán según corresponda.

¹Comparados con los del capítulo 6.

En particular, vamos a estar observando la velocidad c , la amplitud máxima I_{max} y la nocividad S_1 (a definir) del frente de infección en función del parámetro de desorden p y la tasa de transmisión media espacial $\langle \beta_r \rangle_r = \beta_m$ para las distintas heterogeneidades. Veremos cómo cada uno de estos observables muestra un comportamiento crítico cerca del umbral de propagación y determinaremos los exponentes críticos asociados. En el camino discutiremos las diferencias de los resultados entre las heterogeneidades y las posibles interpretaciones en términos epidemiológicos.

5.1. Medio homogéneo (H)

Por completitud comenzamos mostrando los resultados numéricos obtenidos para el problema homogéneo, donde podemos corroborar con lo desarrollado analíticamente en el capítulo 3.

En la figura 5.1a se muestra la evolución del frente de infección para el caso homogéneo, donde se utilizó $\beta = 1, \gamma = 0.2, D_S = 0$ y $D_I = 1$. Nótese que $\beta > \beta_c = \frac{\gamma}{S_0} = 0.2$ de modo que la tasa de transmisión es mayor que la tasa de transmisión crítica y por lo tanto se observa una solución en forma de onda. Se muestra también (fig 5.1a) el campo de desplazamiento del frente $u(y, t)$ (ecuación 3.24), el cual es completamente plano, como es de esperar con una tasa de transmisión homogénea. Si se mira con detenimiento (fig. 5.1b), puede observarse que el perfil del frente es asimétrico, es decir, se comporta de manera distinta en la parte frontal que en la parte posterior, tal como habíamos anticipado analíticamente en el capítulo 3. Los perfiles asintóticos, frontal y posterior, coinciden con los obtenidos de las ecuaciones 3.13 y 3.14 respectivamente.

En la figura 5.1c se muestra la posición del centro de masa del frente de infección $u_{cm}(t)$ (ecuación 3.25) en función del tiempo para distintos valores de γ/β . Puede observarse que para el caso $\gamma/\beta = 1$ el sistema cambia de comportamiento y no es posible identificar un frente de onda propagándose a velocidad constante y la amplitud del mismo se va a cero, en acuerdo con la condición de umbral $\gamma/\beta < S_0 = 1$. Por otro lado, se calculó la velocidad c (ecuación 3.27) para cada una de estas curvas descartando un período transitorio. De ello vemos que la velocidad del frente, estimada analíticamente $c_0 = 2\sqrt{D_I(\beta S_0 - \gamma)}$, concuerda con las velocidades obtenidas numéricamente con un error relativo, en promedio, de 4.5 %.

En la mayoría de los casos c_0 resulta sobreestimar levemente el valor de c , sin embargo, c_0 originalmente era una cota mínima para la velocidad, ¿cómo puede ser? Esto puede explicarse recordando que el valor de c_0 se obtuvo en una aproximación lineal donde se supone que hay más susceptibles de los que realmente hay, por ello es de esperar que la cota mínima estimada c_0 sea sutilmente mayor que la real. En cualquier caso, se observa que el frente prefiere la velocidad mínima posible, esto es un fenómeno común en este tipo de sistemas [40].

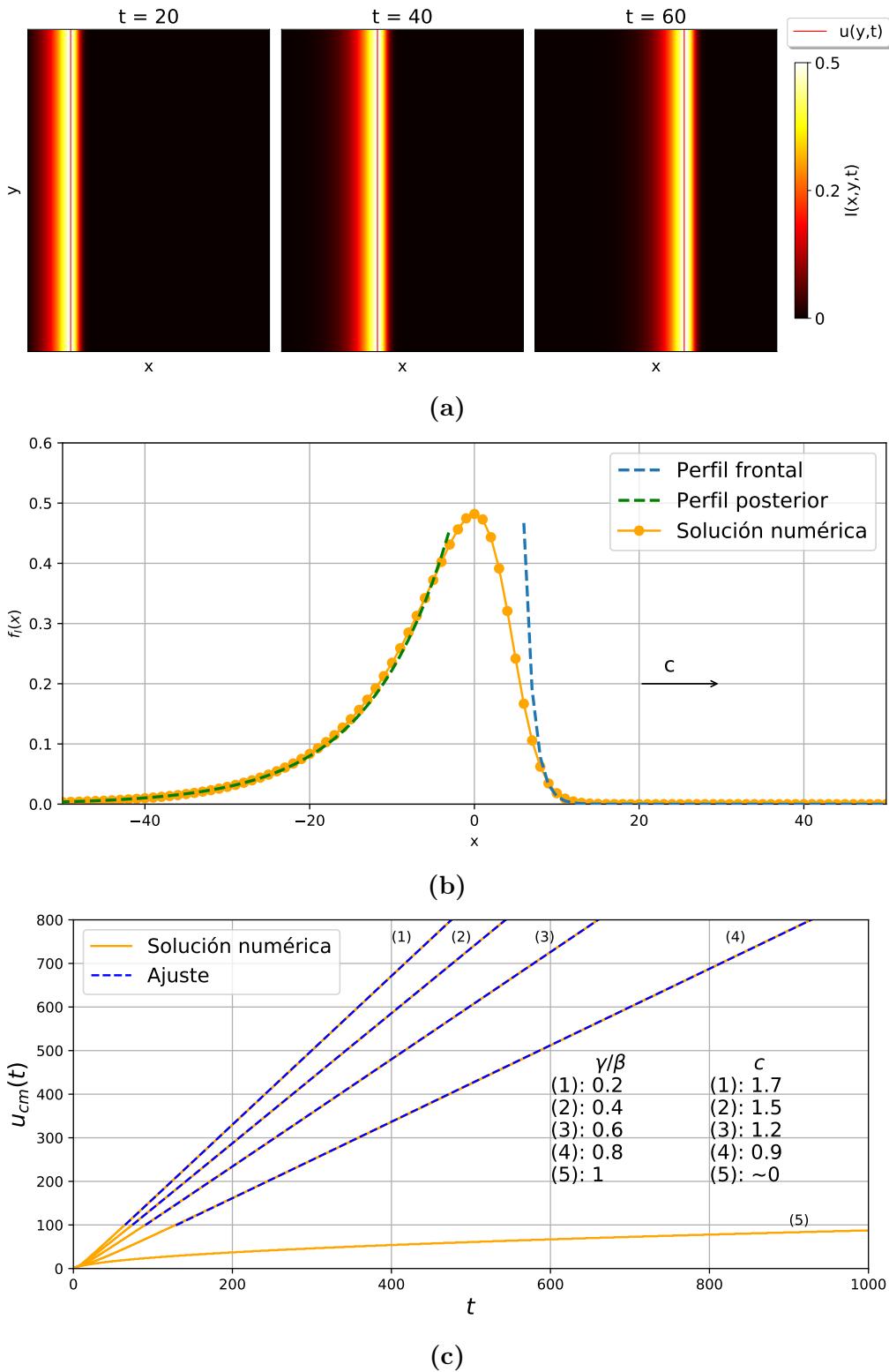


Figura 5.1: (a) Mapa de color de la evolución del frente de infección sobre un medio homogéneo con parámetros $\beta = 1$, $\gamma = 0.2$, $D_S = 0$ y $D_I = 1$. (b) Perfil longitudinal del frente de infección, se muestran también los perfiles asintóticos frontal y posterior del frente de onda calculados en la sección 3.3. (c) Posición del centro de masa del frente de infección en función del tiempo para distintos valores de γ/β junto con los correspondientes ajustes lineales. Se muestra también la velocidad c obtenida para cada caso.

5.2. Medios heterogéneos

5.2.1. DA vs H

Vamos a comenzar estudiando el efecto que tiene la heterogeneidad DA en la solución del sistema. Para ello tendremos de referencia todo lo que sabemos del sistema sobre un medio H, tanto resultados analíticos como numéricos. La comparación entre DA y H nos permitirá entender mejor los efectos de la heterogeneidad.

Observando la figura 5.2, que muestra el frente de infección propagándose sobre un medio DA con $p = 0.3$, encontramos diferencias notables a simple vista con el frente sobre H (fig. 5.1a). En principio, lo más notable es que el frente deja de ser plano y se vuelve rugoso. Esto es de esperar, ya que el frente ahora se encuentra con una fracción p de lugares en donde no pueden darse contagios dado que la tasa de transmisión se anula, dando lugar a un avance tortuoso sobre el espacio, y fluctuante en el tiempo.

Sin embargo, podemos hacer algunas preguntas más difíciles, por ejemplo, ¿cuál es el umbral de propagación para el caso DA?, ¿cuál es la velocidad del frente?, ¿dada la misma tasa de transmisión espacial media, es más rápido el frente sobre el medio DA o sobre el medio H?

$$p, 1 - \frac{\beta^H}{\beta^{DA}} \quad (5.3)$$

Velocidad y umbral de propagación

Uno podría intentar responder estas preguntas pensando en una aproximación de campo medio en la que usamos los resultados del caso H, pero reemplazamos β por la tasa de transmisión media espacial $\langle \beta_r \rangle_r = \beta_m$ del medio DA. Asumiendo un sistema de tamaño infinito, entonces la tasa de transmisión media espacial es igual a la tasa de

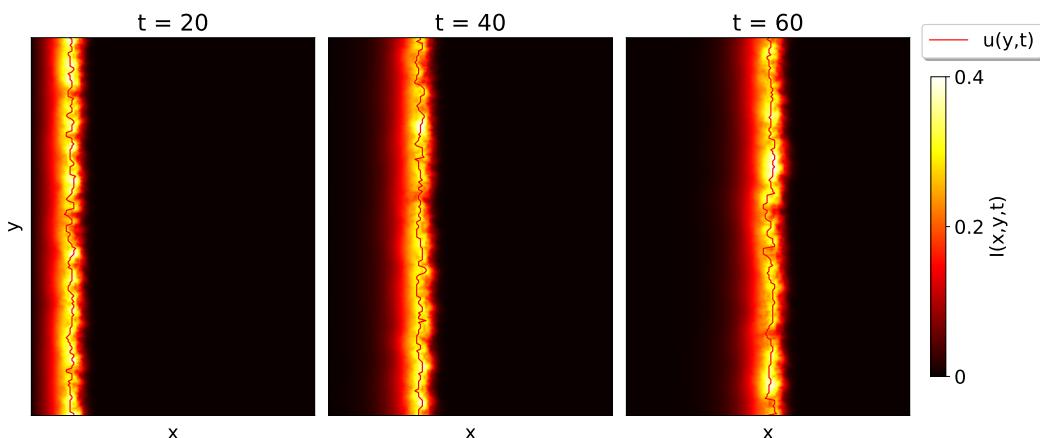


Figura 5.2: Evolución del frente de infección sobre un medio heterogéneo DA con $p = 0.3$, $\gamma = 0.2$, $\beta = 1$, $D_S = 0$ y $D_I = 1$.

transmisión media sobre desorden, es decir,

$$\beta_m = \overline{\beta_r} = \beta(1 - p). \quad (5.4)$$

De esta manera, en lo que respecta al umbral, vimos que en el caso homogéneo la tasa de transmisión crítica a partir de la cual se sostiene un frente de onda es, $\beta_c = \frac{\gamma}{S_0}$. Reemplazando este β_c por β_m encontramos entonces un valor crítico *naive* para el parámetro de desorden p_c dado por,

$$p_c = 1 - \frac{\gamma}{\beta S_0}, \quad (5.5)$$

que define el umbral de propagación. Haciendo algo similar, reemplazando β_m por β en la expresión para la velocidad del frente en el caso homogéneo c_0 (ecuación 3.12), encontramos lo que sería la velocidad del frente en función de p ,

$$c(p) = c_0 \left(1 - \frac{p}{p_c} \right)^{1/2}. \quad (5.6)$$

Desafortunadamente, o quizás afortunadamente, estos resultados predicen incorrectamente los valores obtenidos de las simulaciones. En la figura 5.3 se muestra la velocidad del frente,

$$c = \langle c(t) \rangle_t = \langle \overline{\dot{u}_{cm}(t)} \rangle_t \quad (5.7)$$

para distintos valores del parámetro de desorden p sobre el medio DA, ignorando un transitorio. En la misma se observa efectivamente un fenómeno crítico donde la velo-

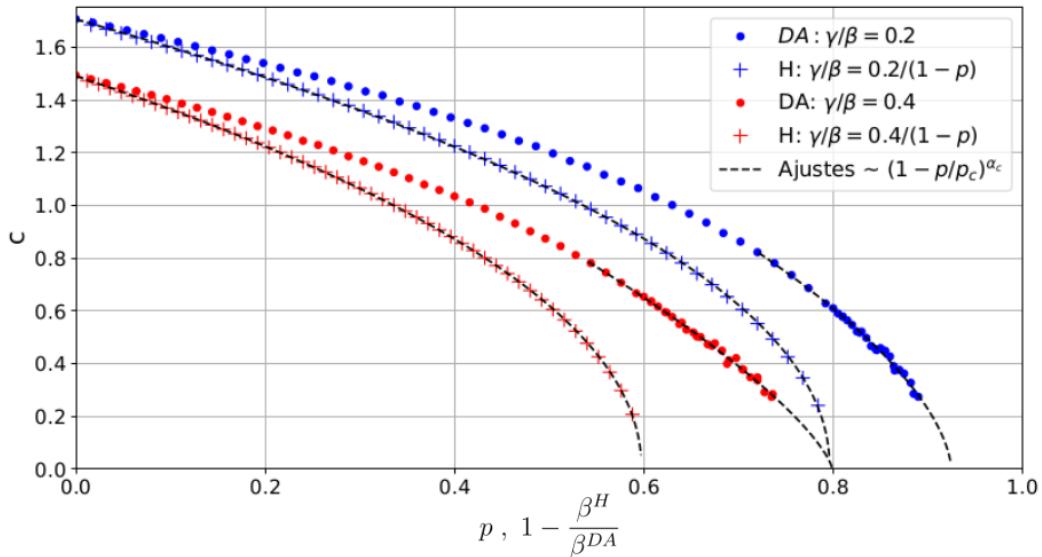


Figura 5.3: Velocidad c del frente de infección en función de p para los medios H y DA con la misma tasa de transmisión media espacial usando $\gamma/\beta = 0.2$ y 0.4 , $D_S = 0$ y $D_I = 1$. Se muestran los ajustes con la regla de potencia $c \propto (1 - p/p_c)^{\alpha_c}$ sobre la región crítica.

ciudad del frente se anula siguiendo una ley de potencia del tipo $c \propto (1 - p/p_c)^{\alpha_c}$, sin embargo, tanto el exponente crítico α_c como el umbral de propagación p_c no coinciden con los valores predichos por la aproximación de campo medio. En la tabla 5.1 se pueden observar los valores obtenidos ajustando la ley de potencias en los resultados para la velocidad.

Se observa que las curvas de velocidad sobre el medio H en función de $p = 1 - \beta$ concuerdan con los resultados obtenidos analíticamente (tabla 5.1), por ejemplo con $(\gamma = 0.2, S_0 = 1)$, resulta que $p_c = 0.8 \pm 0.01$ que concuerda con $\beta_c = 1 - p = 0.2$. De igual manera, los exponentes críticos obtenidos $\alpha_c \approx 0.5$ también coinciden con la expresión para la velocidad c_0 del caso homogéneo.

Por otro lado, calculamos la velocidad del frente sobre un medio H con la misma tasa de transmisión media espacial que la de un medio DA con un dado p (fig. 5.3), es decir, tomamos $\beta^H = \beta^{DA}(1 - p)$. **Resulta que el frente de infección es más veloz sobre el medio DA que sobre el medio H dada la misma tasa de transmisión media espacial.** Esto sugiere que la heterogeneidad introducida en el sistema da lugar a un efecto de aceleración del frente de infección. Esta aceleración no puede explicarse utilizando una tasa de transmisión efectiva igual a la tasa de transmisión media espacial del medio DA. Se trata de un fenómeno emergente no trivial. En términos epidemiológicos, si uno piensa en p como la fracción de lugares donde la población está vacunada o en cuarentena, entonces una estrategia de vacunación o cuarentena con distribución espacial de tipo DA daría lugar a un frente de infección más veloz que una estrategia de vacunación o cuarentena espacialmente homogénea.

Más aún, no solo hay un efecto de aceleración, si no que el umbral de propagación se reduce a un valor de transmisión media espacial menor en el medio DA. Esto es, tal como se muestra en la figura 5.3, dados $(\gamma = 0.2, S_0 = 1)$ la tasa de transmisión crítica en un medio H es $\beta_c = \gamma/S_0 = 0.2$. Sin embargo, en un medio DA con estos parámetros obtenemos el valor crítico $p_c = 0.92 \pm 0.04$, lo que se traduce en una tasa de transmisión media de $\beta_m = \beta(1 - p) \approx 0.08$. Es decir, es más fácil propagar la infección en un medio DA que en un medio H dada la misma tasa de transmisión media espacial.

Por lo tanto, la “homogeneización” naive no es una buena aproximación: subestima el umbral y la velocidad de propagación.

Tabla 5.1: Parámetros críticos p_c y α_c de los medios H y DA con diferentes γ/β .

Medio	γ/β	p_c	α_c
DA	0.2	0.92 ± 0.04	0.6 ± 0.1
H	0.2	0.80 ± 0.01	0.47 ± 0.04
DA	0.4	0.80 ± 0.04	0.7 ± 0.1
H	0.4	0.60 ± 0.01	0.48 ± 0.04

Por lo visto hasta ahora uno pensaría, al menos en términos epidemiológicos, que es preferible tener una tasa de transmisión con una distribución homogénea antes que una de tipo DA, ya que así habría una propagación más lenta. Sin embargo, antes de asegurar esto debiéramos tener en cuenta otras características epidemiológicas relevantes. Una de ellas es la amplitud del frente, una amplitud grande del frente implicaría muchos individuos infectados al mismo tiempo, lo cual podría demandar una cantidad de recursos e infraestructura de salud simultáneamente inaccesible. Esto podría producir fácilmente el colapso de los servicios de salud dando lugar a muchas tragedias. Entonces, ¿cómo es la amplitud del frente sobre el medio DA en comparación con H?

Amplitud del frente

De manera similar a lo que hicimos con la velocidad del frente, calculamos la amplitud del mismo sobre los medios H y DA. Más precisamente, tomamos un valor para cada simulación promediando temporalmente y descartando un transitorio, es decir,

$$I_{max} = \langle I_{max}(t) \rangle_t. \quad (5.8)$$

Los resultados obtenidos se muestran en la figura 5.4. En primer lugar observamos que la amplitud del frente se anula en los mismos valores de p_c en los que se anula la velocidad. Esto es razonable dado que al detenerse, el frente de infección no puede encontrar nuevos susceptibles, de modo que los infectados terminan recuperándose en un período $\frac{1}{\gamma}$ y la amplitud del frente se anula.

Por otro lado, más interesante, vemos que dados los mismos parámetros y la misma tasa de transmisión media espacial, **la amplitud máxima del frente es mayor**

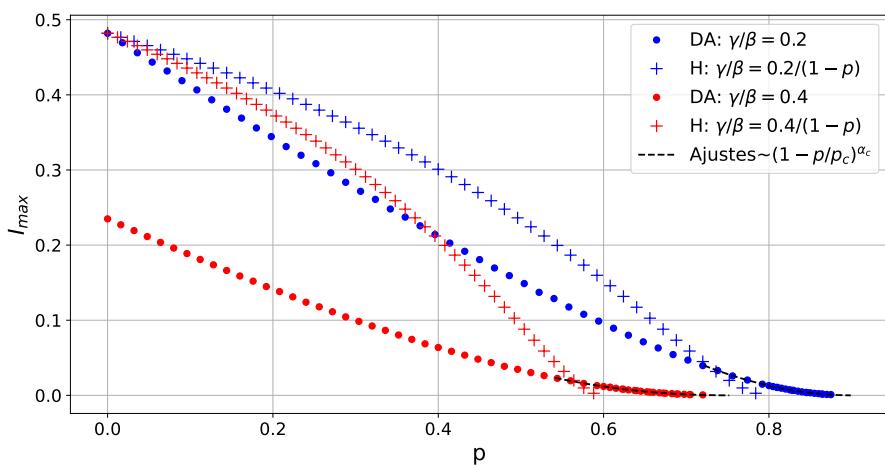


Figura 5.4: Amplitud media I_{max} del frente de propagación en función de p con $\gamma/\beta = 0.2$ y 0.4 y $D_I = 1$. Se muestran los ajustes con la regla de potencia $I_{max} \propto (1 - p/p_c)^{\alpha_f}$ sobre la región crítica.

Tabla 5.2: Parámetros críticos p_c y α_I del medio DA con diferentes γ/β .

Medio	γ/β	p_c	α_I
DA	0.2	0.90 ± 0.04	1.93 ± 0.05
DA	0.4	0.75 ± 0.04	2.12 ± 0.05

sobre el medio H que sobre el medio DA en tanto $p < p_c^H$, donde p_c^H es el umbral del medio H. De modo que, al contrario de lo que sucede con la velocidad, la heterogeneidad introducida en el sistema da lugar a un efecto de disminución de la amplitud del frente. En este sentido entonces es preferible una distribución DA antes que una H.

Es posible relacionar los resultados de la velocidad y la amplitud conceptualmente de la siguiente manera: en general, si un frente de infección es más rápido entonces pasa menos tiempo localizado en un lugar, consecuentemente se contagia menos gente en ese lugar y por lo tanto la amplitud es menor, si el frente es lento se da el efecto contrario.

Observamos también que la amplitud máxima media sigue una ley de potencias $I_{max} \propto (1 - p/p_c)^{\alpha_I}$ con exponente crítico α_I . En la tabla 5.2 mostramos los valores de p_c y α_I obtenidos con este ajuste. Se ve que en ambos casos el exponente crítico es aproximadamente igual a 2.

En resumen, observamos que para decidir en términos epidemiológicos qué distribución β_r es más conveniente, es necesario primero decidir si es preferible un frente rápido con menor amplitud o uno rápido con mayor amplitud. Para ello quizás sea conveniente definir otra magnitud. Quizás lo interesante sea pensar que las epidemias se dan en espacios finitos, de modo que lo importante en definitiva sería saber qué daño causará realmente el frente de infección en determinado espacio. Para ello vamos a introducir lo que llamamos *nocividad* S_1 del frente de infección, la cual se definirá más adelante a la vez que mostramos resultados sobre las demás heterogeneidades.

Los resultados obtenidos hasta aquí de la comparación entre el medio H y el DA pueden describirse cualitativamente de la siguiente manera:

- Si el valor medio espacial de la tasa de transmisión β_m es el mismo, la velocidad del frente de infección es *mayor* sobre el medio DA que sobre el H.
- Si el valor medio espacial de la tasa de transmisión β_m es el mismo, la amplitud máxima media del frente de infección I_{max} es *mayor* sobre el medio H que sobre el DA en tanto $p < p_c^H$, con p_c^H el umbral del medio H.
- El valor de umbral de la tasa de transmisión media β_m para el cual se extingue el frente de infección es *menor* sobre el medio DA que sobre el H.

- Los exponentes críticos de la velocidad α_c para el medio homogéneo *coinciden* con el estimado por c_0 en la aproximación analítica de la solución de onda. ($\alpha_c = \frac{1}{2}$)
- Los exponentes críticos de la velocidad α_c para el medio DA son aparentemente *mayores* que para el medio H. Sin embargo, más precisión es necesaria para decidir si α_c es universal o no. ($\alpha_c \approx 0.5$)

5.2.2. Medios correlacionados

En esta sección sumamos a los resultados los efectos de las heterogeneidades con correlación, es decir, la heterogeneidad «suavizada» (S) y la «dicotómica-correlacionada» (DC) únicamente con $n = 1$, un paso del algoritmo que genera S y DC discutido en el capítulo 3.

A continuación mostramos los resultados de velocidad y amplitud del frente de infección sobre todos los medios propuestos. La diferencia es que ahora estaremos viendo los resultados directamente en función de la tasa de transmisión media β_m en lugar de p .

Velocidad y amplitud del frente

En la figura 5.5 se muestra la velocidad en función de la tasa de transmisión media espacial β_m para todos los medios, H, DA, S y DC. En primer lugar observamos que la curva asociada al medio S cae entre las curvas del medio H y el medio DA. Esto es razonable dado que el medio S se genera del medio DA y tiende a un homogéneo cuanto más los suavizamos.

Por otro lado, la curva de velocidad asociada al medio DC muestra la mayor velocidad del frente infección de todos los medios. Este resultado es muy interesante, es decir, **un ligero cambio en la estructura del medio, sin modificar la tasa de transmisión media, produce un cambio notable en la velocidad del frente**. Fundamentalmente, comparando el resultado de DC con el de DA, a pesar de que ambos medios tengan un carácter dicotómico, la correlación espacial introducida en DC genera un cambio apreciable sobre la velocidad.

Para entender mejor qué es lo que produce este cambio de velocidad entre DA y DC es interesante observar con mayor detenimiento la estructura de estos medios. En las figuras 5.6 y 5.7 se muestra la distribución de la tasa de transmisión β_r sobre el espacio de los medios DA y DC respectivamente, donde ambos tienen la misma tasa de transmisión media espacial. Observando la distribución espacial de cada uno de ellos a gran escala, sobre el espacio de 1024×1024 , no es sencillo señalar ninguna diferencia. Sin embargo, al observar la distribución sobre una región acotada de 100×100 se puede ver que son estructuralmente distintos en esta escala. La distribución de DC con

Tabla 5.3: Parámetros críticos β_c y α_c de los medios DC, DA ,S y H.

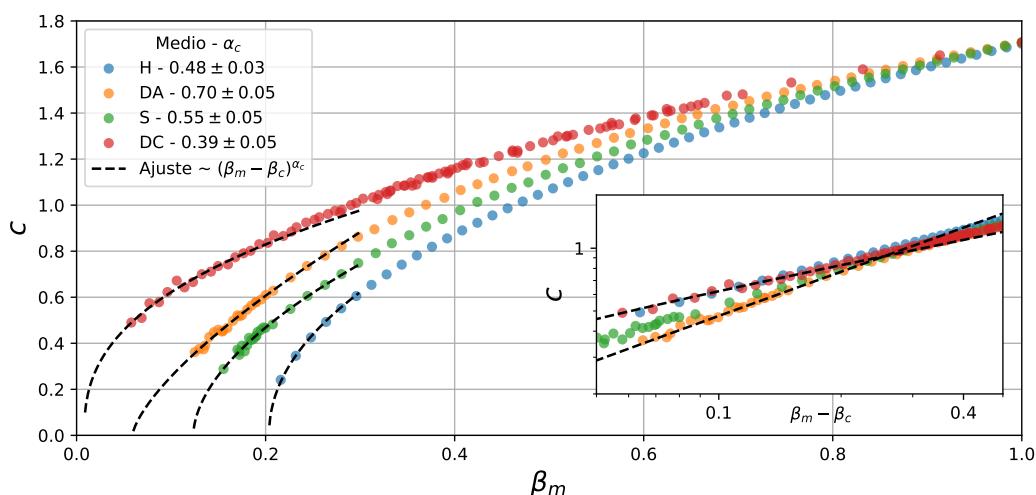
Medio	β_c	α_c
H	0.20 ± 0.01	0.48 ± 0.05
DA	0.06 ± 0.02	0.7 ± 0.1
DC	0.01 ± 0.01	0.4 ± 0.1
S	0.12 ± 0.02	0.6 ± 0.1

$n = 1$ alcanza a formar un patrón distinguible, se forman regiones agrupadas con tasa de transmisión no nula β , mientras que en DA, por supuesto, se observa el carácter aleatorio. Es decir, la velocidad de propagación es mayor si la distribución de la tasa de transmisión forma estructuras como las que se ven en la figura 5.7 en lugar de estar distribuida aleatoriamente, por más que la tasa de transmisión media sea la misma.

Se realizaron ajustes con la ley de potencia $c \propto (\beta_m - \beta_c)^{\alpha_c}$ para las curvas de velocidad en la cercanía del umbral de propagación para los medios DC, DA, S y H. De ello se determinó la tasa de transmisión crítica β_c , que define el umbral de propagación, y el exponente crítico α_c para cada uno de ellos. En la tabla 5.3 se muestran los resultados obtenidos.

De estos resultados también queda claro otro fenómeno notable al que da lugar la estructura del medio DC, y es que sobre este la tasa de transmisión crítica es *menor* comparada con las demás. Es decir, basta con una tasa de transmisión extremadamente chica, comparada con los demás medios, para que tenga lugar un frente de infección.

Por otro lado, en la figura 5.8 se muestran los resultados de la amplitud del frente en función de la tasa de transmisión media para los distintos medios. Observamos que el medio homogéneo es el que sostiene la mayor amplitud del frente de infección en tanto

**Figura 5.5:** Velocidad del frente de infección en función del valor medio de la tasa de transmisión β_m sobre los medios S, DC, DA y H. Se muestran los ajustes con la regla de potencia $c \propto (\beta_m - \beta_c)^{\alpha_c}$ sobre la región crítica.

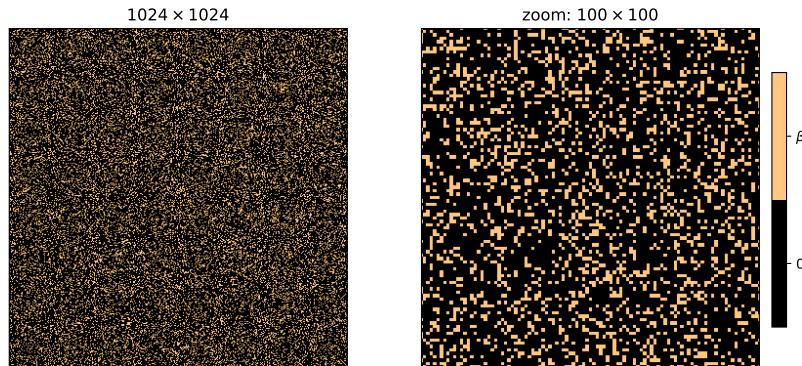


Figura 5.6: Distribución de la tasa de transmisión β_r del medio DA. A izquierda se muestra la distribución sobre todo el espacio de 1024×1024 y a la derecha un acercamiento a una región de 100×100 .

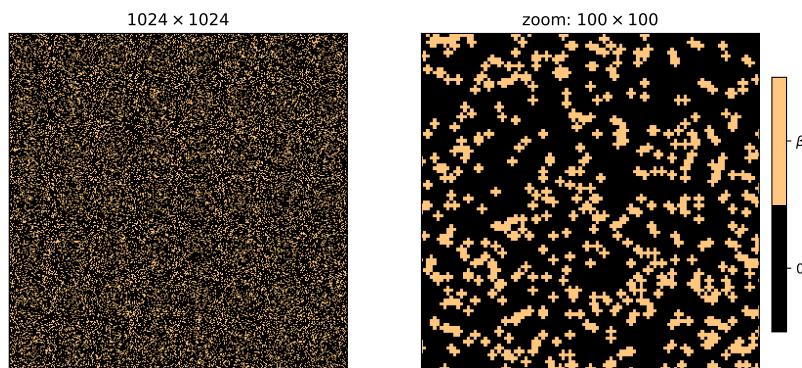


Figura 5.7: Distribución de la tasa de transmisión β_r del medio DC. A izquierda se muestra la distribución sobre todo el espacio de 1024×1024 y a la derecha un acercamiento a una región de 100×100 .

$\beta_m > 0.35$. Vemos aquí nuevamente que la curva de S queda entre las dadas por H y DA. Por otro lado, el medio DC, en consistencia con lo observado para la velocidad, otorga la mayor amplitud del frente para valores bajos de la tasa de transmisión media $\beta_m < 0.35$. Es decir, el medio DC favorece nuevamente, en este aspecto y sobre este régimen, a la propagación del frente.

Sin embargo, nuevamente, no queda claro si es preferible un medio sobre otro, dado que observamos sistemáticamente que cuando la velocidad del frente es mayor la amplitud del mismo es menor en comparación con los frentes de menor velocidad. Para tratar de obtener un criterio más sobre el que juzgar introducimos la *nocividad* de los frentes y vemos cómo es en cada medio.

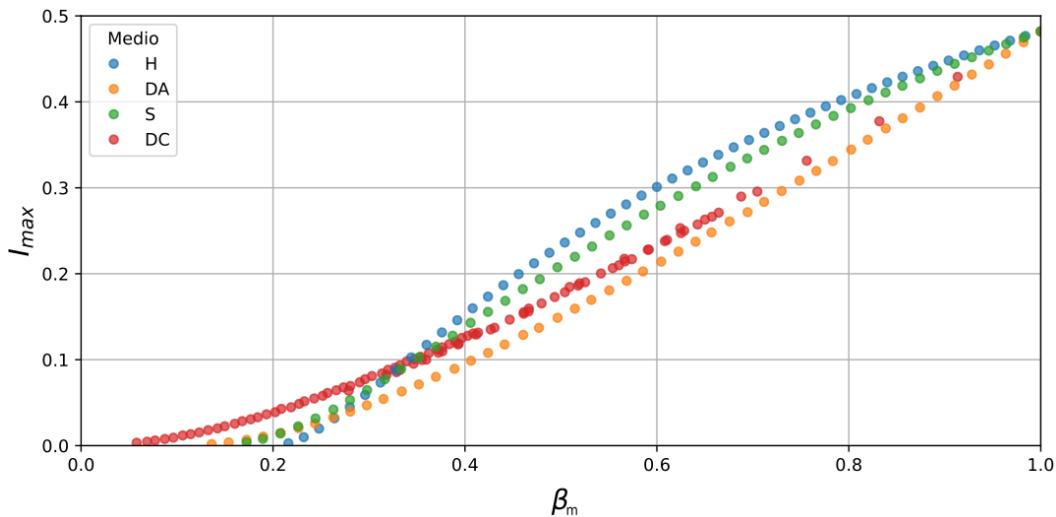


Figura 5.8: Amplitud media del frente de infección/incendio I_{max} en función de la tasa de transmisión media $\bar{\beta}_r$ para los medios S, DA, DC y H.

Nocividad

Un aspecto de gran importancia en lo que respecta a problemáticas tanto epidemiológicas como de incendios es la posibilidad de cuantificar los daños que deja un brote de enfermedad infecciosa o un incendio. En particular, estaremos interesados en estimar el daño que queda tras el paso del frente de infección/incendio sobre una dada región. Para ello utilizaremos como cuantificador de daños la fracción de susceptibles remanentes S_1 que queda tras el paso del frente. Entendiendo que mientras menor sea esta fracción, mayor será el daño ocasionado por el frente.

Para ser más precisos al respecto, tomaremos

$$S_1 = \langle S(x, y, T) \rangle_{x,y} \quad (5.9)$$

con $0.2L_x < x < u_{cm}(T) - 0.2L_x$, siendo T el tiempo que dure la simulación correspondiente². Definido de esta manera S_1 , quitamos los efectos del transitorio que se da en la primera región de evolución del sistema y posibles efectos de tamaño finito por estar cerca de los bordes de la región de simulación. Además imponemos que x se encuentre relativamente lejos del centro de masa del frente para no contar los susceptibles en la región del frente mismo.

Utilizando este criterio para todas las simulaciones, podremos comparar la nocividad de cada frente al modificar los parámetros que interesen. En particular, veremos cómo cambia esta magnitud S_1 con la tasa de transmisión media espacial β_m .

En la figura 5.9 se muestran los resultados obtenidos de S_1 en función de β_m sobre

²Notar que T varía en cada realización, fundamentalmente dado que la velocidad de los frentes en los distintos medios cambia. De modo que esta definición pone el foco en el daño que se hace en una dada región sin importar el tiempo que le lleve al frente recorrerla.

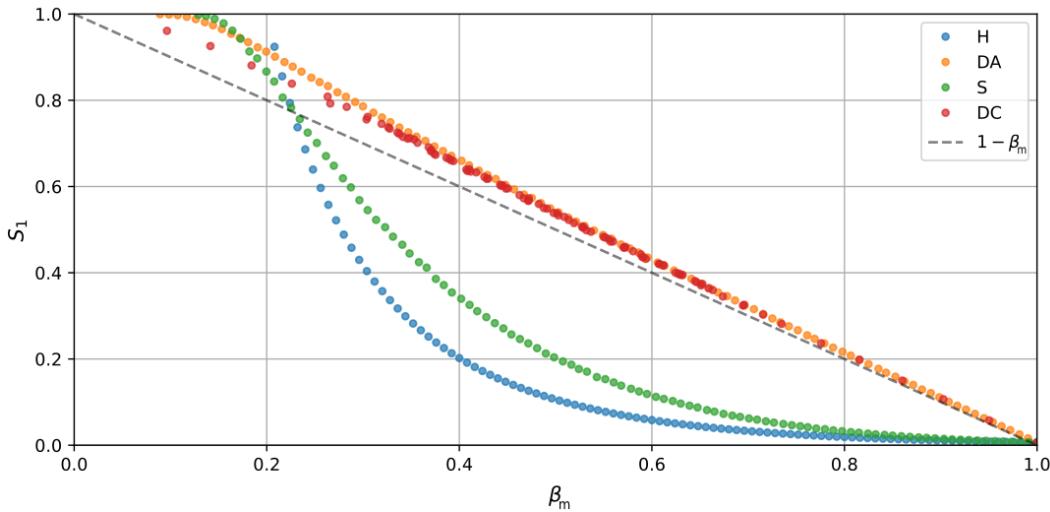


Figura 5.9: Susceptibles remanentes S_1 en función de la tasa de transmisión media β_m para los medios H, S, DA y DC. Se muestra también la curva $1 - \beta_m$ que aproxima S_1 sobre los medios dicotómicos.

los medios H, S, DA y DC. Se puede observar que el frente de propagación más nocivo se da sobre el medio homogéneo, ya que da lugar a la menor fracción de susceptibles S_1 tras el paso del frente. Sin embargo, tal como ya hemos apreciado antes (Tabla 5.3), la tasa de transmisión crítica es mayor que en los demás casos $\beta_c \approx 0.2$.

El siguiente medio de nocividad elevada es S, mientras que los medios DC y DA presentan curvas similares de carácter lineal. Estos nuevos resultados dan una perspectiva nueva acerca de los medios. Por ejemplo, habíamos visto que la velocidad de propagación del frente es mayor sobre el medio DC que sobre los demás, lo cual es una característica indeseable en problemáticas tanto epidemiológicas como de incendios. Sin embargo, vemos ahora que a pesar de ello la nocividad del frente, tal como la hemos caracterizado aquí, es menor sobre DC que sobre los demás medios, lo cual es deseable. A modo cualitativo es entendible que si el frente se desplaza a mayor velocidad tenga menos tiempo de ocasionar grandes daños.

Respecto al carácter lineal que presentan las curvas de S_1 sobre los medios dicotómicos, podemos hacernos una idea de a qué se debe: los susceptibles en los lugares donde la tasa de transmisión es nula resultan completamente inaccesible al frente dado que solo hemos trabajado con el término difusivo de infectados en las ecuaciones del sistema (6.1 - 6.2), ($D_S = 0, D_I = 1$). Es decir, los infectados pueden ingresar a esas zonas, donde no pueden contagiar a nadie dado que la tasa de transmisión es nula, pero los susceptibles nunca salen de modo que nunca pueden infectarse. Es por ello que como mínimo la cantidad de susceptibles remanentes debe ser igual a la cantidad de susceptibles inaccesibles, que es igual a la fracción de lugares donde la tasa de transmisión es nula, es decir, $S_1 \geq 1 - \beta_m$. Esta curva se muestra graficada en la figura 5.9 como una línea punteada y representa la contribución fundamental a S_1 en los medios

dicotómicos, lo que explica el comportamiento lineal.

Resumiendo, los resultados obtenidos son los siguientes (Tabla 5.4):

- Si la tasa de transmisión media espacial es la misma, la velocidad del frente de infección es la *mayor* sobre el medio DC.
- Si la tasa de transmisión media espacial es la misma y $\beta_m > 0.35$, la amplitud media del frente de infección es *mayor* sobre el medio H.
- Si la tasa de transmisión media espacial es la misma y $\beta_m < 0.35$, la amplitud media del frente de infección es *mayor* sobre el medio DC.
- La tasa de transmisión crítica que define el umbral de propagación es la *menor* sobre el medio DC y la *mayor* sobre el medio H.
- Si $\beta_m > 0.2$, el frente de infección *más nocivo* (menor S_1) se da sobre el medio H.
- Si $\beta_m < 0.2$, el frente de infección *más nocivo* (menor S_1) se da sobre el medio DC.
- Sobre los medios dicotómicos DC y DA, la *nocividad* S_1 es aproximadamente lineal con β_m .

Tabla 5.4: Ranking esquemático de heterogeneidades en función de los observables c , I_{max} y S_1 ordenado de mayor a menor.

Posición	c	$I_{max}(\beta_m > 0.35)$	$I_{max}(\beta_m < 0.35)$	S_1
1°	DC	H	DC	DA
2°	DA	S	S	DC
3°	S	DC	DA	S
4°	H	DA	H	H

Capítulo 6

Geometría del frente de infección

En este capítulo exploramos en profundidad las propiedades microscópicas del frente de infección, buscando describir la dinámica fundamental y universal del mismo. Nuevamente, estaremos resolviendo el sistema del modelo SIR espacial heterogéneo propuesto en el capítulo 3. Es decir, el sistema

$$\partial_t S = -\beta_{\mathbf{r}} SI + D_S \nabla^2 S, \quad (6.1)$$

$$\partial_t I = \beta_{\mathbf{r}} SI - \gamma I + D_I \nabla^2 I, \quad (6.2)$$

con $(x, y) \in [0, L_x] \times [0, L_y]$ y condiciones iniciales

$$I(x, y, 0) = \begin{cases} I_0 & \text{si } (x, y) \in \Omega_0 \\ 0 & \text{si } (x, y) \notin \Omega_0 \end{cases}; \quad S(x, y, 0) = \begin{cases} 1 - I_0 & \text{si } (x, y) \in \Omega_0 \\ S_0 & \text{si } (x, y) \notin \Omega_0. \end{cases}$$

donde $\Omega_0 = (0, \delta x) \times [0, Ly]$, es decir, la condición inicial consiste en un frente de infección plano. En cuanto a las condiciones de contorno tomamos condiciones periódicas en la dirección y y condiciones de Dirichlet en la dirección x dadas por

$$I(0, y, t) = I(L_x, y, t) = S(0, y, t) = S(L_x, y, t) = 0.$$

Esta vez nos centraremos únicamente en obtener resultados sobre el medio DA. El tamaño de los sistemas utilizado para algunos resultados de este capítulo alcanzó ($L_x = 2^{16}, L_y = 2^{11}$). Esto fue necesario para lograr alcanzar el tiempo de saturación t_x del campo de desplazamiento del frente de infección.

En lo que sigue mostraremos resultados del ancho $\omega(L_y, t)$ y el factor de estructura $S(q, t)$ de la interfaz, a partir de lo cual intentaremos determinar la clase de universalidad que la describe. Por otro lado, mostramos que una potencial ecuación de evolución para el campo de desplazamiento debe tener un término no lineal de tipo $(\nabla u)^2$, que concuerda con la clase de universalidad encontrada.

6.1. Ancho del frente de infección

6.1.1. Diferencias por definición del campo de desplazamiento

Comenzamos por estudiar el ancho del campo de desplazamiento $u(y, t)$ que define la interfaz del frente de infección. Recordemos que el ancho $\omega(L_y, t)$, dado $u(y, t)$, es

$$\omega^2(L_y, t) = \langle (u(y, t) - u_{cm}(t))^2 \rangle_y, \quad (6.3)$$

donde $u_{cm}(t) = \langle u(y, t) \rangle_y$ y $\langle \dots \rangle_y$ toma la media espacial sobre la coordenada y . En el capítulo anterior utilizamos como campo de desplazamiento, que llamaremos ahora $u_0(y, t)$, a las posiciones x que maximizan $I(x, y, t)$ dados y y t , es decir,

$$\max_x[I(x, y, t)] = I(u_0(y, t), y, t) \quad (6.4)$$

En este capítulo, tendremos en consideración además una definición alternativa para el campo de desplazamiento, dada por

$$u_1(y, t) = \frac{\int_0^{L_x} x I(x, y, t)}{\int_0^{L_x} I(x, y, t)}. \quad (6.5)$$

La diferencia fundamental entre estas definiciones, es que en un espacio discretizado el campo $u_0(y, t)$ está limitado a tomar los valores de la discretización en x . En cambio, el campo $u_1(y, t)$ es continuo y puede tomar cualquier valor entre 0 y L_x . A continuación veremos que hay ciertas diferencias entre los resultados obtenidos con cada una de estas definiciones y que se deben tener en cuenta algunas sutilezas para que los resultados obtenidos sean consistentes entre sí.

Obsérvese la figura 6.1, donde mostramos los resultados del ancho de la interfaz con las distintas definiciones del campo de desplazamiento, $u_0(y, t)$ y $u_1(y, t)$, sobre un sistema de $L_x = 2^{16} \times L_y = 2^{11}$. Se realizaron 150 realizaciones utilizando un medio DA con $p = 0.15$, $\beta = 1$, $\gamma = 0.2$, $D_S = 0$ y $D_I = 1$. Denotamos por ω_0 y ω_1 los anchos de $u_0(y, t)$ y $u_1(y, t)$, respectivamente. Recordemos también que $\overline{\dots}$ denota promedio sobre realizaciones. Se observa que el ancho de la interfaz, en ambos casos, crece siguiendo una ley de potencias con el tiempo hasta llegar a un valor de saturación, a partir del cual ya no crece más. Sin embargo, es claro que las curvas de $\overline{\omega_0}$ y $\overline{\omega_1}$ difieren notablemente, de hecho se observa que la pendiente en log-log es distinta, de modo que se podría pensar que la rugosidad del frente evoluciona con exponentes de crecimiento α/z distintos dependiendo de la definición que utilicemos para la interfaz. Lo que realmente está sucediendo es que no estamos teniendo en cuenta el ancho intrínseco ω_i [2, 49, 50] asociado a la definición de $u_0(y, t)$, que no es otro que la discretización del espacio en x , igual a la unidad en este caso, es decir, $\omega_i = 1$. Si sustraemos ω_i de $\overline{\omega_0}$ obtenemos la

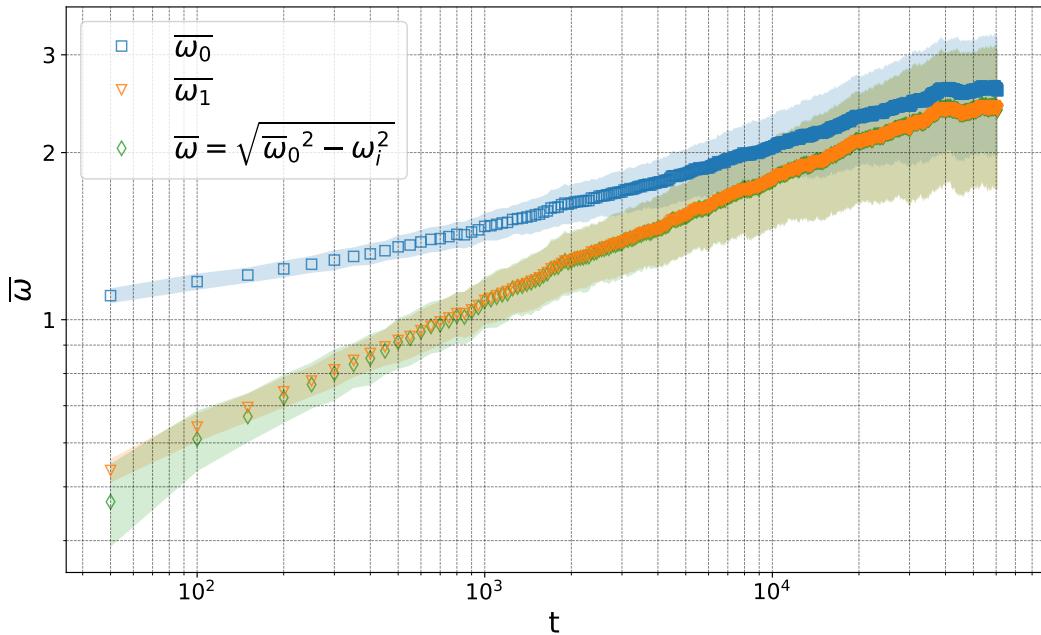


Figura 6.1: Se muestran los promedios del ancho del campo de desplazamiento, $\bar{\omega}_0$ y $\bar{\omega}_1$, correspondientes a distintas definiciones del mismo, $u_0(u, t)$ y $u_1(y, t)$ respectivamente, sobre 150 simulaciones en un sistema de tamaño $L_x = 2^{16} \times L_y = 2^{11}$. Se observa que al sustraer el ancho intrínseco $\omega_i = 1$ de $\bar{\omega}_0$ se obtiene $\bar{\omega}$, una curva similar (dentro del error) a la de $\bar{\omega}_1$ que representa correctamente la dinámica del crecimiento rugoso. Se utilizaron los parámetros $\beta = 1$, $\gamma = 0.2$, $D_I = 1$, $D_S = 0$ y $p = 0.15$.

curva $\bar{\omega}$ que es similar a la curva de $\bar{\omega}_1$ dentro del error. Esto muestra que para obtener los exponentes de universalidad que describen correctamente la interfaz, es necesario tener en cuenta el ancho intrínseco al utilizar $u_0(y, t)$. Nótese, sin embargo, que $\bar{\omega}_0$ y $\bar{\omega}_1$ comienzan a juntarse para valores grandes de t ($t > 10^4$). Esto es así porque en la medida que pasa el tiempo, mayor es el ancho para ambas definiciones y entonces el ancho intrínseco se vuelve despreciable. Si utilizáramos un sistema con un tamaño de interfaz L_y aún más grande y con suficiente tiempo de simulación veríamos que después de determinado tiempo las curvas de $\bar{\omega}_1$ y $\bar{\omega}_0$ efectivamente se juntan y describen los mismos exponentes de crecimiento. Con sistemas de tamaño $L_y = 2048$ como el que usamos acá vemos que alcanzamos la saturación antes de que pueda suceder esto, de modo que no podemos usar $\bar{\omega}_0$ para estimar los exponentes de universalidad.

Por otro lado, el hecho de que el ancho sature a un valor finito, recordando lo discutido en el capítulo 2, indica que hay una correlación entre puntos de la interfaz, la cual se extiende lateralmente, caracterizada por una longitud de correlación $\xi_{||}(t)$ que crece hasta alcanzar el tamaño del sistema, momento en el cual satura el ancho de la interfaz. Este fenómeno permite descartar una dinámica de tipo Poisson para la interfaz, ya que en este caso el ancho no satura.

Podríamos intentar obtener el exponente de crecimiento α/z que describe la ley de potencias con que crece el ancho $t^{\alpha/z}$ a partir de los resultados de la figura. Pero no es

el método más aconsejable para determinar los exponentes de universalidad, dado que el ancho $\omega(L_y, t)$ es una cantidad integrada sobre todas las escalas, de modo que esto nos daría un exponente de crecimiento efectivo. Es más seguro obtener los exponentes de un observable como el factor de estructura, tal como haremos más adelante, que nos permite discriminar distintas escalas. Sin embargo, para no quedarnos con un solo método, a continuación determinamos los exponentes de rugosidad α y dinámicos z , utilizando los datos del ancho de la interfaz con diferentes tamaños de sistema L_y .

6.1.2. Exponente de rugosidad y dinámico

Para determinar los exponentes de rugosidad y dinámico de la interfaz, debemos recordar algunas leyes de potencias discutidas en el capítulo 2. Si t_x , el tiempo de saturación, es el tiempo en el cual el ancho de la interfaz satura, entonces

$$\omega(L_y, t \gg t_x) = \omega_{sat}(L_y) \propto L_y^\alpha, \quad (6.6)$$

donde α es el exponente de rugosidad de la interfaz. Mientras que,

$$t_x \propto L_y^z, \quad (6.7)$$

donde z es el exponente dinámico.

En la figura 6.2 se muestra $\bar{\omega}_1(L_y, t)$ para distintos tamaños de interfaz L_y con $L_x = 2^{16}$ fijo. El tamaño de L_x se eligió así para alcanzar el tiempo de saturación t_x , dado que el mismo crece como L_y^z lleva mucho tiempo de simulación en sistemas grandes y por lo tanto se necesita mucho espacio a recorrer en x para alcanzarlo. Se determinó el ancho de saturación $\bar{\omega}_{sat}(L_y)$ para cada L_y , este se indica en línea a trazos horizontal en la figura 6.2. Luego, determinamos los tiempos de saturación t_x para cada L_y , se consideró como tal al mínimo valor de tiempo en el cual el ancho de la interfaz se encuentra a menos de una desviación estándar del valor de saturación $\omega_{sat}(L_y)$.

En las figuras 6.3a y 6.3b se muestran $\bar{\omega}_{sat}$ y t_x respectivamente, como función de L_y . Se observan los ajustes de las leyes de potencia 6.6 y 6.7, a partir de los cuales se determinó los exponentes de rugosidad α y dinámico z . Los resultados se muestran en la tabla 6.1.

El exponente de rugosidad α se calculó con $\bar{\omega}_0$, $\bar{\omega}_1$ y $\bar{\omega}$. En la figura 6.3a se observa nuevamente cómo la curva asociada a $\bar{\omega}_0$ difiere de las curvas de $\bar{\omega}_1$ y $\bar{\omega}$ dando lugar a un exponente de rugosidad erróneo como consecuencia de no ignorar el ancho intrínseco. Más aún, fíjese que el ajuste por ley de potencia con $\bar{\omega}_0$ es “pobre” en comparación con los de $\bar{\omega}_1$ y $\bar{\omega}$, es decir, no parece una recta, más bien parece una curva que asintóticamente se acerca a las curvas de $\bar{\omega}_1$ y $\bar{\omega}$. Lo cual tiene sentido, nuevamente, en tanto mayor sea el sistema menor es la contribución del ancho intrínseco al ancho total,

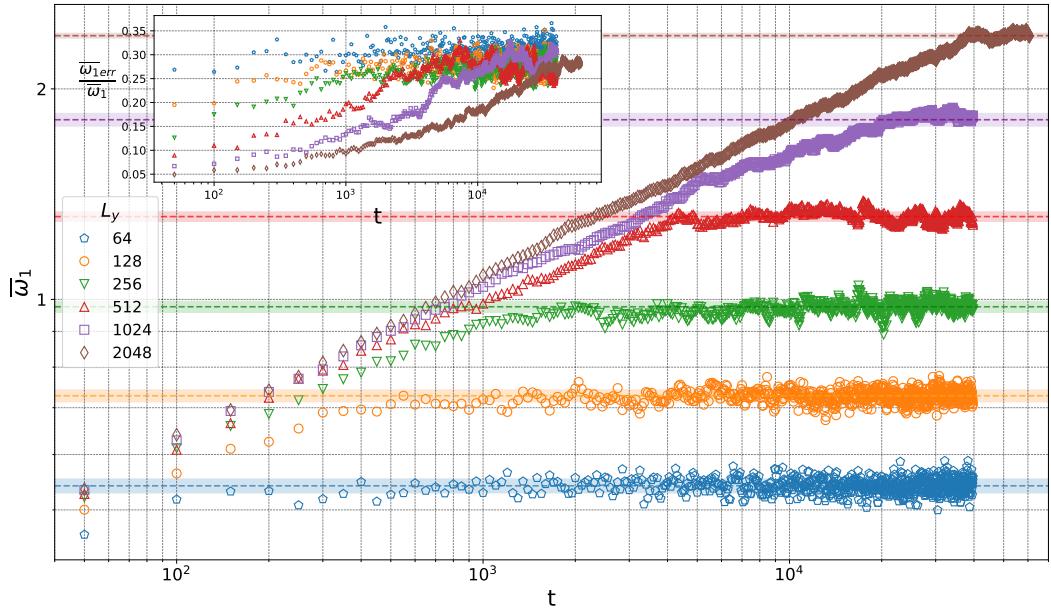


Figura 6.2: Ancho del frente $\bar{\omega}_1$ para diferentes tamaños L_y del sistema con $L_x = 2^{16}$. Se observa que el ancho del frente sigue una ley de potencias hasta alcanzar un valor de saturación que se indica con línea a trazos horizontal (---) y varía con L_y . En el *inset* se muestra el error relativo de cada curva. En sombras de colores se muestra una desviación estándar del valor de saturación para cada curva. Se promedió sobre 150 realizaciones de desorden. Se utilizaron los parámetros $\beta = 1$, $\gamma = 0.2$, $D_I = 1$, $D_S = 0$ y $p = 0.15$.

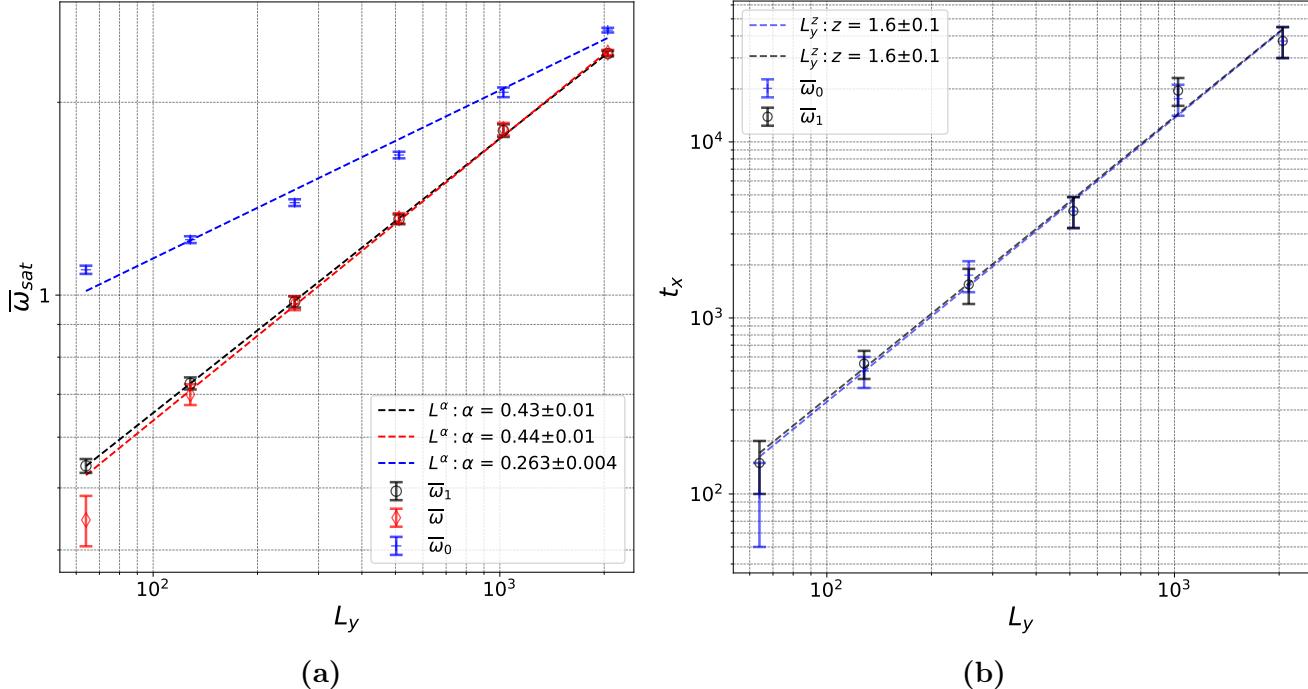


Figura 6.3: (a) Ancho de saturación $\bar{\omega}_{sat}$, usando $\bar{\omega}_0$, $\bar{\omega}_1$ y $\bar{\omega}$, en función del tamaño L_y de la interfaz. Se muestran los ajustes de potencia L^α con los que se determinó el exponente de rugosidad α . (b) Tiempo de saturación t_x del ancho de la interfaz, $\bar{\omega}_0$ y $\bar{\omega}_1$, en función del tamaño L_y de la interfaz. Se muestran los ajustes de potencia L_y^z con los que se determinó el exponente dinámico z .

	α	z
$\bar{\omega}_1$	0.43 ± 0.01	1.6 ± 0.1
$\bar{\omega}$	0.45 ± 0.01	1.6 ± 0.1

Tabla 6.1: Exponentes de rugosidad y dinámico obtenidos de los ajustes $\omega_{sat}(L_y) \propto L_y^\alpha$ y $t_x \propto L_y^z$ respectivamente, para distintas definiciones del ancho de la interfaz.

por lo cual es de esperar que en sistemas grandes ($L_y > 2048$) $\bar{\omega}_0$ describa correctamente la evolución de $\bar{\omega}_{sat}$. Por otro lado, observamos que el ancho intrínseco no interfiere a la hora de obtener el exponente dinámico dado que no afecta al tiempo de saturación t_x (ver figura 6.3b), por lo que $\bar{\omega}_0$ y $\bar{\omega}_1$ dan exponentes dinámicos similares.

De las clases de universalidad discutidas en el capítulo 2 vemos que el exponente dinámico z obtenido aquí únicamente concuerda con KPZ, $z_{KPZ} = \frac{3}{2}$. Sin embargo, el exponente de rugosidad $\alpha \approx 0.42$ determinado no es el mismo que en KPZ, $\alpha_{KPZ} = \frac{1}{2}$. Esto podría deberse a que quizás estamos subestimando los valores de saturación del ancho en los sistemas de mayor tamaño, como $L_y = 1024, 2048$, y que quizás sea necesario llegar a un mayor tiempo de simulación para obtener estos valores más precisamente.

En la figura 6.4 se muestra el colapso de $\bar{\omega}_1$ que se obtiene escaleando $\bar{\omega}_1$ por L^α y el tiempo t por L^z . Para ello se utilizó los exponentes determinados previamente con $\bar{\omega}_1$. Se observa que el colapso es bueno, lo que indica que los exponentes determinados con estos datos son acertados.

En resumen, determinamos los exponentes de rugosidad α y dinámico z , y obser-

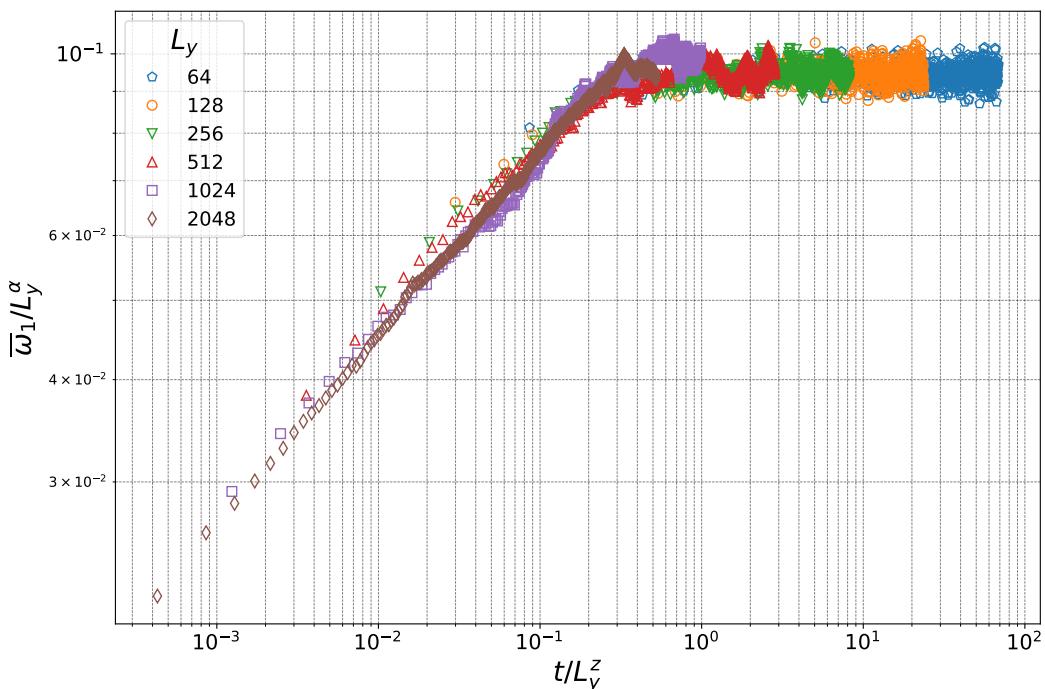


Figura 6.4: Se muestra el colapso de las curvas para el ancho $\bar{\omega}_1$ al dividir por L_y^α y escalar el tiempo por L^z .

vamos que el exponente dinámico concuerda con el exponente dado por KPZ, aunque el exponente de rugosidad se encuentra subestimado. Sin embargo, la coincidencia del exponente dinámico nos motiva a pensar en la idea de que la dinámica del frente puede estar descrita por un término no lineal $(\nabla u)^2$ de tipo KPZ. A continuación intentaremos verificar esta hipótesis observando si hay alguna dependencia de la velocidad del frente con la inclinación del mismo.

6.2. Velocidad con frente inclinado

Para verificar si la dinámica de la interfaz presenta algún término no lineal debemos recordar nuevamente lo visto en el capítulo 2 respecto a la velocidad de una interfaz descrita por la ecuación de KPZ (ecuación 2.17). Ahí vimos que la velocidad media de dicho frente depende del término no lineal de la siguiente manera (ecuación 2.25):

$$v = v_0 + \frac{\lambda}{2L} \int_0^L d^d \mathbf{x} \overline{(\nabla h)^2}. \quad (6.8)$$

Reemplazando por la notación que hemos estado utilizando para la velocidad del frente de infectados c y el campo de desplazamiento $u(y, t)$, esto es,

$$c = c_0 + \frac{\lambda}{2L_y} \int_0^{L_y} dy \overline{(\partial_y u)^2}. \quad (6.9)$$

De modo que podemos forzar la interfaz a tener una inclinación determinada m en la condición inicial, de manera tal que $\partial_y u \approx m$. Luego, al reemplazar esto en 6.9, obtenemos que la velocidad del frente depende cuadráticamente con la inclinación del mismo,

$$c \simeq c_0 + \frac{\lambda}{2} m^2, \quad (6.10)$$

donde c_0 sería la velocidad del frente con $m = 0$.

Recordemos que esto es así solo si existe un término no lineal que contribuye a la dinámica de la interfaz. De otra manera, la velocidad del frente permanece inalterada ante una inclinación del mismo [2, 51].

Para verificar esto alcanzó con trabajar con sistema “chicos” de 1024×1024 , se utilizó un medio DA con $\gamma = 0.2$, $D_S = 0$, $D_I = 1$, $p = 0.15$ y distintos valores de β . En la figura 6.5a se muestra la velocidad del frente en función de la inclinación inicial del mismo para distintos valores de β . Efectivamente, encontramos que la velocidad depende cuadráticamente con la pendiente del frente, lo que prueba la existencia de una componente no lineal en la dinámica del mismo.

Más aún, al realizar las simulaciones con distintos valores de β pudimos determinar la constante λ de KPZ ajustando las paráolas y encontramos que λ es proporcional

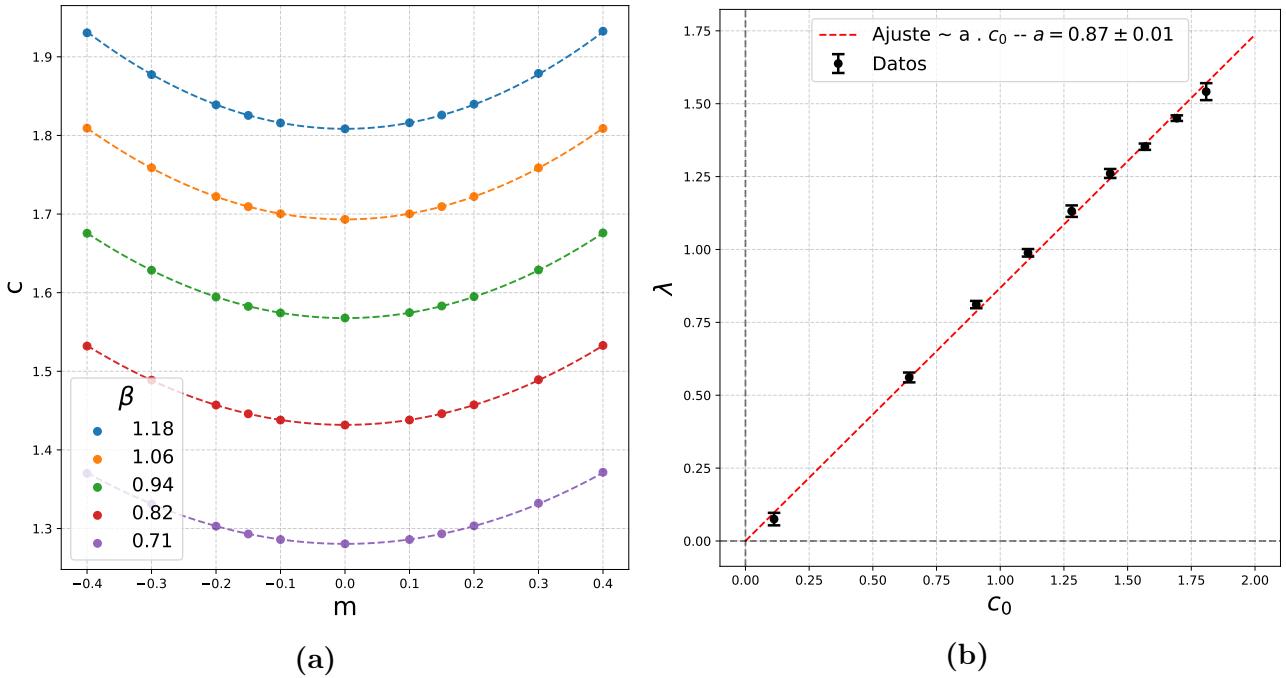


Figura 6.5: (a) Velocidad del frente c como función de la inclinación inicial del mismo m para distintos valores de β . Se observa una dependencia cuadrática. Se utilizó un sistema de 1024×1024 con $\gamma = 0.2$, $D_S = 0$, $D_I = 1$ y $p = 0.15$. (b) λ como función de la velocidad del frente sin inclinación c_0 , se observa que $\lambda \propto c_0$.

a la velocidad c_0 , de hecho $\lambda \approx c_0$, tal como se muestra en la figura 6.5b. Esto implica que el efecto que tiene el término no lineal, propio de KPZ, depende directamente de la velocidad del frente, cumpliéndose que $\lambda \rightarrow 0$ cuando $c_0 \rightarrow 0$, característico de la clase de universalidad de KPZ con origen cinético [2]. Esta podría ser otra razón por la cual el exponente de rugosidad α fue subestimado en la sección 6.1.2, es decir: dado que λ depende de la velocidad del frente, podría ser que la velocidad, fijada por β , γ , D_I y p , no sea lo suficientemente alta como para observar el efecto de KPZ en la escala de tiempo que se utilizó para medir $\bar{\omega}_{sat}(L_y)$ y determinar el exponente de rugosidad.

6.3. Factor de estructura

Aquí mostramos una metodología alternativa para determinar los exponentes de universalidad utilizando el factor de estructura $S(q, t)$ dado por

$$S(q, t) = \overline{\tilde{u}(q, t)\tilde{u}(-q, t)}, \quad (6.11)$$

donde $\tilde{u}(q, t)$ es la transformada de Fourier del campo de desplazamiento $u(y, t)$. Recordamos una vez más del capítulo 2, que el factor de estructura con una condición

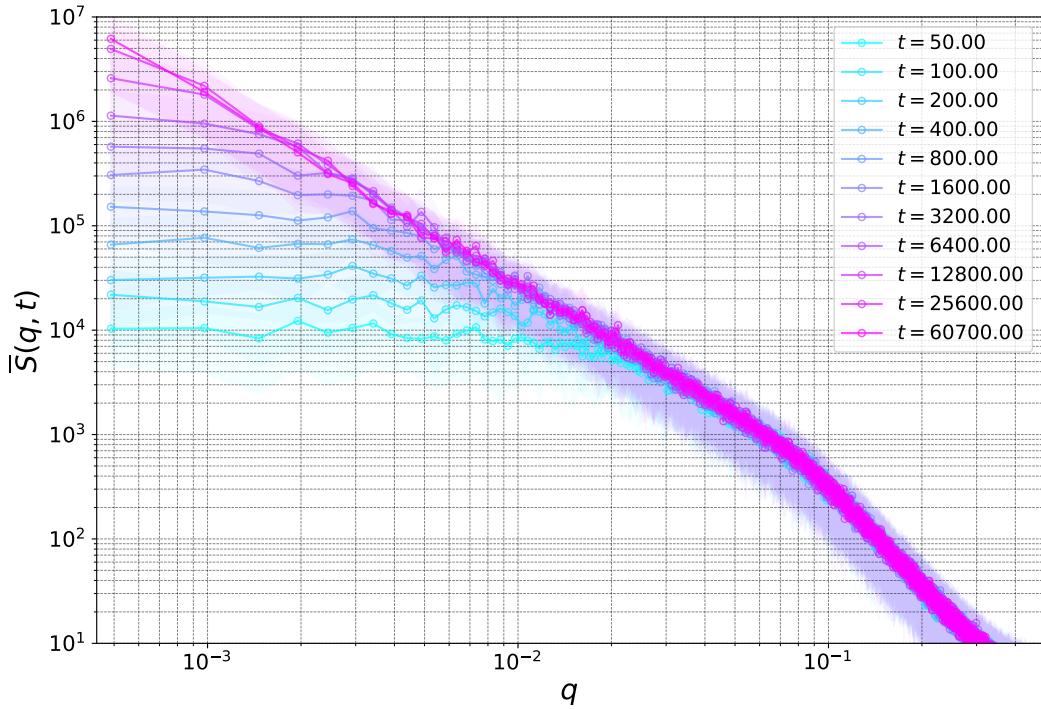


Figura 6.6: Factor de estructura $\bar{S}(q, t)$ de la interfaz en distintos instantes de tiempo promediado sobre 150 realizaciones y utilizando la definición $u_1(y, t)$ para el campo de desplazamiento.. Se utilizó un sistema de tamaño $L_y = 2^{11} \times L_x = 2^{16}$ con $\gamma = 0.2$, $D_S = 0$, $D_I = 1$, $p = 0.15$ y $\beta = 1$. Las sombras de color indican la desviación estándar sobre las realizaciones.

inicial plana para la interfaz cumple que,

$$S(q, t) \propto \begin{cases} t^{(1+2\alpha)/z} & \text{si } q \ll \xi_{||}(t)^{-1} \\ q^{-(1+2\alpha)} & \text{si } q \gg \xi_{||}(t)^{-1}. \end{cases} \quad (6.12)$$

A partir de estas leyes de potencia podremos determinar los exponentes de universalidad α y z . Para empezar, observamos la figura 6.6 donde se muestra el factor de estructura $\bar{S}(q, t)$ de la interfaz en distintos instantes de tiempo promediado sobre 150 realizaciones y utilizando la definición $u_1(y, t)$ para el campo de desplazamiento. Se utilizó un sistema de tamaño $L_y = 2^{11} \times L_x = 2^{16}$ con $\gamma = 0.2$, $D_S = 0$, $D_I = 1$, $p = 0.15$ y $\beta = 1$. Se observa precisamente el comportamiento descrito por la ecuación 6.12. Es decir, para escalas mayores a la longitud de correlación $\xi_{||}(t)$ el sistema se encuentra descorrelacionado y muestra un espectro de ruido blanco de amplitud creciente con el tiempo, mientras que a distancias menores a $\xi_{||}(t)$ el sistema se encuentra correlacionado y exhibe un espectro de potencias. Cuando la longitud de correlación $\xi_{||}(t)$ alcanza el tamaño del frente L_y , el sistema se encuentra completamente correlacionado y queda en un estado estacionario.

Para determinar el exponente de rugosidad α definimos el factor de estructura en el estado estacionario, promediando el mismo cuando el tiempo es mayor al tiempo de

saturación $t > t_x$, es decir,

$$\bar{S}_{est}(q) = \langle \bar{S}(q, t > t_x) \rangle_x. \quad (6.13)$$

De esta manera aprovechamos el hecho de que el factor de estructura no cambia en el tiempo cuando $t > t_x$ para obtener una medida más precisa del mismo. Los resultados se muestran en la figura 6.7a, utilizando $u_0(y, t)$ y $u_1(y, t)$ y ajustando la ley de potencias $q^{-(1+2\alpha)}$ para grandes escalas ($q < 10^{-2}$) obtenemos $\alpha = 0.40(2)$ con $u_0(y, t)$ y $\alpha = 0.42(2)$ con $u_1(y, t)$. Ambos resultados compatibles entre sí y con los exponentes de rugosidad determinados en la sección 6.1.2. Sin embargo, nuevamente quedan por debajo de $\alpha_{KPZ} = \frac{1}{2}$ que predice KPZ. Esto puede deberse, como mencionamos en la sección 6.1.2, a que quizás no estamos alcanzando plenamente el estado estacionario, lo cual reduce la pendiente de la ley de potencias y da lugar a una subestimación del exponente de rugosidad.

Es importante remarcar el hecho de que determinamos estos exponentes de rugosidad en escalas grandes del sistema $q < 10^{-2}$ (distancias mayores a 100 unidades espaciales), es decir, no obtenemos la misma ley de escalamiento a todas las escalas, en el *inset* de la figura 6.7a puede observar cómo varía la estimación del exponente de rugosidad al incluir escalas cada vez más chicas, esto muestra la conveniencia de utilizar el $\bar{S}(q, t)$ para estimar los exponentes por sobre el ancho $\bar{\omega}(L_y, t)$. Por otro lado, este podría ser otro motivo de la subestimación de α , quizás sea necesario trabajar con interfaces más grandes ($L_y > 2048$) para observar efectivamente un exponente de tipo KPZ y lo que está sucediendo aquí es que estamos observando un *crossover* entre el espectro microscópico y el macroscópico.

Nótese también de la figura 6.7a que el factor de estructura proveniente de $u_0(y, t)$ coincide con el dado por $u_1(y, t)$ a grandes escalas $q < 10^{-2}$ (distancias mayores a 100 unidades espaciales). Sin embargo, para $q > 10$ las curvas se separan, potencialmente debido nuevamente a que al acercarnos al tamaño de discretización la medición del factor de estructura con $u_0(y, t)$ se ve comprometida.

Por otro lado, para determinar el exponente dinámico determinamos los valores del factor de estructura en escalas mayores a la longitud de correlación $\xi_{||}(t)$, es decir,

$$\bar{S}_{sat}(t) = \bar{S}(q < \xi_{||}^{-1}(t), t), \quad (6.14)$$

donde debe cumplirse que

$$\bar{S}_{sat}(t) \propto t^{(1+2\alpha)/z}. \quad (6.15)$$

Esto es lo que se muestra en la figura 6.7b. Se realizó el ajuste en dos regiones distintas, ya que se observó un *crossover* en $t \in (2000, 3000)$, donde pareciera saturar, pero luego sigue creciendo. Los exponentes dinámicos obtenidos fueron $z = 1.64 \pm 0.12$ y

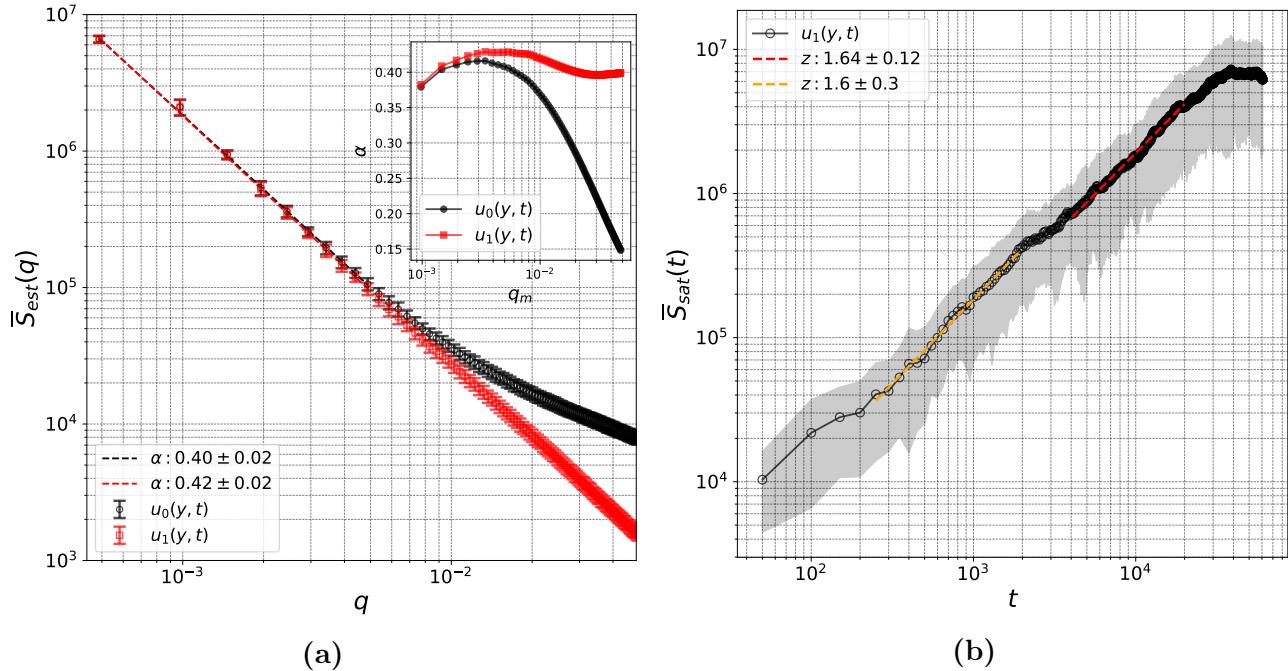


Figura 6.7: (a) Factor de estructura en el estado estacionario $\bar{S}_{est}(t) = \langle \bar{S}(q, t > t_x) \rangle_t$ promediado sobre 150 realizaciones. Se muestra el resultado utilizando $u_0(y, t)$ y $u_1(y, t)$. Se realizó el ajuste dado por la ley de potencia $q^{-(1+2\alpha)}$ para determinar el exponente de rugosidad a grandes escalas ($q \leq 10^{-2}$). En el *inset* se muestra cómo varía el exponente de rugosidad al ajustar la ley de potencias hasta q_m . (b) Valor de saturación del factor de estructura en función del tiempo $\bar{S}_{sat}(t) = \bar{S}(q < \xi_{||}^{-1}(t), t)$ promediado sobre 150 realizaciones. Se muestra solo el resultado con $u_1(y, t)$. Se realizó el ajuste con ley de potencia $t^{(1+2\alpha)/z}$ para determinar el exponente dinámico z . Se muestran dos regiones de ajuste porque en el medio se observa un *crossover*.

$z = 1.6 \pm 0.3$ que concuerdan con los obtenidos previamente y además son compatibles con el exponente dinámico $z_{KPZ} = \frac{3}{2}$ dado por KPZ.

Finalmente, para verificar visualmente la efectividad de los exponentes obtenidos, en la figura 6.8 mostramos el colapso del factor de estructura, dividiendo por $t^{(1+2\alpha)/z}$ y escaleando q por $t^{1/z}$.

Discusión

Los resultados obtenidos en este capítulo nos llevan a pensar que el frente de desplazamiento $u(y, t)$, cualquiera sea, dado por el modelo SIR heterogéneo con medio DA cerca del límite homogéneo ($p = 0.15$), puede describirse fundamentalmente y a grandes escalas por una ecuación de KPZ del tipo

$$\partial_t u = \nu \nabla^2 u + \frac{\lambda}{2} (\nabla u)^2 + c_0 + \eta(y, t). \quad (6.16)$$

Donde adicionalmente observamos que $\lambda \propto c_0$. Sin embargo, si bien encontramos concordancia con el exponente dinámico $z_{KPZ} = \frac{3}{2}$ (aunque sistemáticamente sobreestima-

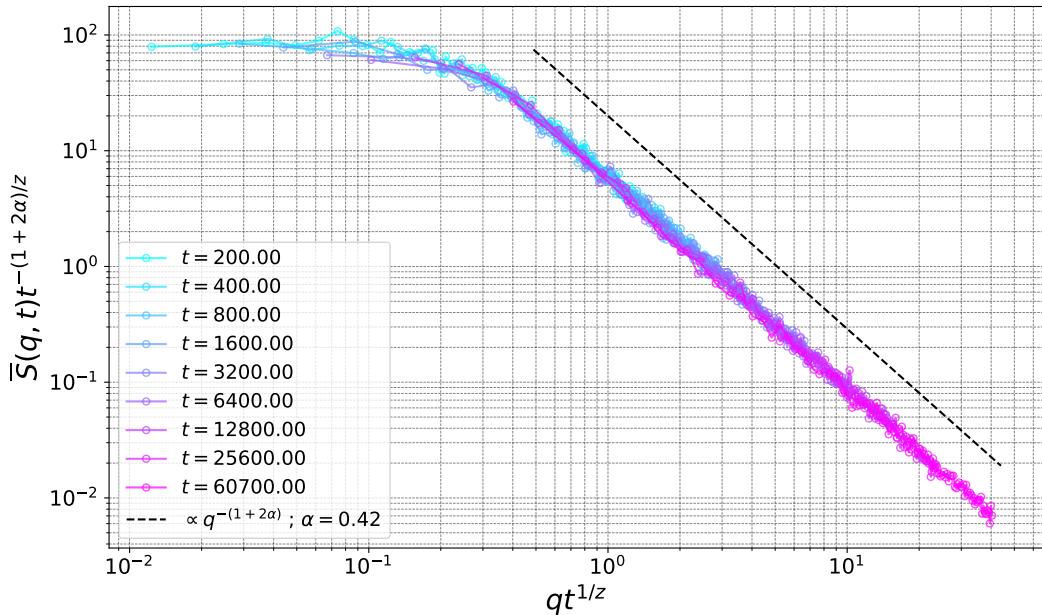


Figura 6.8: Colapso del factor de estructura dado por $u_1(y, t)$ utilizando los exponentes de universalidad $\alpha = 0.42 \pm 0.02$ y $z = 1.64 \pm 0.12$.

do), el exponente de rugosidad obtenido resultó menor que el dado por KPZ, $\alpha_{KPZ} = \frac{1}{2}$. Hay diversas razones que pueden explicar la posible subestimación del exponente de rugosidad.

Una de ellas, como mencionamos previamente, cae en el hecho de que posiblemente el tamaño de la interfaz utilizado $L_y = 2048$ no es lo suficientemente grande. Naturalmente, la pregunta entonces es ¿por qué no probamos con un tamaño mayor? Para empezar, pensamos que este tamaño iba a ser suficiente, pero más allá de eso hay una limitación importante, y es que necesitamos alcanzar el tiempo de saturación t_x para ajustar correctamente el exponente de rugosidad y el tiempo de saturación crece como $t_x \propto L_y^z$ y $z \geq 1.5$. De modo que cada vez que aumentamos el tamaño, el tiempo de saturación crece apreciablemente y por ende también el tiempo de simulación. Aun así, esto no es un problema realmente, ya que en teoría podríamos simular sistemas considerablemente más grandes ($L_y = 8192$) en un tiempo razonable, las GPUs lo permiten. Sin embargo, el problema es que para alcanzar el tiempo de saturación necesitamos no solo tiempo, sino mucho espacio en la dirección x , como mencionamos al principio del capítulo. Cuando resolvemos un sistema con una interfaz $L_y = 2048 = 2^{11}$ se necesita que, al menos, $L_x = 65536 = 2^{16}$, esto es una matriz de 2^{27} elementos que con un tipo de dato de 64 bits ocupa $2^{27} \times 8 = 2^{30}$ bytes, es decir, 1 GB. Y tenemos 5 matrices de estas más los datos que se van almacenando en la GPU para no hacer transferencias entre GPU y RAM, que son lentas. En definitiva, estamos casi en el límite de las memorias de las GPU convencionales. Ahora bien, cuento esto simplemente porque me resulta interesante, pero es posible pensar en más de una solución a este problema

bastante rápido. Lo que haría si tuviera que hacerlo nuevamente es evitar utilizar 5 matrices, se puede reducir a tres por lo menos. Otra alternativa muy accesible sería optar por trabajar con números de 32bits, la pérdida en precisión no me pareció significante considerando unas pruebas que hice.

Por otro lado, más allá de la discusión previa, también es posible que una interfaz de mayor tamaño no muestre ningún cambio en el exponente de rugosidad. En este caso hay otras razones que pueden explicarlo. La dinámica de KPZ resulta de la competencia entre el término difusivo y el término no lineal de 6.16, vimos que λ , la constante del término no lineal, depende de la velocidad del frente. Podría ser que la velocidad del caso particular analizado no sea lo suficientemente grande para competir con el término difusivo, lo cual explicaría por qué el exponente de rugosidad queda subestimado y el dinámico algo sobreestimado, en ese caso la ecuación se parecería más a una de tipo EW (ecuación 2.20), sin el término no lineal, en donde $z_{EW} = 2$ y $\alpha_{EW} = \frac{1}{2}$.

En cualquier caso, es interesante notar que la dinámica rugosa del campo de desplazamiento $u(y, t)$ que define la interfaz de un frente de infección en un modelo SIR como el discutido aquí presenta características similares a las descritas por la universalidad de KPZ con origen cinético. Por último, recordemos que la ecuación de KPZ es esencialmente estocástica mientras que el modelo SIR de difusión discutido en este trabajo es determinista: dada la misma tasa de transmisión espacial y las mismas condiciones iniciales, el sistema evolucionará siempre de la misma manera. Es decir, **el comportamiento universal emerge al observar la dinámica del frente de forma estadística sobre una variedad de β_r generados aleatoriamente**. Recordemos que en KPZ se asume un ruido $\eta(y, t)$ totalmente descorrelacionado en el espacio y el tiempo,

$$\overline{\eta(y, t)\eta(y', t')} = 2\kappa\delta(y - y')\delta(t - t'). \quad (6.17)$$

Sin embargo, el ruido emergente del modelo SIR espacial heterogéneo tiene correlaciones de corto alcance, espaciales y temporales, que pueden despreciarse solo cuando analizamos grandes interfaces a grandes tiempos. Es de esperar entonces que estas correlaciones introduzcan correcciones en los exponentes a “pequeñas” escalas y den lugar a exponentes efectivos.

Capítulo 7

Conclusiones

Comenzamos este trabajo (Capítulo 2) con una investigación en donde observamos, a modo general, la relevancia física que tienen las inferfases fuera del equilibrio. Mostramos una diversidad de ejemplos en la naturaleza donde la dinámica de un sistema queda caracterizada por una interfaz. Todos ellos ampliamente estudiados en la literatura, ya sea desde los frentes de incendio o epidemiológicos hasta las olas producidas por las *abejas gigantes* y el crecimiento de bacterias [3–10, 14–18]. Luego, se describieron las herramientas necesarias y discutimos conceptos de escaleo para poder cuantificar y describir precisamente las características fundamentales de las interfaces.

Pasamos de la visión general a un problema particular (Capítulo 3), orientado a describir la propagación de una epidemia en un medio heterogéneo, sustentado sobre un modelo de tipo SIR espacial con difusión (ecuaciones 3.22 - 3.23), en donde introdujimos heterogeneidad por medio de una tasa de transmisión espacialmente dependiente. Inicialmente, discutimos las soluciones de onda solitaria asociadas a este modelo en el caso homogéneo, con tasa de transmisión constante en el espacio, y determinamos la velocidad del frente de infección (ecuación 3.12), el umbral de propagación dado por una tasa de transmisión crítica (ecuación 3.21) y los perfiles asintóticos del mismo (ecuaciones 3.13 - 3.15).

Luego, nos dispusimos a estudiar el sistema con medios heterogéneos, con el objetivo de determinar la influencia de las heterogeneidades sobre el frente de infección. Primero investigamos los observables macroscópicos (Capítulo 5), en particular estudiamos la velocidad (3.27), la amplitud máxima (3.26) y la nocividad (5.9) del frente de infección en función de la tasa de transmisión media espacial β_m utilizando distintas heterogeneidades. Encontramos fenómenos críticos para todas ellas asociadas al umbral de propagación del frente y determinamos los exponentes críticos asociados. Observamos un fenómeno no trivial y emergente de aceleración del frente al usar medios heterogéneos respecto de medios homogéneos. Más aún, observamos que el umbral de propagación es sensible a la heterogeneidad, dando lugar a la propagación más fá-

cilmente en medios heterogéneos que en medios homogéneos. Es de relevancia notar una interpretación pragmática de estos resultados: imagine que se quiere evitar una epidemia reduciendo la tasa de transmisión por vacunación o aislamiento social, sería muy productivo saber cómo depende el umbral de propagación con la distribución espacial de la tasa de transmisión para diseñar una estrategia espacial de vacunación o aislamiento eficiente. De manera similar, si se quiere evitar la propagación de incendios utilizando cortafuegos podría diseñarse una buena estrategia para la ubicación espacial de los mismos.¹

Finalmente, nos centramos en estudiar propiedades microscópicas y universales del frente de infección con la heterogeneidad DA (Capítulo 6). Determinamos los exponentes de rugosidad y dinámico de la interfaz, los cuales definen la clase de universalidad del frente. Encontramos que el exponente dinámico coincide con el exponente dado por la clase de universalidad de KPZ. Por otro lado, verificamos la existencia de una contribución no lineal a la dinámica de la interfaz viendo si había alguna dependencia de la velocidad del frente con la inclinación del mismo. Más aún, encontramos que esta contribución es de origen cinético y que es proporcional a la velocidad del frente. Sin embargo, el exponente de rugosidad requiere de estudios más precisos dado que aparece ligeramente subestimado.

De todas formas, es importante notar que la dinámica rugosa del campo de desplazamiento que define la interfaz de un frente de infección en un modelo SIR como el discutido aquí presenta características similares a las descritas por la universalidad de KPZ con origen cinético. Recordemos que la ecuación de KPZ es esencialmente estocástica mientras que el modelo SIR de difusión discutido en este trabajo es determinista: dada la misma tasa de transmisión espacial y las mismas condiciones iniciales, el sistema evolucionara siempre de la misma manera. Es decir, **el comportamiento universal emerge al observar la dinámica del frente de forma estadística sobre una variedad de tasas de transmisión β_r generadas aleatoriamente**.

A futuro sería interesante intentar derivar una ecuación para el frente $u(y, t)$ utilizando técnicas como las que se usaron en Ref. [4]. Por otro lado, sería importante verificar si utilizando una interfaz de mayor tamaño se obtiene un exponente de rugosidad distinto y más cercano al de KPZ. Se podría intentar hacer una caracterización más exhaustiva de los exponentes de universalidad en función del parámetro de desorden p para el medio DA y estudiarlos también en con otras heterogeneidades. Por último, sería interesante también ver qué efecto tiene sobre la rugosidad de la interfaz incluir difusión en la ecuación de susceptibles, es decir, $D_S \neq 0$, dado que como comentamos en el Capítulo 3 esto afecta a la estabilidad del mismo y podría producir alteraciones no triviales en la interfaz.

¹Aclaro que no estamos diciendo que los resultados obtenidos con este modelo en particular sean útiles para casos reales, solo es para dar una idea cualitativa de la importancia.

Por otro lado, lo más atractivo probablemente sea extender las herramientas desarrolladas aquí para estudiar diversos modelos físicos, químicos, ecológicos, etc. que estén definidos con ecuaciones de reacción difusión y presenten algún tipo de heterogeneidad. Pueden encontrarse varios trabajos recientes al respecto, algunos de ellos son *Front roughening of flames in discrete media* [24] y *Fisher waves and front roughening in a two-species invasion model with preemptive competition* [25].

La completitud de los resultados obtenidos en este trabajo se obtuvieron utilizando una librería de *Python* desarrollada personalmente que se encuentra accesible en un repositorio abierto de *GitHub* (<https://github.com/RenzoZS/tesis>)². La misma precisa únicamente del paquete *CuPy* y una GPU compatible con *CUDA-toolkit* para poder ejecutarse, además de *Python* claro.

²Espero que se encuentre lo más documentado posible al momento de leer esto.

Bibliografía

- [1] Riley, S., Eames, K., Isham, V., Mollison, D., Trapman, P. **Five challenges for spatial epidemic models.** *Epidemics*, **10**, 68–71, 2015. Challenges in Modelling Infectious Disease Dynamics.
- [2] Barabási, A.-L., Stanley, H. E., *et al.* **Fractal concepts in surface growth.** Cambridge university press, 1995.
- [3] Zhang, J., Zhang, Y.-C., Alstrøm, P., Levinsen, M. **Modeling forest fire by a paper-burning experiment, a realization of the interface growth mechanism.** *Physica A: Statistical Mechanics and its Applications*, **189** (3-4), 383–389, 1992.
- [4] Provatas, N., Ala-Nissila, T., Grant, M., Elder, K., Piché, L. **Scaling, propagation, and kinetic roughening of flame fronts in random media.** *Journal of statistical physics*, **81** (3), 737–759, 1995.
- [5] Maunuksela, J., Myllys, M., Kähkönen, O.-P., Timonen, J., Provatas, N., Alava, M. J., *et al.* **Kinetic Roughening in Slow Combustion of Paper.** *Phys. Rev. Lett.*, **79**, 1515–1518, Aug 1997.
- [6] Jullien, R., Kertész, J., Meakin, P., Wolf, D. E. **Surface disordering : growth, roughening, and phase transitions.** 1992.
- [7] Yunker, P. J., Lohr, M. A., Still, T., Borodin, A., Durian, D. J., Yodh, A. G. **Effects of Particle Shape on Growth Dynamics at Edges of Evaporating Drops of Colloidal Suspensions.** *Phys. Rev. Lett.*, **110**, 035501, Jan 2013.
- [8] Kastberger, G., Hoetzl, T., Maurer, M., Kranner, I., Weiss, S., Weihmann, F. **Speeding up social waves. Propagation mechanisms of shimmering in giant honeybees.** *PloS one*, **9** (1), e86315, 2014.
- [9] Kastberger, G., Weihmann, F., Hoetzl, T. **Social waves in giant honeybees (*Apis dorsata*) elicit nest vibrations.** *Naturwissenschaften*, **100** (7), 595–609, 2013.
- [10] Kastberger, G., Schmelzer, E., Kranner, I. **Social waves in giant honeybees repel hornets.** *PLoS One*, **3** (9), e3141, 2008.

- [11] Farkas, I., Helbing, D., Vicsek, T. Mexican waves in an excitable medium. *Nature*, **419** (6903), 131–132, 2002.
- [12] Farkas, I., Helbing, D., Vicsek, T. Human waves in stadiums. *Physica A: statistical mechanics and its applications*, **330** (1-2), 18–24, 2003.
- [13] ViralHog. Honeybees Mesmerizing Defensive Wave. *Youtube*, 2019.
- [14] Matsushita, M., Fujikawa, H. Diffusion-limited growth in bacterial colony formation. *Physica A: Statistical Mechanics and its Applications*, **168** (1), 498–506, 1990.
- [15] Bhattacharjee, T., Amchin, D. B., Alert, R., Ott, J. A., Datta, S. S. Chemotactic smoothing of collective migration. *Elife*, **11**, e71226, 2022.
- [16] Abramson, G. Tópicos de Biología Matemática. págs. 203–208, 2018.
- [17] Barbieri, R., Texier, G., Keller, C., Drancourt, M. Soil salinity and aridity specify plague foci in the United States of America. *Scientific reports*, **10** (1), 1–9, 2020.
- [18] Stenseth, N. C., Atshabar, B. B., Begon, M., Belmain, S. R., Bertherat, E., Carniel, E., et al. Plague: past, present, and future. *PLoS medicine*, **5** (1), e3, 2008.
- [19] Farahzadi, A. Systematic studies on structural and optical properties of organic thin films on inorganic substrates. Tesis Doctoral, Aachen, Techn. Hochsch., Diss., 2007, 2008.
- [20] Family, F., Vicsek, T. Scaling of the active zone in the Eden process on percolation networks and the ballistic deposition model. *Journal of Physics A: Mathematical and General*, **18** (2), L75, feb 1985.
- [21] Bustingorry, S., Guyonnet, J., Paruch, P., Agoritsas, E. A numerical study of the statistics of roughness parameters for fluctuating interfaces. *Journal of Physics: Condensed Matter*, **33**, 06 2021.
- [22] Family, F. Scaling of rough surfaces: effects of surface diffusion. *Journal of Physics A: Mathematical and General*, **19** (8), L441, jun 1986.
- [23] Kardar, M., Parisi, G., Zhang, Y.-C. Dynamic Scaling of Growing Interfaces. *Phys. Rev. Lett.*, **56**, 889–892, Mar 1986.
- [24] Lam, F., Mi, X., Higgins, A. J. Front roughening of flames in discrete media. *Phys. Rev. E*, **96**, 013107, Jul 2017.

- [25] O’Malley, L., Kozma, B., Korniss, G., Rácz, Z., Caraco, T. Fisher waves and front roughening in a two-species invasion model with preemptive competition. *Phys. Rev. E*, **74**, 041116, Oct 2006.
- [26] Glatter, K. A., Finkelman, P. History of the plague: An ancient pandemic for the age of COVID-19. *The American journal of medicine*, **134** (2), 176–181, 2021.
- [27] Cesana, D., Benedictow, O., Bianucci, R. The origin and early spread of Black Death in Italy: first evidence of plague victims from 14th century Liguria (Northern Italy). *Anthropological Science (Japanese Series)*, **125**, 03 2017.
- [28] Christakos, G., Olea, R. A., Serre, M. L., Wang, L.-L., Yu, H.-L. *Interdisciplinary public health reasoning and epidemic modelling: the case of black death*. Springer, 2005.
- [29] Murray, J. D. *Mathematical Biology II: Spatial Models and Biomedical Applications*, tomo 18 de *Interdisciplinary Applied Mathematics*. Springer New York, 2003.
- [30] Taubenberger, J. K. The origin and virulence of the 1918 “Spanish” influenza virus. *Proceedings of the American Philosophical Society*, **150** (1), 86, 2006.
- [31] Worobey, M., Han, G.-Z., Rambaut, A. Genesis and pathogenesis of the 1918 pandemic H1N1 influenza A virus. *Proceedings of the National Academy of Sciences*, **111** (22), 8107–8112, 2014.
- [32] Mills, C. E., Robins, J. M., Lipsitch, M. Transmissibility of 1918 pandemic influenza. *Nature*, **432** (7019), 904–906, 2004.
- [33] Kermack, W. O., McKendrick, A. G., Walker, G. T. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, **115** (772), 700–721, 1927.
- [34] Keeling, M. J., Rohani, P. *Modeling Infectious Diseases in Humans and Animals*. págs. 15–26. Princeton University Press, 2008.
- [35] Harko, T., Lobo, F. S., Mak, M. Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates. *Applied Mathematics and Computation*, **236**, 184–194, 2014.
- [36] Provatas, N., Ala-Nissila, T., Grant, M., Elder, K., Piché, L. Flame propagation in random media. *Physical Review E*, **51** (5), 4232, 1995.

- [37] Ipsen, M., Kramer, L., Sørensen, P. G. **Amplitude equations for description of chemical reaction–diffusion systems.** *Physics Reports*, **337** (1-2), 193–235, 2000.
- [38] Field, R. J. **Oscillations and traveling waves in chemical systems.** Wiley, 1985.
- [39] Horváth, D., Petrov, V., Scott, S. K., Showalter, K. **Instabilities in propagating reaction-diffusion fronts.** *The Journal of chemical physics*, **98** (8), 6332–6343, 1993.
- [40] Murray, J. D. **Mathematical Biology I. An Introduction**, tomo 17 de *Interdisciplinary Applied Mathematics*. New York: Springer, 2002.
- [41] Crank, J. **The mathematics of diffusion.** Oxford university press, 1979.
- [42] Turing, A. **The chemical basis of morphogenesis.** *Philosophical Transactions of the Royal Society B*, **237**, 37–72, 1952.
- [43] NVIDIA, Vingelmann, P., Fitzek, F. H. **CUDA**, release: 10.2.89, 2020.
- [44] Van Rossum, G., Drake, F. L. **Python 3 Reference Manual.** Scotts Valley, CA: CreateSpace, 2009.
- [45] Okuta, R., Unno, Y., Nishino, D., Hido, S., Loomis, C. **CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations.** En: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS). 2017.
- [46] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., *et al.* **Array programming with NumPy.** *Nature*, **585** (7825), 357–362, sep. 2020.
- [47] Lam, S. K., Pitrou, A., Seibert, S. **Numba: A llvm-based python jit compiler.** En: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, págs. 1–6. 2015.
- [48] Lattner, C., Adve, V. **LLVM: A Compilation Framework for Lifelong Program Analysis and Transformation.** págs. 75–88. San Jose, CA, USA, 2004.
- [49] Kertesz, J., Wolf, D. E. **Noise reduction in Eden models: II. Surface structure and intrinsic width.** *Journal of Physics A: Mathematical and General*, **21** (3), 747, 1988.
- [50] Wolf, D., Kertesz, J. **Surface width exponents for three-and four-dimensional eden growth.** *EPL (Europhysics Letters)*, **4** (6), 651, 1987.

- [51] Krug, J., Spohn, H. Mechanism for rough-to-rough transitions in surface growth. *Physical review letters*, **64** (19), 2332, 1990.

Agradecimientos

Agradezco al Instituto Balseiro, a la Universidad Nacional de Cuyo, a la Comisión Nacional de Energía Atómica y fundamentalmente a todos los argentinos, que con sus impuestos financian a estas instituciones, por brindarme estudios universitarios gratuitamente. Agradezco enteramente a todo el personal de estas instituciones que hacen posible, con gran esfuerzo, que la educación universitaria llegue a todo el país. Particular agradecimiento para mi director, Dr. Alejandro Kolton, por su gran apoyo en el desarrollo de este trabajo y buena predisposición para todo. Agradezco también a mis compañeros de camada Amir Zablotsky, Pedro Llauradó, Esteban Acerbo, Marco Madile, Ezequiel Saidman, Martín Famá y Francisco Gymnich por estar siempre presentes.

