Disorderly Surface Growth

Le Chen

Department of Mathematics and Statistics

Auburn University

Graduate Student Seminar
Auburn University
Nov. 01, 2023

Acknowledgment

DMS-Probability, No. *2246850*
2023 – 2026

Co-PI: Dr. Panqiu Xia
(on market)

Collaboration grant: No. *959981*
2022 – 2027

Plan

Plan

Discrete growth models

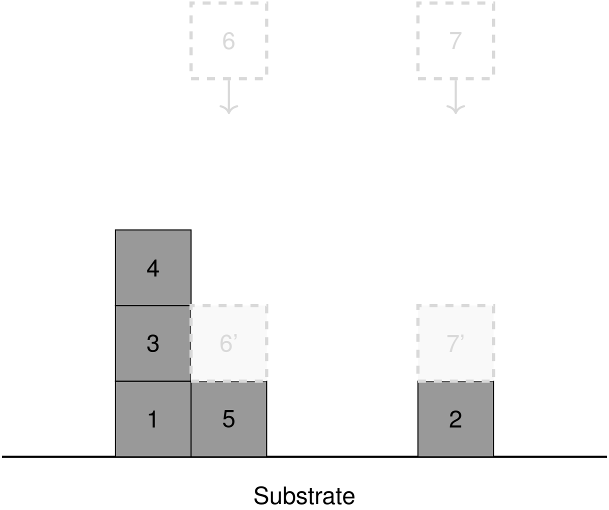
Continuous growth models – KPZ and SHE

More about SHE

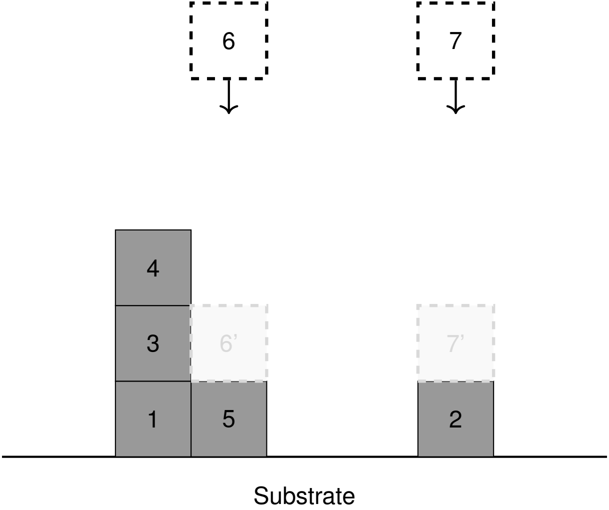
SHE with sublinear growth – Case Study

Final remarks

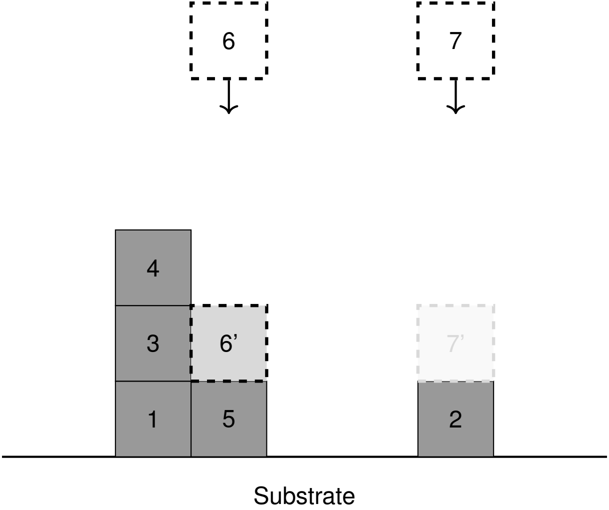
Random deposition



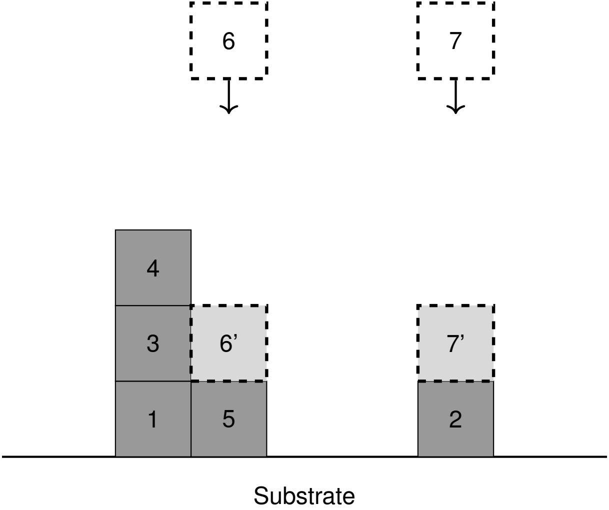
Random deposition



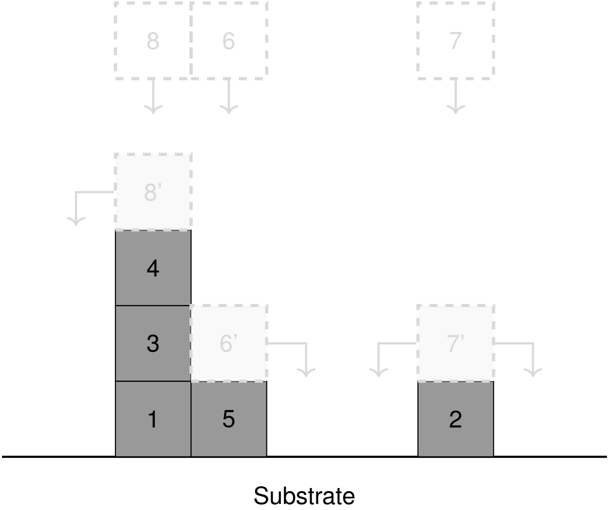
Random deposition



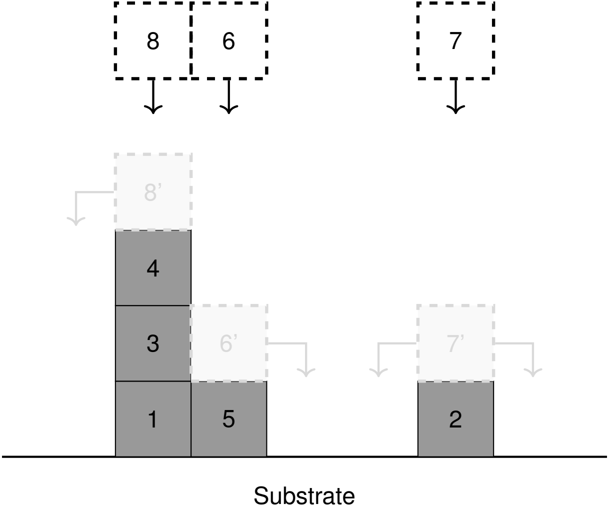
Random deposition



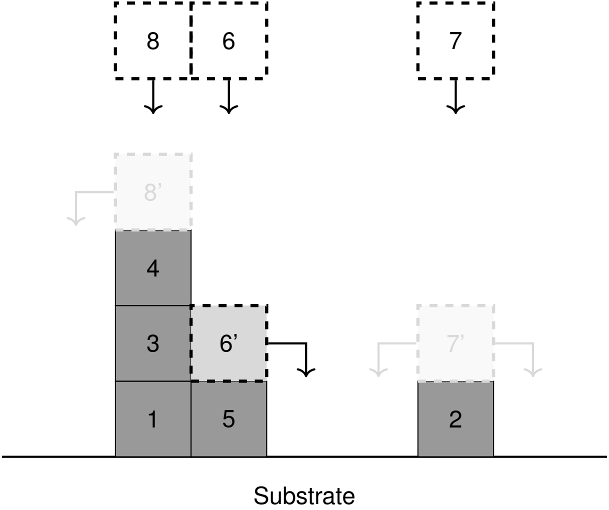
Random deposition with surface relaxation



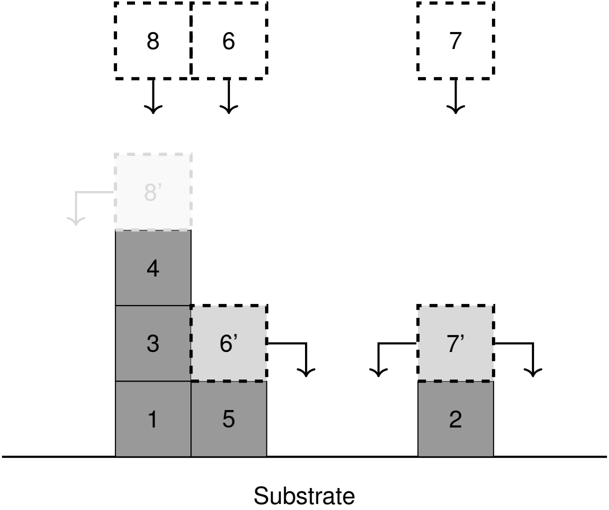
Random deposition with surface relaxation



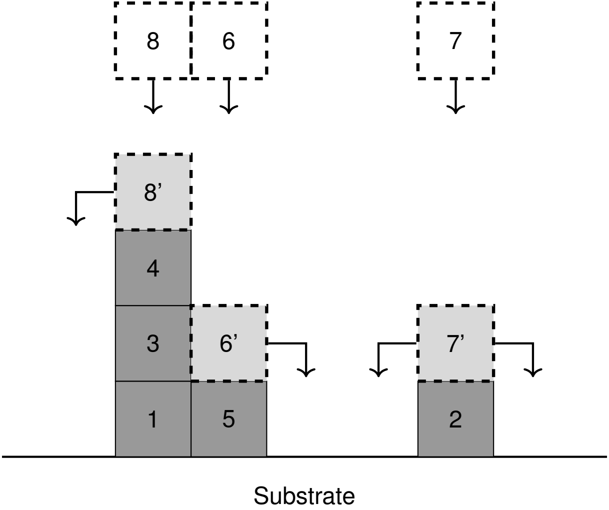
Random deposition with surface relaxation



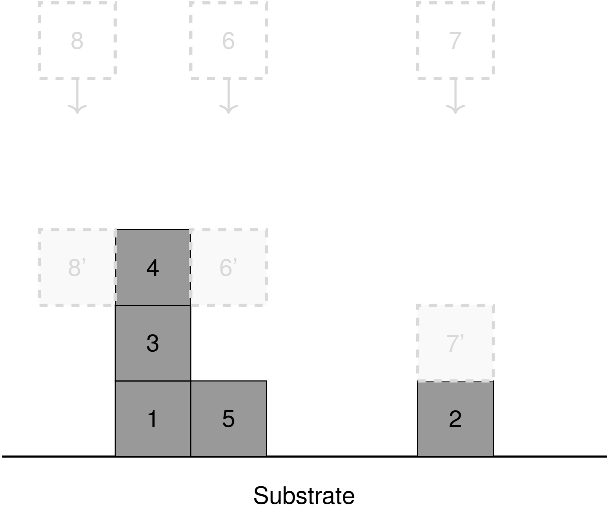
Random deposition with surface relaxation



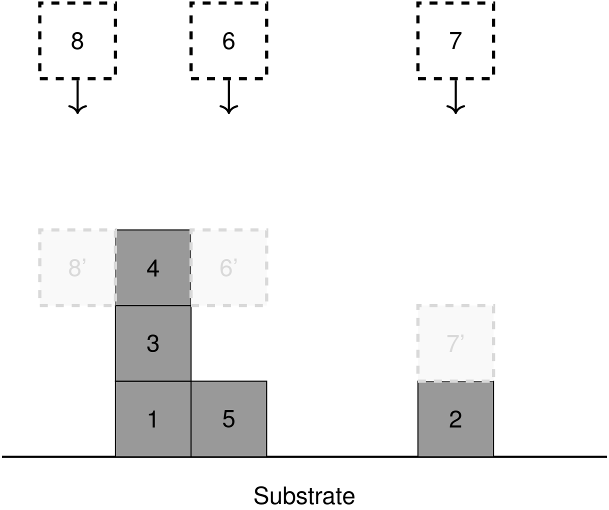
Random deposition with surface relaxation



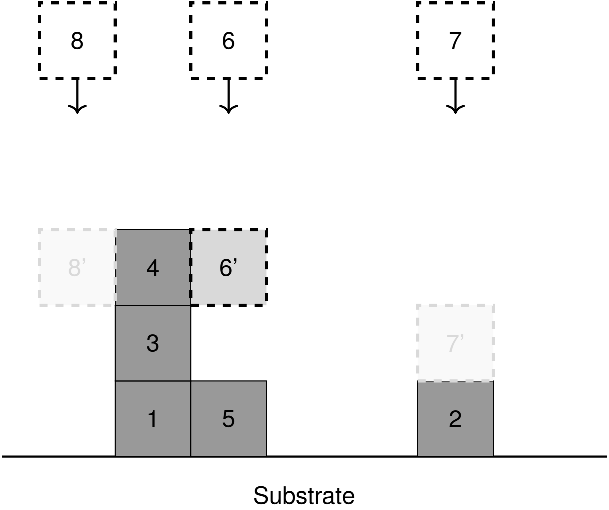
Ballistic deposition



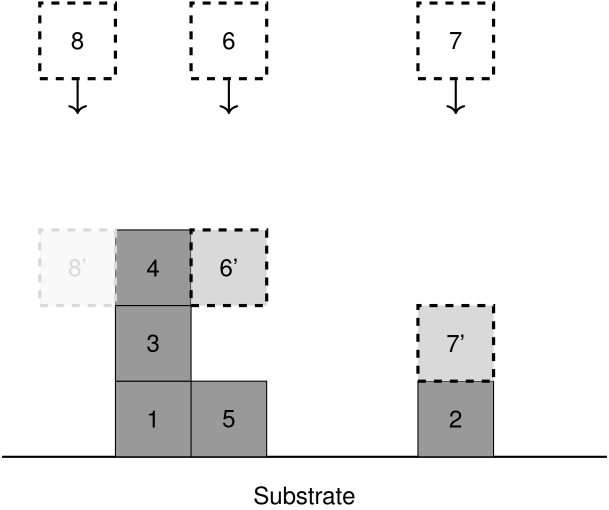
Ballistic deposition



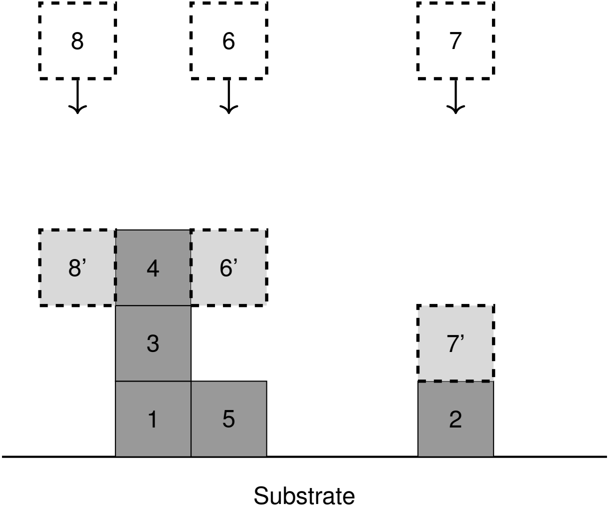
Ballistic deposition



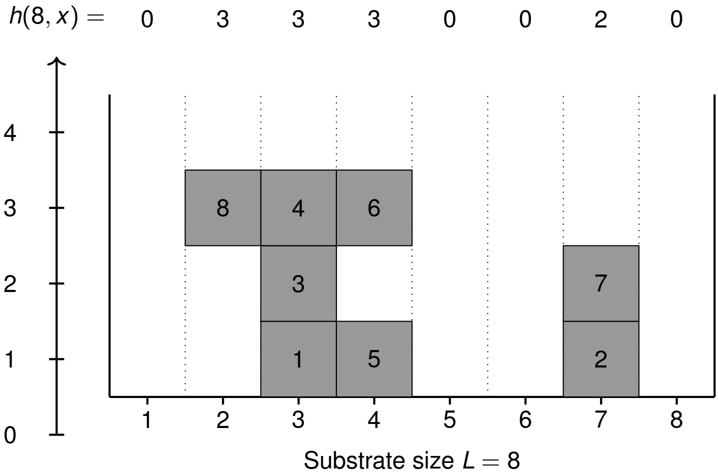
Ballistic deposition



Ballistic deposition



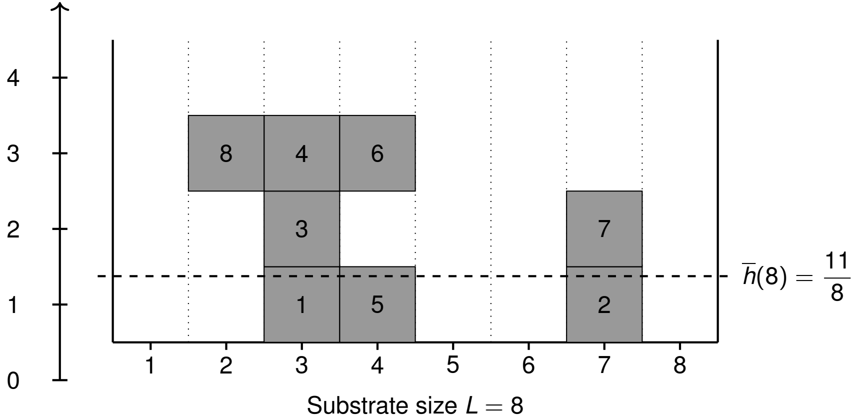
Average height and fluctuation



Average height and fluctuation

$$\overline{h}(t) = \frac{1}{L}\sum_{x=1}^{L} h(t,x) \qquad \text{Fluctuation } W(L,t) = \sqrt{\frac{1}{L}\sum_{x=1}^{L}\left[h(t,x) - \overline{h}(t)\right]^2}$$

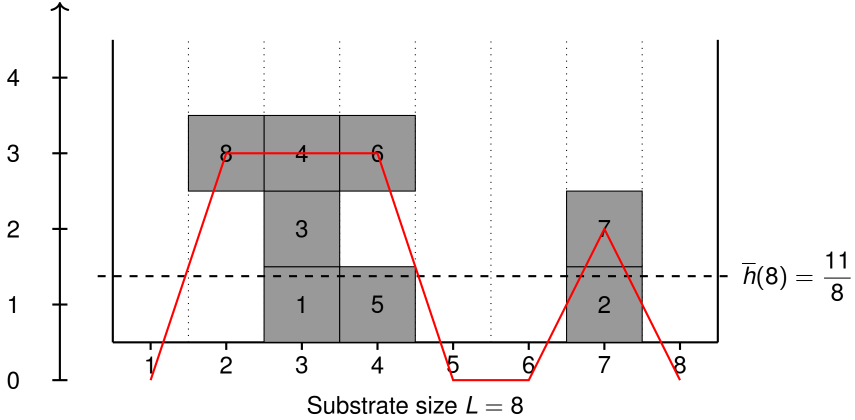
$h(8,x) =$ 0 3 3 3 0 0 2 0



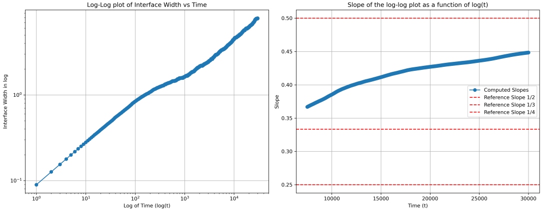
Average height and fluctuation

$$\overline{h}(t) = \frac{1}{L}\sum_{x=1}^{L} h(t,x) \qquad \text{Fluctuation } W(L,t) = \sqrt{\frac{1}{L}\sum_{x=1}^{L}\left[h(t,x) - \overline{h}(t)\right]^2}$$

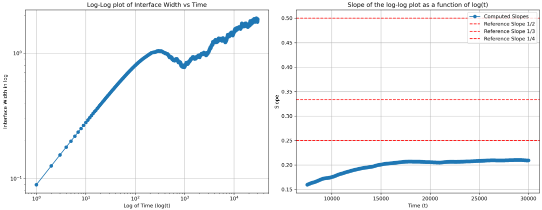
$h(8,x) =$ 0 3 3 3 0 0 2 0



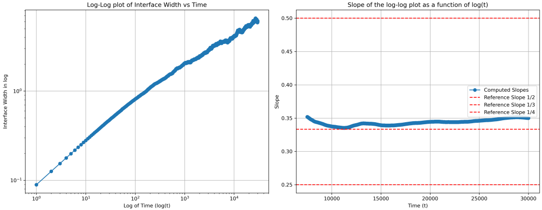
RD

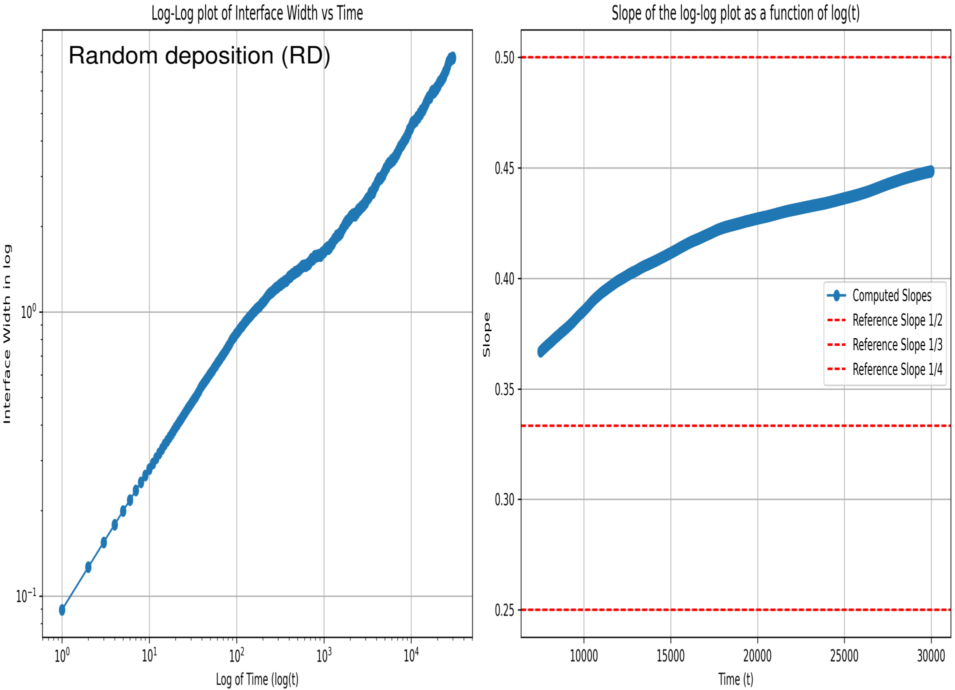


RD Relaxed

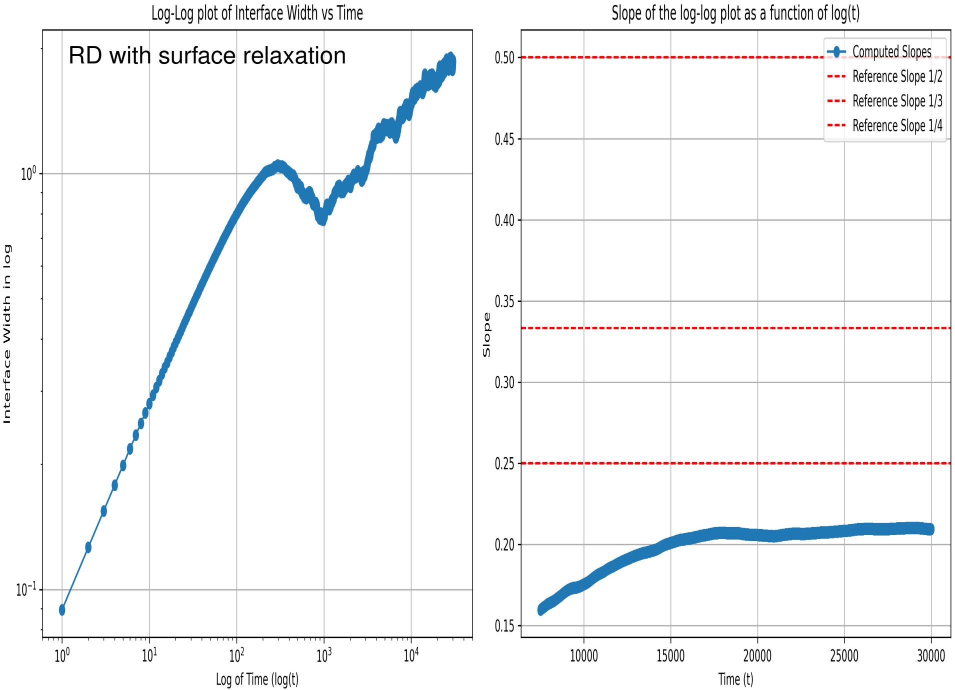


BD

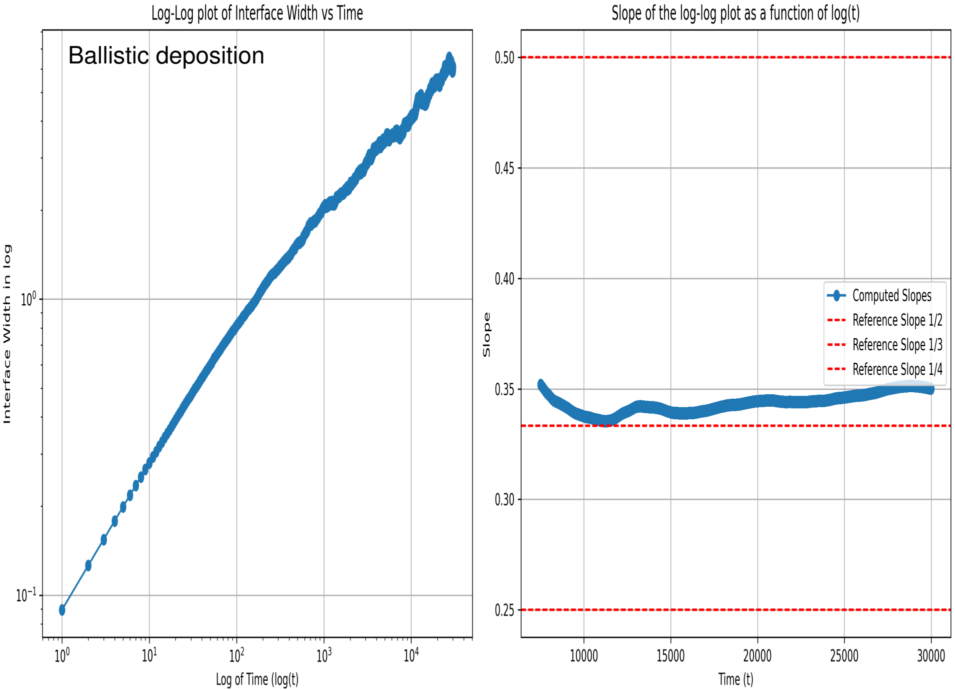




Log-Log plot of Interface Width vs Time
Random deposition (RD)
Interface Width in log
Log of Time (log(t))
Slope of the log-log plot as a function of log(t)
Slope
Time (t)
Computed Slopes
Reference Slope 1/2
Reference Slope 1/3
Reference Slope 1/4



Log-Log plot of Interface Width vs Time
RD with surface relaxation
Interface Width in log
Log of Time (log(t))
10^0
10^{-1}
10^0
10^1
10^2
10^3
10^4
Slope of the log-log plot as a function of log(t)
Computed Slopes
Reference Slope 1/2
Reference Slope 1/3
Reference Slope 1/4
Slope
Time (t)
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
10000
15000
20000
25000
30000


Log-Log plot of Interface Width vs Time
Ballistic deposition
Interface Width in log
Log of Time (log(t))
10^0
10^{-1}
10^0 10^1 10^2 10^3 10^4
Slope of the log-log plot as a function of log(t)
Slope
Time (t)
0.50 0.45 0.40 0.35 0.30 0.25
10000 15000 20000 25000 30000
Computed Slopes
Reference Slope 1/2
Reference Slope 1/3
Reference Slope 1/4

```
❯ ./RD_CLI.py --help
usage: RD_CLI.py [-h] [-w WIDTH] [-e HEIGHT] [-s STEPS] [--relax] [--BD] [-m]

    Simulate Random Deposition on a substrate.
    Outputs: 1. Substrate_WIDTHxHEIGHT_Particles=STEPS_[Relaxed/BD].txt
                A text file for the substrate.
             2. Statistical figures, loglog plot for the interface width and the estimated slope.

    Author: Le Chen (le.chen@auburn.edu, chenle02@gmail.com)
    Date: 2023-10-22



options:
  -h, --help            show this help message and exit
  -w WIDTH, --width WIDTH
                        Width of the substrate (default: 100)
  -e HEIGHT, --height HEIGHT
                        Maximum height of the substrate (default: 60)
  -s STEPS, --steps STEPS
                        Number of particles to drop (default: 5000)
  --relax               Surface Relaxation: go to the nearest lowest neighbor (default: False)
  --BD                  Ballistic decomposition (default: False)
  -m, --movie           Generate the mp4 movie (default: False)
```

```python
    outputfile = f'Substrate_{width}x{height}_Particles={steps}_Relaxed.txt'
    np.savetxt(outputfile, substrate, fmt='%d', delimiter=',')
    print(f"{outputfile} saved!")
    return outputfile


def Ballistic_Deposition(width, height, steps):
    substrate = np.zeros((height, width))
    topmost = height - 1

    for step in range(steps):
        position = np.random.randint(0, width)

        # Determine the landing rows for middle, left, and right columns
        column_data = substrate[:, position]
        landing_positions = np.where(column_data > 0)
        if landing_positions[0].size > 0:
            landing_row_mid = np.min(landing_positions) - 1
        else:
            landing_row_mid = height - 1

        column_data = substrate[:, max(position - 1, 0)]
        landing_positions = np.where(column_data > 0)
        if landing_positions[0].size > 0:
            landing_row_left = np.min(landing_positions)
        else:
            landing_row_left = height - 1

        column_data = substrate[:, min(position + 1, width - 1)]
        landing_positions = np.where(column_data > 0)
        if landing_positions[0].size > 0:
            landing_row_right = np.min(landing_positions)
        else:
            landing_row_right = height - 1

        # Stick to the top neighbor
        landing_row = min(landing_row_mid, landing_row_left, landing_row_right)      ■■■ Argume
        substrate[landing_row, position] = step + 1

        if landing_row < topmost:
            topmost = landing_row

        if (step + 1) % 200 == 0:
            print(f"Step: {step + 1}/{steps}, Level at {height - topmost}/{height}")

        if topmost < height * 0.10 or topmost <= 2:
            print(f"Stopped at step {step + 1}, Level at {height - topmost}/{height}")
            break

    outputfile = f'Substrate_{width}x{height}_Particles={steps}_BD.txt'
    np.savetxt(outputfile, substrate, fmt='%d', delimiter=',')
    print(f"{outputfile} saved!")
    return outputfile


def interface_width(filename):
    # Main function to visualize the simulation
    # Load substrate from file
    substrate = np.loadtxt(filename, delimiter=',')

    # Parameters
    height, width = substrate.shape
    print(f"Height: {height}, Width: {width}")
    steps = int(np.max(substrate))
    interface = np.zeros(steps)

    # Compute the interface width
    for step in range(1, steps + 1):
```

V-LINE master ~1 1 5 1 RD_CLI.py   utf-8 python 30% 82:35

```python
def Random_Deposition(width, height, steps):
    substrate = np.zeros((height, width))
    topmost = height - 1

    for step in range(steps):
        position = np.random.randint(0, width)
        landing_row = np.max(np.where(substrate[:, position] == 0))
        substrate[landing_row, position] = step + 1

        if landing_row < topmost:
            topmost = landing_row

        if (step + 1) % 200 == 0:
            print(f"Step: {step + 1}/{steps}, Level at {height - topmost}/{height}")

        if topmost < height * 0.10 or topmost <= 2:
            print(f"Stopped at step {step + 1}, Level at {height - topmost}/{height}")
            break

    outputfile = f'Substrate_{width}x{height}_Particles={steps}.txt'
    np.savetxt(outputfile, substrate, fmt='%d', delimiter=',')
    print(f"{outputfile} saved!")
    return outputfile
```

V-LINE master ~1 5 1 RD_CLI.py

-- VISUAL LINE --

```python
def Random_Deposition_Surface_Relaxation(width, height, steps):
    substrate = np.zeros((height, width))
    topmost = height - 1

    for step in range(steps):
        position = np.random.randint(0, width)

        # Determine the lnading rows for middle, left, and right columns
        landing_row_mid = np.max(np.where(substrate[:, position] == 0))
        landing_row_left = np.max(np.where(substrate[:, max(position - 1, 0)] == 0))
        landing_row_right = np.max(np.where(substrate[:, min(position + 1, width - 1)] == 0))

        # Surface relaxation
        if landing_row_right > max(landing_row_mid, landing_row_left) and position < width - 1:
            # Landing on the right column
            substrate[landing_row_right, position + 1] = step + 1
            landing_row = landing_row_right
        elif landing_row_left > landing_row_mid and position > 1:
            # Landing on the left column
            substrate[landing_row_left, position - 1] = step + 1
            landing_row = landing_row_left
        else:
            # Default, Landing in the middle column
            substrate[landing_row_mid, position] = step + 1
            landing_row = landing_row_mid

        if landing_row < topmost:
            topmost = landing_row

        if (step + 1) % 200 == 0:
            print(f"Step: {step + 1}/{steps}, Level at {height - topmost}/{height}")

        if topmost < height * 0.10 or topmost <= 2:
            print(f"Stopped at step {step + 1}, Level at {height - topmost}/{height}")
            break

    outputfile = f'Substrate_{width}x{height}_Particles={steps}_Relaxed.txt'
    np.savetxt(outputfile, substrate, fmt='%d', delimiter=',')
    print(f"{outputfile} saved!")
    return outputfile
```

V-LINE master ~1 5 1 RD_CLI.py [+]

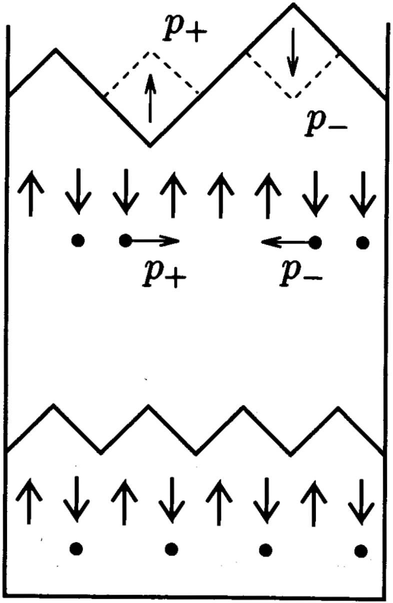
```
0, 0, 0, 0,  0, 0,  0,  0,  0,  0
0, 0, 0, 0,  0, 0,  0,  0,  0,  0
0, 0, 0, 0,  0, 0,  21, 0,  0,  0
0, 0, 0, 0,  0, 0,  20, 19, 16, 17
0, 0, 0, 0,  0, 0,  0,  0,  15, 0
0, 0, 0, 0,  0, 0,  0,  0,  13, 0
0, 0, 0, 18, 0, 0,  0,  0,  11, 9
0, 0, 0, 14, 0, 12, 0,  0,  0,  6
0, 0, 7, 8,  0, 3,  0,  0,  0,  5
0, 0, 2, 0,  0, 1,  0,  10, 0,  4
```

Simulations on

Random deposition vs. Ballistic decomposition

More models? Even more simpler?

time t



Solid-on-Solid

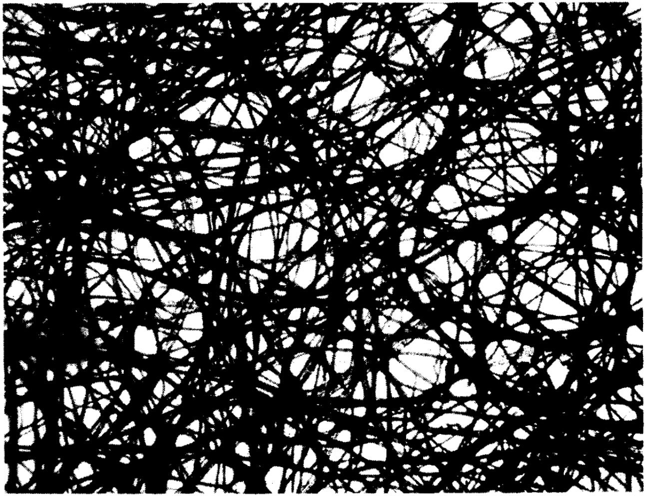
Ising

Lattice Gas

Initial state

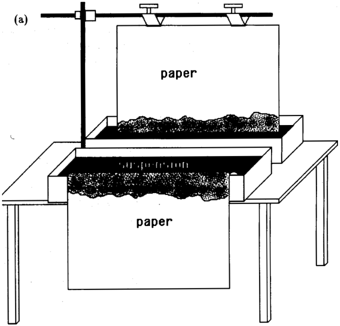
https://www.youtube.com/watch?v=nh2P_I3SGoU

Paper – a random environment



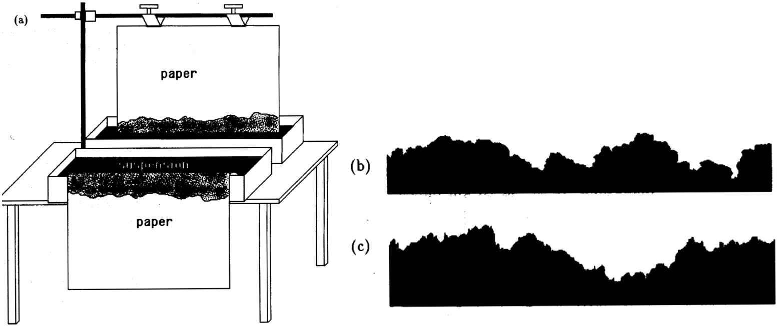
Zhang, J., Zhang, Y.-C., Alstrøm, P., Levinsen, M., *Phys. A: Stat. Mech. Appl.*, 1992

Paper wetting experiment



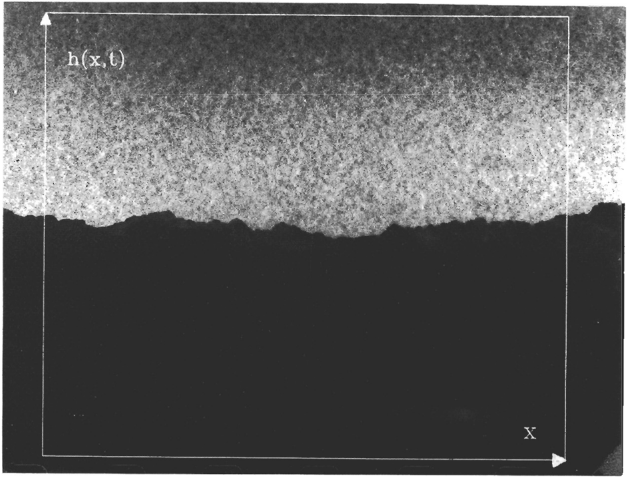
Barabási, A.-L., Stanley, H. E., 1995

Paper wetting experiment



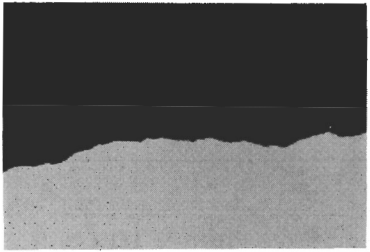
Barabási, A.-L., Stanley, H. E., 1995

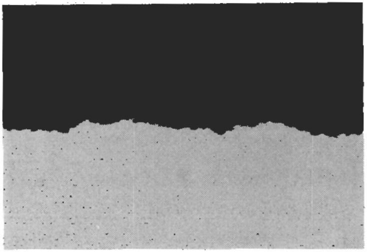
Paper burning experiment



Zhang, J., Zhang, Y.-C., Alstrøm, P., Levinsen, M., *Phys. A: Stat. Mech. Appl.*, 1992

Paper rupture experiment



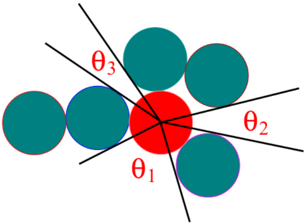


Kertész, J., Horváth, V. k., Weber, F., *Fractals*, 1993

Rule of replication of **cells**

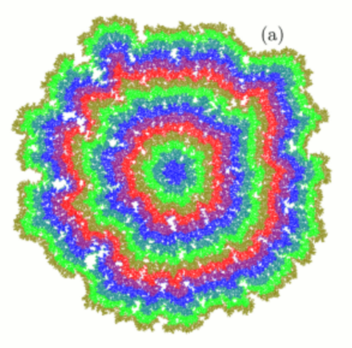
Replication probability $\propto$ Aperture angle θ_i

Microscopic world



$\Rightarrow$

Macroscopic world



Santalla, S. N., Ferreira, S. C., *Phys. Rev. E*, 2018

Study of growing interfaces in a thin film

— Convection of nematic liquid crystal*

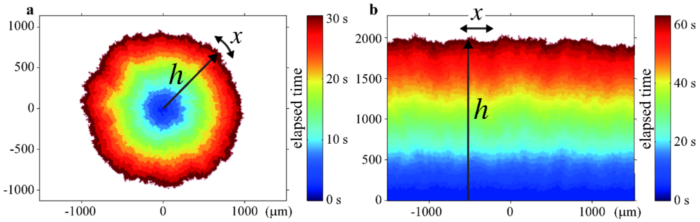
Show movies !

---

*Takeuchi, K. A., Sano, M., Sasamoto, T., Spohn, H., *Sci. Rep.*, 2011

Study of growing interfaces in a thin film

— Convection of nematic liquid crystal*



Prediction from KPZ equation:

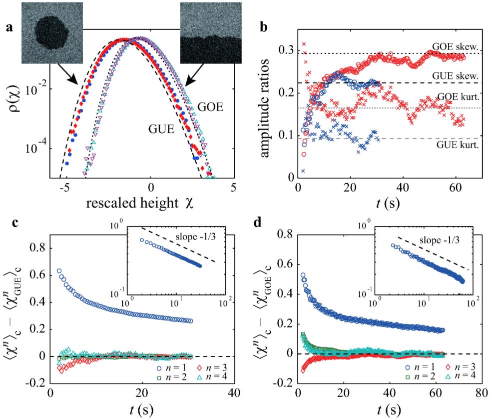
$$h \asymp v_\infty t + (\Gamma t)^{1/3}\, \xi$$

Takeuchi, K. A., Sano, M., Sasamoto, T., Spohn, H., *Sci. Rep.*, 2011

Study of growing interfaces in a thin film

— Convection of nematic liquid crystal*

$$h \asymp v_\infty t + (\Gamma t)^{1/3}\,\xi$$



Takeuchi, K. A., Sano, M., Sasamoto, T., Spohn, H., *Sci. Rep.*, 2011

Plan

KPZ Equation ’86

$$\frac{\partial}{\partial t} h(t,x) = \frac{1}{2}\Delta h(t,x) + \frac{\lambda}{2}(\nabla h)^2 + \dot{W}(t,x) \qquad \text{(KPZ)}$$

Mehran Kardar (1957 –)

Giorgio Parisi (1948 –)

Yicheng Zhang

Kardar, M., Parisi, G., Zhang, Y.-C., *Phys. Rev. Lett.*, 1986

Two ways to obtain KPZ

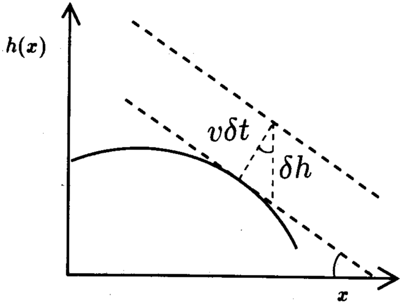
$$\frac{\partial}{\partial t} h(t,x) = \frac{1}{2}\Delta h(t,x) + \frac{\lambda}{2}\left(\nabla h\right)^2 + \dot{W}(t,x) \qquad \text{(KPZ)}$$

1. Via symmetry arguments; as Barabási, A.-L., Stanley, H. E., 1995 and Kardar, M., Parisi, G., Zhang, Y.-C., *Phys. Rev. Lett.*, 1986

Two ways to obtain KPZ

$$\frac{\partial}{\partial t}h(t,x) = \frac{1}{2}\Delta h(t,x) + \frac{\lambda}{2}(\nabla h)^2 + \dot{W}(t,x) \qquad \text{(KPZ)}$$

1. Via symmetry arguments; as Barabási, A.-L., Stanley, H. E., 1995 and Kardar, M., Parisi, G., Zhang, Y.-C., *Phys. Rev. Lett.*, 1986

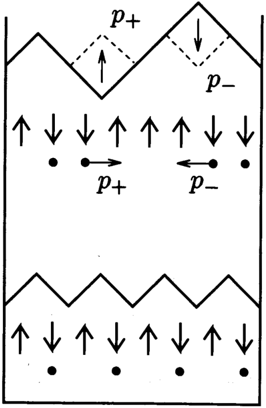


Two ways to obtain KPZ

2. Hydrodynamic limit of SOS; as Bertini, L., Giacomin, G., *Probab. Theory Related Fields*, 1999

Two ways to obtain KPZ

2. Hydrodynamic limit of SOS; as Bertini, L., Giacomin, G., *Probab. Theory Related Fields*, 1999



KPZ & SHE: Two Sides, One Mathematical Coin

$$h(t,x) = \log u(t,x)$$

$$\frac{\partial}{\partial t} h(t,x) = \frac{1}{2}\Delta h(t,x) + \frac{\lambda}{2}\left(\nabla h\right)^2 + \dot{W}(t,x) \tag{KPZ}$$

$$\frac{\partial}{\partial t} u(t,x) = \frac{1}{2}\Delta u(t,x) + u(t,x)\dot{W}(t,x) \tag{SHE}$$

Giorgio Parisi
Facts

Ill. Niklas Elmehed © Nobel Prize Outreach

Giorgio Parisi
The Nobel Prize in Physics 2021

Born: 4 August 1948, Rome, Italy

Affiliation at the time of the award: Sapienza University of Rome, Rome, Italy

Prize motivation: "for the discovery of the interplay of disorder and fluctuations in physical systems from atomic to planetary scales."

Prize share: 1/2

https://www.nobelprize.org/prizes/physics/2021/parisi/facts/

Sir Martin Hairer
KBE FRS

Martin Hairer at the Royal Society admissions day in London, July 2014

Born	14 November 1975 (age 45) Geneva, Switzerland
Citizenship	Austrian British
Education	College Claparede, Geneva
Alma mater	University of Geneva
Spouse(s)	Xue-Mei Li (m. 2003)[1][2]
Awards	Whitehead Prize (2008) Philip Leverhulme Prize (2008) Wolfson Research Merit Award (2009) Fermat Prize (2013) Fröhlich Prize (2014) Fields Medal (2014) Breakthrough Prize in Mathematics (2021)
	Scientific career
Fields	Probability theory[3] Analysis[3]

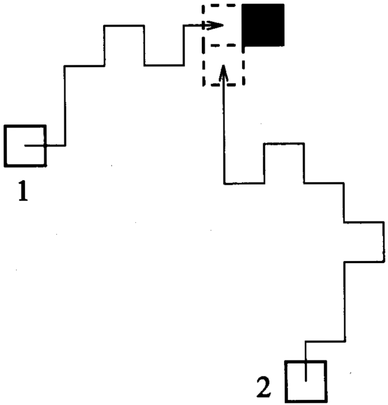
Fields Medal in 2014

- Solving the KPZ equation.
- A theory of regularity structures.

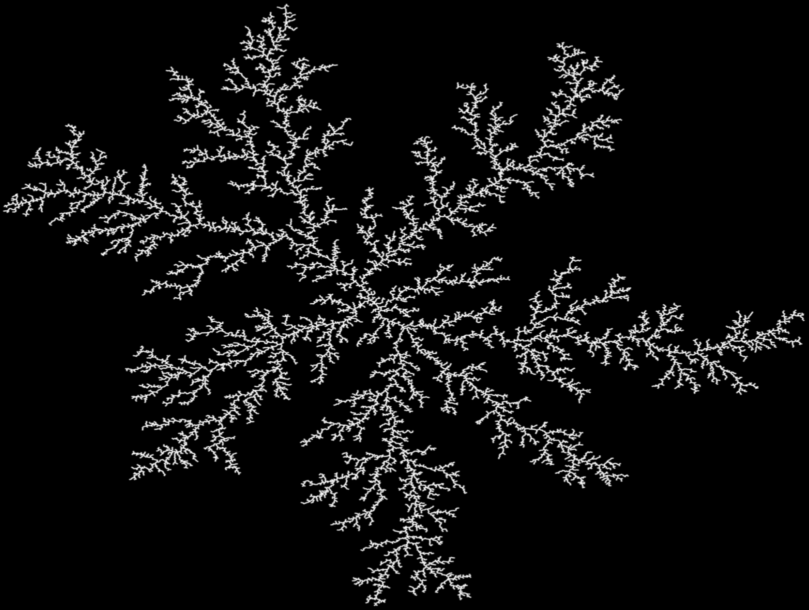
Hairer, M., *Ann. of Math. (2)*, 2013

Hairer, M., *Invent. Math.*, 2014

Nonlocal model – Diffusion Limited Aggregation (DLA)



Barabási, A.-L., Stanley, H. E., 1995



Self avoid random walk

Random walk in random environment

Percolation

Random polymers

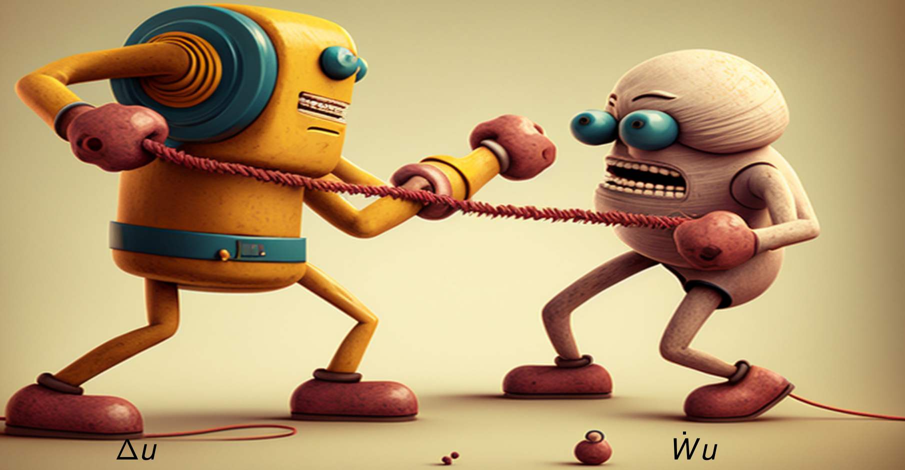
Phase transitions

Plan

$$\frac{\partial^b}{\partial t^b} u(t,x) = \left(\frac{1}{2}\Delta + \dot{W}(t,x)\right) u(t,x)$$

Δu	$\dot{W} u$
Smoothing	Roughening

$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\Delta + \dot{W}(t,x)\right)u(t,x)$$



Δu $\dot{W}u$

Smoothing Roughening

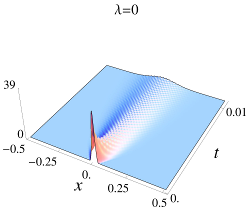
$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

— Propagation of tall peaks*

Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

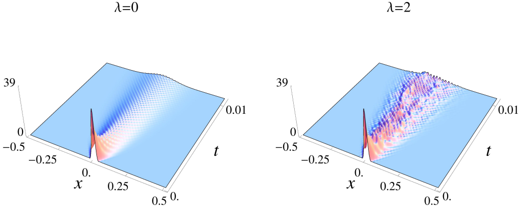
— Propagation of tall peaks*



Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

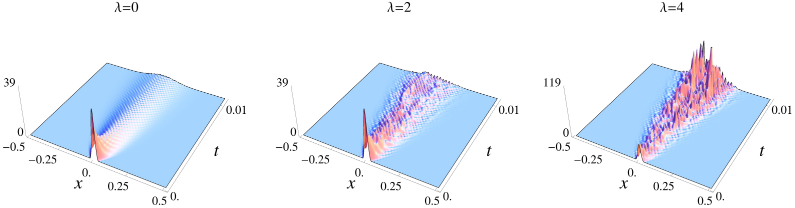
— Propagation of tall peaks*



Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t} u(t,x) = \left(\frac{1}{2} \frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x) \right) u(t,x)$$

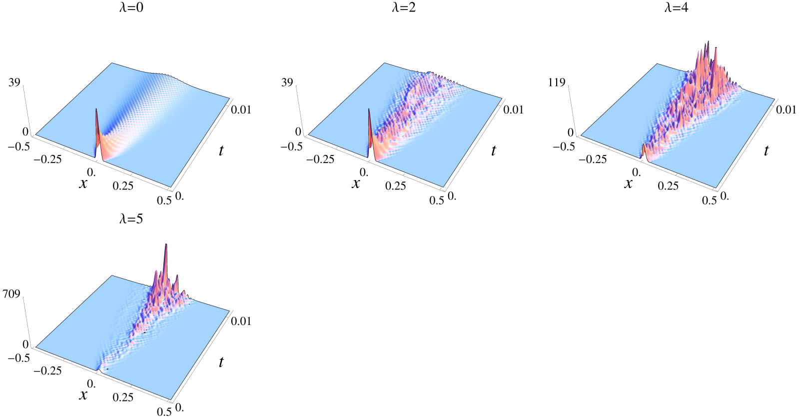
— Propagation of tall peaks*



* Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t} u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

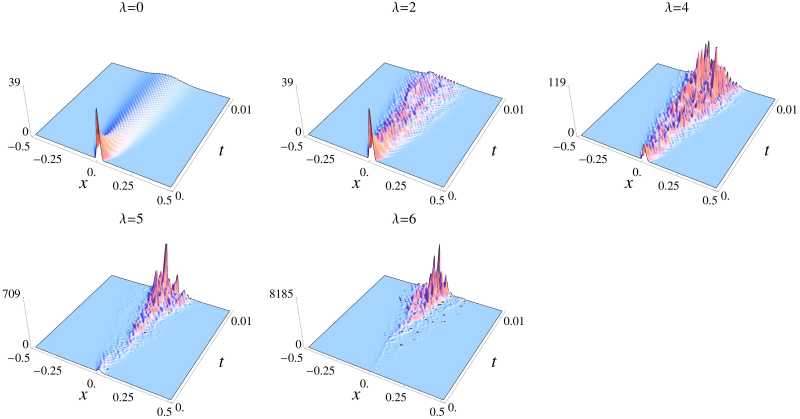
— Propagation of tall peaks*



Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

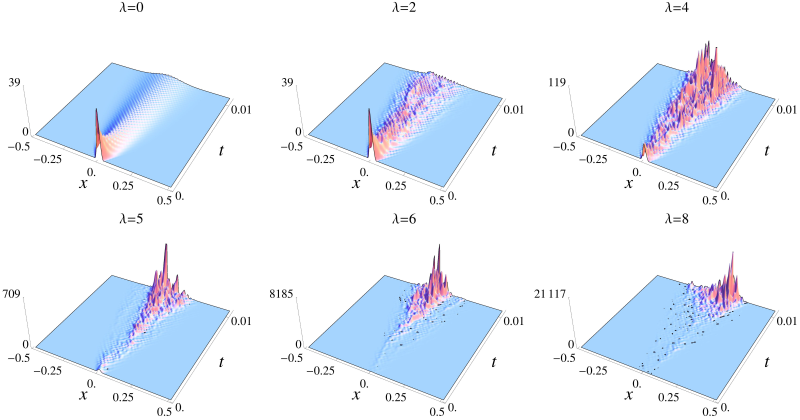
— Propagation of tall peaks*



*Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right)u(t,x)$$

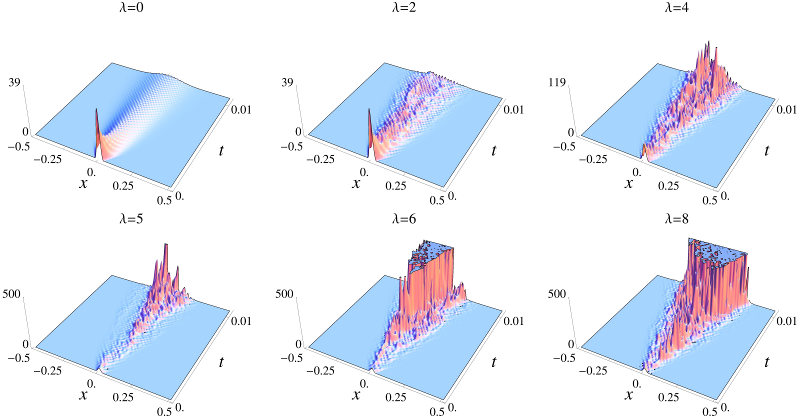
— Propagation of tall peaks*



Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

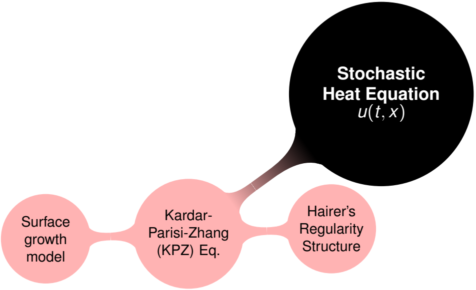
$$\frac{\partial}{\partial t}u(t,x) = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} + \lambda \dot{W}(t,x)\right) u(t,x)$$

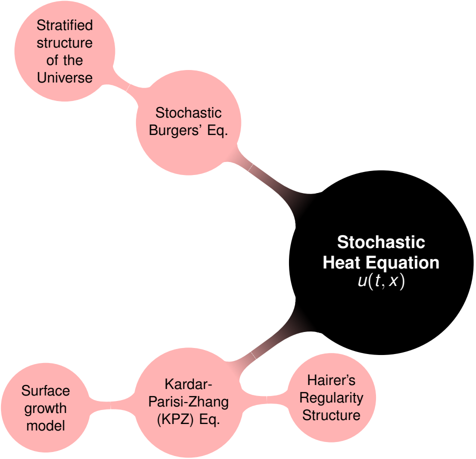
— Propagation of tall peaks*

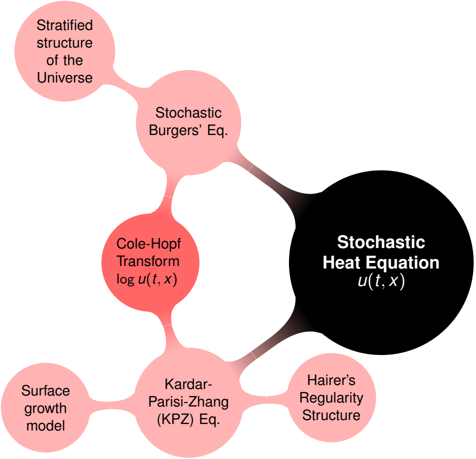


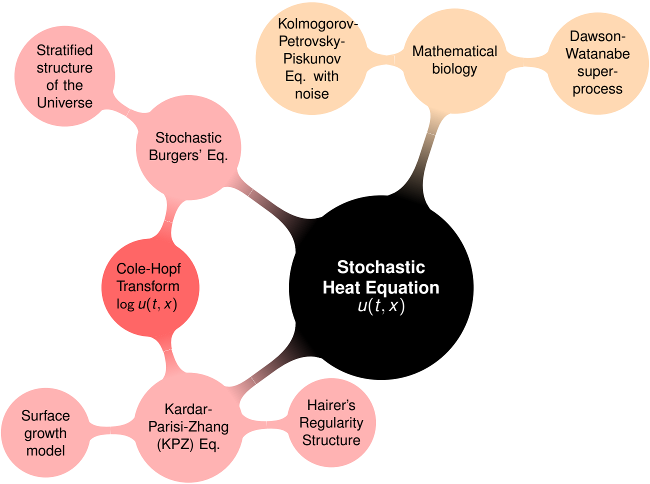
Chen, L., Dalang, R. C., *Ann. Probab.*, 2015

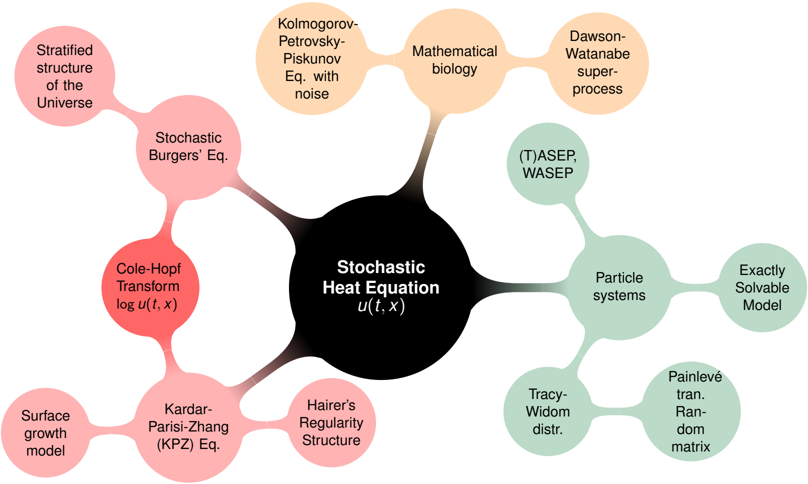
Stochastic Heat Equation
$u(t, x)$

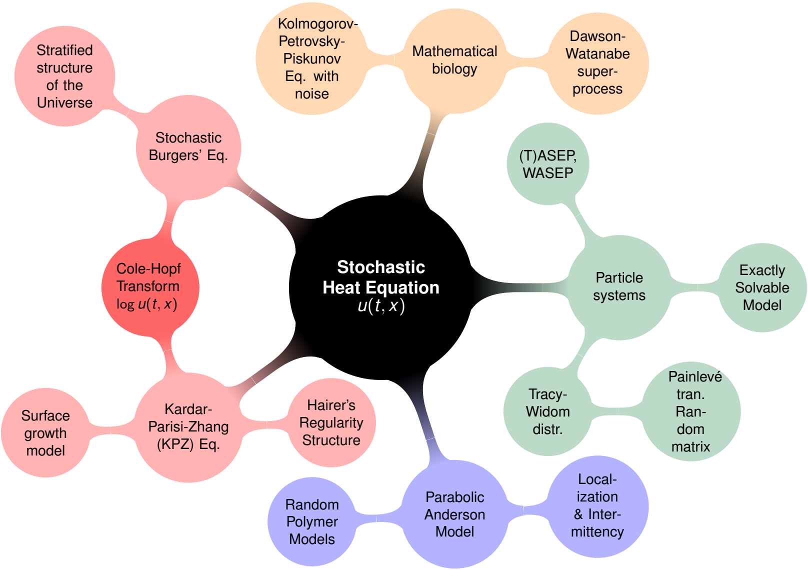

Stochastic
Heat Equation
$u(t, x)$
Surface
growth
model
Kardar-
Parisi-Zhang
(KPZ) Eq.
Hairer's
Regularity
Structure


Stratified structure of the Universe
Stochastic Burgers' Eq.
Stochastic Heat Equation $u(t, x)$
Surface growth model
Kardar-Parisi-Zhang (KPZ) Eq.
Hairer's Regularity Structure


Stratified structure of the Universe
Stochastic Burgers' Eq.
Cole-Hopf Transform $\log u(t,x)$
Stochastic Heat Equation $u(t,x)$
Surface growth model
Kardar-Parisi-Zhang (KPZ) Eq.
Hairer's Regularity Structure


Stochastic Heat Equation $u(t, x)$
Mathematical biology
Kolmogorov-Petrovsky-Piskunov Eq. with noise
Dawson-Watanabe super-process
Stochastic Burgers' Eq.
Stratified structure of the Universe
Cole-Hopf Transform $\log u(t, x)$
Kardar-Parisi-Zhang (KPZ) Eq.
Surface growth model
Hairer's Regularity Structure


Stochastic Heat Equation $u(t,x)$
Stochastic Burgers' Eq.
Stratified structure of the Universe
Cole-Hopf Transform $\log u(t,x)$
Kardar-Parisi-Zhang (KPZ) Eq.
Surface growth model
Hairer's Regularity Structure
Mathematical biology
Kolmogorov-Petrovsky-Piskunov Eq. with noise
Dawson-Watanabe super-process
Particle systems
(T)ASEP, WASEP
Exactly Solvable Model
Tracy-Widom distr.
Painlevé tran. Random matrix


Stochastic Heat Equation $u(t, x)$
Stochastic Burgers' Eq.
Stratified structure of the Universe
Cole-Hopf Transform $\log u(t, x)$
Kardar-Parisi-Zhang (KPZ) Eq.
Surface growth model
Hairer's Regularity Structure
Mathematical biology
Kolmogorov-Petrovsky-Piskunov Eq. with noise
Dawson-Watanabe super-process
Particle systems
(T)ASEP, WASEP
Exactly Solvable Model
Tracy-Widom distr.
Painlevé tran. Random matrix
Parabolic Anderson Model
Random Polymer Models
Localization & Intermittency

Stochastic heat equation (SHE)

$$\begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\Delta\right) u(t,x) = \rho(u(t,x))\dot{W}(t,x), \quad t>0,\ x\in\mathbb{R}^d \\[2em] u(0,\cdot) = \mu \end{cases} \tag{SHE}$$

$$u(t,x) = J_0(t,x) + \int_0^t \int_{\mathbb{R}^d} p_{t-s}(x-y)\rho(u(s,y))W(\mathrm{d}s,\mathrm{d}y)$$

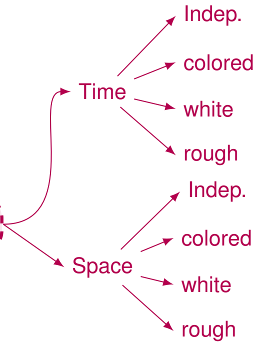
---

* $p_t(x) := (2\pi t)^{-d/2} \exp\left(-|x|^2/(2t)\right)$.

† $J_0(t,x) := \int_{\mathbb{R}^d} p_t(x-y)\mu(\mathrm{d}y)$ — solution to the homogeneous equation.

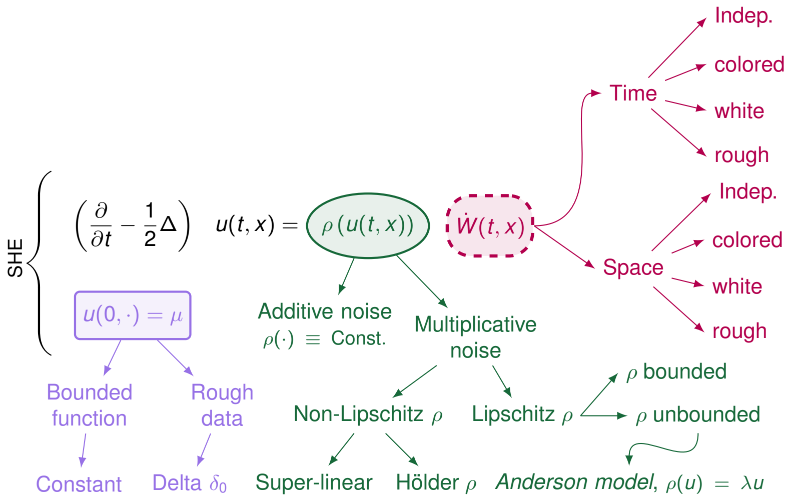
$$\text{SHE}\begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\Delta\right) \quad u(t,x) = \quad \rho\left(u(t,x)\right) \qquad \dot{W}(t,x) \\[2em] u(0,\cdot) = \mu \end{cases}$$

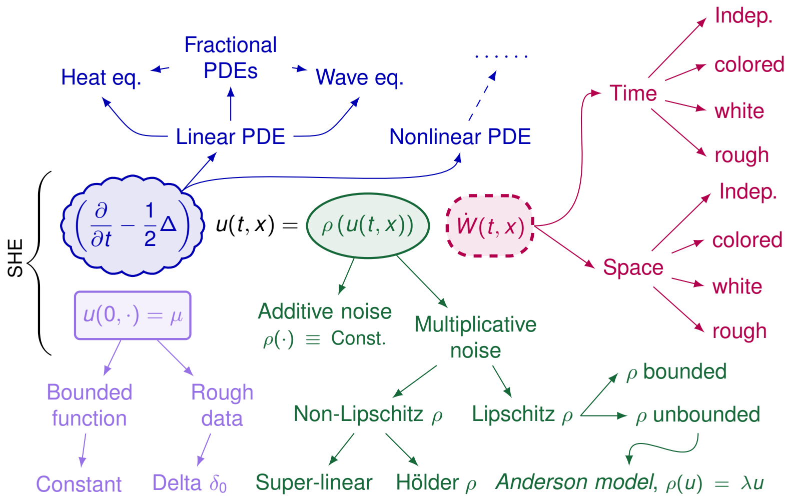
$$\text{SHE}\begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\Delta\right) u(t,x) = \rho(u(t,x)) \boxed{\dot{W}(t,x)} \\[2em] u(0,\cdot) = \mu \end{cases}$$



$$\text{SHE}\begin{cases} \left(\dfrac{\partial}{\partial t}-\dfrac{1}{2}\Delta\right) u(t,x) = \rho\,(u(t,x))\ \dot{W}(t,x) \\ u(0,\cdot) = \mu \end{cases}$$

- $\dot{W}(t,x)$
  - Time
    - Indep.
    - colored
    - white
    - rough
  - Space
    - Indep.
    - colored
    - white
    - rough
- $\rho\,(u(t,x))$
  - Additive noise $\rho(\cdot) \equiv$ Const.
  - Multiplicative noise
    - Non-Lipschitz ρ
      - Super-linear
      - Hölder ρ
    - Lipschitz ρ
      - ρ bounded
      - ρ unbounded
        - *Anderson model*, $\rho(u) = \lambda u$


SHE
$\left(\frac{\partial}{\partial t} - \frac{1}{2}\Delta\right) u(t,x) = \rho\,(u(t,x))\,\dot{W}(t,x)$
$u(0,\cdot) = \mu$
Time
Indep.
colored
white
rough
Space
Indep.
colored
white
rough
Additive noise
$\rho(\cdot) \equiv$ Const.
Multiplicative noise
Non-Lipschitz ρ
Lipschitz ρ
ρ bounded
ρ unbounded
Super-linear
Hölder ρ
Anderson model, $\rho(u) = \lambda u$
Bounded function
Rough data
Constant
Delta δ_0


Fractional PDEs
Heat eq.
Wave eq.
. . . . . .
Linear PDE
Nonlinear PDE
SHE
$\left(\frac{\partial}{\partial t} - \frac{1}{2}\Delta\right) u(t,x) = \rho\left(u(t,x)\right) \dot{W}(t,x)$
$u(0,\cdot) = \mu$
Time
Indep.
colored
white
rough
Space
Indep.
colored
white
rough
Additive noise
$\rho(\cdot) \equiv$ Const.
Multiplicative noise
Bounded function
Rough data
Constant
Delta δ_0
Non-Lipschitz ρ
Lipschitz ρ
ρ bounded
ρ unbounded
Super-linear
Hölder ρ
Anderson model, $\rho(u) = \lambda u$

Moment Lyapunov exponents:

$$p \mapsto \lim_{t\to\infty} \frac{1}{t} \log \mathbb{E}\left[|u(t,x)|^p\right]$$

The growth of the above mapping	$u(t,x)$
the faster	the more chaotic the more **interemittent** the farther from equilibrium

Plan

$$
\text{(PAM)} \quad \begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\dfrac{\partial^2}{\partial x^2}\right) u(t,x) = \lambda u(t,x)\dot{W}(t,x), & t > 0, x \in \mathbb{R}, \\ u(0,\cdot) = \mu. \end{cases}
$$

For SHE on $\mathbb{R}$ with space-time white noise, it is known that

$$
\lim_{p\to\infty} p^{-3} \log \mathbb{E}\left[u(t,x)^p\right] = \frac{\lambda^4}{24}, \qquad \text{for all } t > 0 \text{ and } x \in \mathbb{R} \text{ and,}
$$

$$
\lim_{t\to\infty} t^{-1} \log \mathbb{E}\left[u(t,x)^p\right] = \frac{1}{24} p(p^2-1)\lambda^4, \quad \text{for all } p \geq 2 \text{ and } x \in \mathbb{R}.
$$

Chen, X., *Ann. Inst. Henri Poincaré Probab. Stat.*, 2015
Bertini, L., Cancrini, N., *J. Statist. Phys.*, 1995

$$
\text{(Additive SHE)} \quad \begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\dfrac{\partial^2}{\partial x^2}\right) u(t,x) = \lambda \times 1 \times \dot{W}(t,x), \\ u(0,\cdot) = \mu. \end{cases}
$$

$$
\text{(Bounded } \rho\text{)} \quad \begin{cases} \left(\dfrac{\partial}{\partial t} - \dfrac{1}{2}\dfrac{\partial^2}{\partial x^2}\right) u(t,x) = \lambda \times \dfrac{u(t,x)}{1+u(t,x)} \times \dot{W}(t,x), \\ u(0,\cdot) = \mu. \end{cases}
$$

Bounded ρ case is called "*weak diffusion*". The phenomenon of smaller maxima of the solution due to the smaller growth of ρ is called "*smoothing intermittency*".

$$
\mathbb{E}\left[u(t,x)^p\right] \asymp t^{p/2}.
$$

Zel′dovich, Y. B., Ruzmauikin, A. A., Sokoloff, D. D., 1990

Proc. Natl. Acad. Sci. USA
Vol. 84, pp. 6323–6325, September 1987
Applied Mathematical and Physical Sciences

Self-excitation of a nonlinear scalar field in a random medium

YA. B. ZELDOVICH†, S. A. MOLCHANOV‡, A. A. RUZMAIKIN§, AND D. D. SOKOLOFF‡

†Institute of Physical Problems, Academy of Sciences, Moscow, U.S.S.R.; ‡Moscow State University, Moscow, U.S.S.R.; and §Institute of Terrestrial Magnetizm, Ionosphere and Radiowave Propagation, Academy of Sciences, Moscow Region, U.S.S.R.

Contributed by Ya. B. Zeldovich, April 29, 1987

ABSTRACT We discuss the evolution in time of a scalar field under the influence of a random potential and diffusion. The cases of a short-correlation in time and of stationary potentials are considered. In a linear approximation and for sufficiently weak diffusion, the statistical moments of the field grow exponentially in time at growth rates that progressively increase with the order of the moment; this indicates the intermittent nature of the field. Nonlinearity halts this growth and in some cases can destroy the intermittency. However, in many nonlinear situations the intermittency is preserved: high, persistent peaks of the field exist against the background of a smooth field distribution. These widely spaced peaks may make a major contribution to the average characteristics of the field.

suppression of the growth of high maxima in the solution, thus smoothing the intermittency. However, the result proves to be less trivial and depends radically on both the time behavior of the potential and the form of nonlinearity.

For the sake of simplicity, put æ = 0; i.e., neglect the spatial diffusion of the field. We shall indicate the influence of a weak diffusion on the results separately. We consider two types of potentials. The first is characterized by a short correlation time

$$U(t, \mathbf{x}, \omega) = u + \tau^{-1/2}\frac{d\mathrm{w}_t}{dt}, \qquad [2]$$

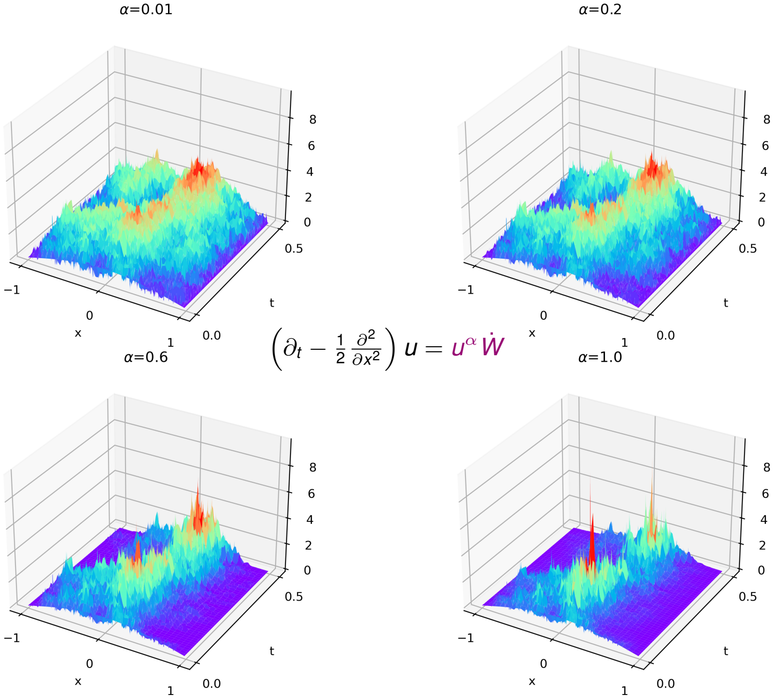
where u is the stationary value of the potential, τ is a certain characteristic time similar to the characteristic turbulent dif-

Zel′dovich, Y. B., Ruzmauikin, A. A., Sokoloff, D. D., 1990

Simulations of

$$\left(\partial_t - \frac{1}{2}\frac{\partial^2}{\partial x^2}\right) u = u^{\alpha}\,\dot{W}$$

with $\alpha = 0.01, 0.2, 0.6$ and $\dot{W}$ is space-time white noise.



$$\left(\partial_t - \frac{1}{2}\frac{\partial^2}{\partial x^2}\right) u = u^{\alpha}\,\dot{W}$$

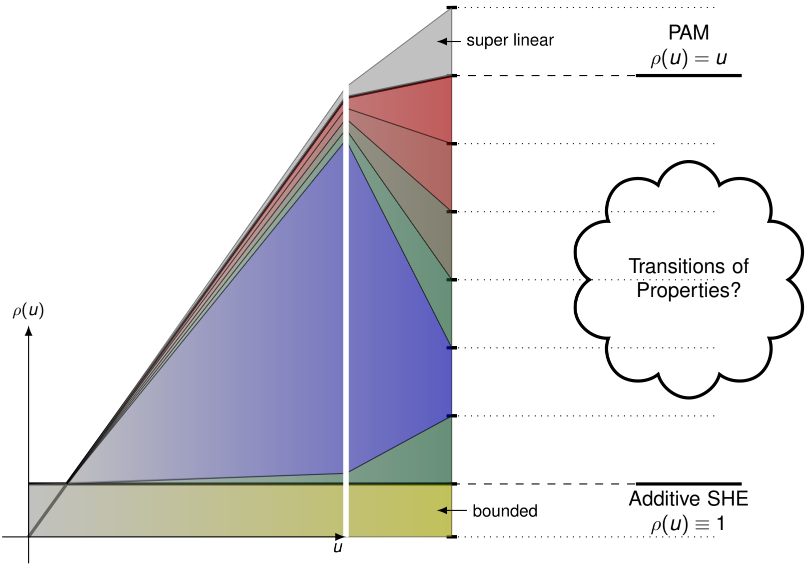
When $d = 1$, $f = \delta_0$, & $\mu \equiv 1$, Conus, Joseph, Khoshnevisan, and Shiu, 2013 proved that with probability 1,

$$\begin{cases} \displaystyle\sup_{|x| \le R} u(t,x) \asymp [\log R]^{1/2} & \text{SHE with bounded } \rho; \\ \displaystyle\log \sup_{|x| \le R} u(t,x) \asymp [\log R]^{2/3} & \text{PAM.} \end{cases}$$

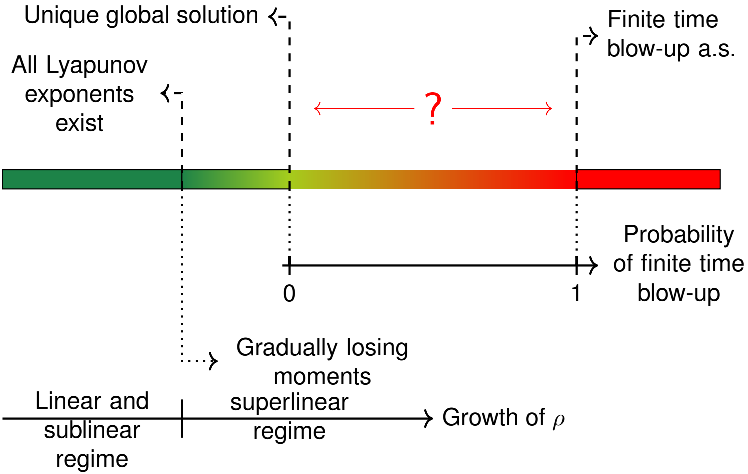
Question: What if ρ has sublinear growth at infinity?

Conus, D., Joseph, M., Khoshnevisan, D., Shiu, S.-Y., *Probab. Theory Related Fields*, 2013

PAM vs Additive SHE

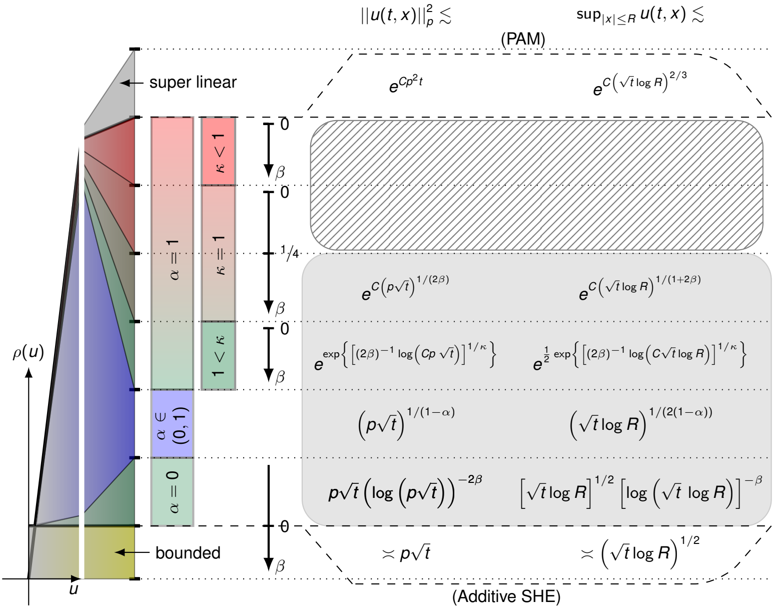


SHE on superlinear growth regime



Chen, L., Foondun, M., Huang, J., Salins, M., *preprint arXiv:2310.02153*, 2023

Chen, L., Huang, J., *Proc. Amer. Math. Soc.*, 2023



Chen, L., Xia, P., *preprint arXiv:2306.06761*, 2023

Main References*:

Chen, L., & Eisenberg, N. (2022). Invariant measures for the nonlinear stochastic heat equation with no drift term. *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*.

Chen, L., & Eisenberg, N. (2023). Interpolating the stochastic heat and wave equations with time-independent noise: Solvability and exact asymptotics. *Stoch. Partial Differ. Equ. Anal. Comput.*, *11*(3), 1203–1253.

Chen, L., & Xia, P. (2023). Asymptotic properties of stochastic partial differential equations in the sublinear regime. *preprint arXiv:2306.06761*.

Conus, D., Joseph, M., Khoshnevisan, D., & Shiu, S.-Y. (2013). On the chaotic character of the stochastic heat equation, II. *Probab. Theory Related Fields*, *156*(3-4), 483–533.

Zel′dovich, Y. B., Ruzmauikin, A. A., & Sokoloff, D. D. (1990). *The almighty chance* (Vol. 20) [Translated from the Russian by Anvar Shukurov]. World Scientific Publishing Co., Inc., River Edge, NJ.

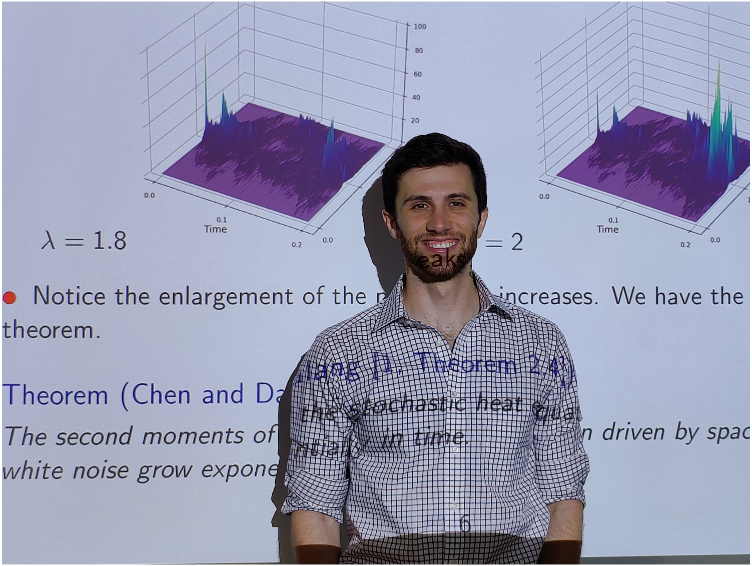
* References are produced from *SPDEs-Bib*: https://github.com/chenle02/SPDEs-Bib

* Download the bib file: https://github.com/chenle02/SPDEs-Bib/blob/main/All.bib

Plan

Doing a Ph.D. in this exciting field?

One good example by Dr. *Nicholas Eisenberg*



One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)

One good example by Dr. *Nicholas Eisenberg*

One good example by Dr. *Nicholas Eisenberg*

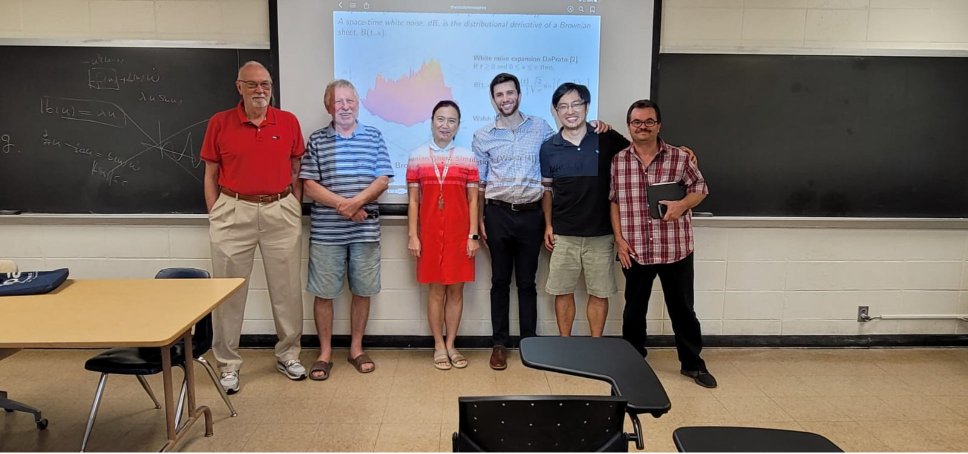
1. 2018 fall starts with me at UNLV
2. 2019 – 2020, via Zoom when I was at Emory (visited me once at Atlanta)
3. 2021 – 2022, Auburn

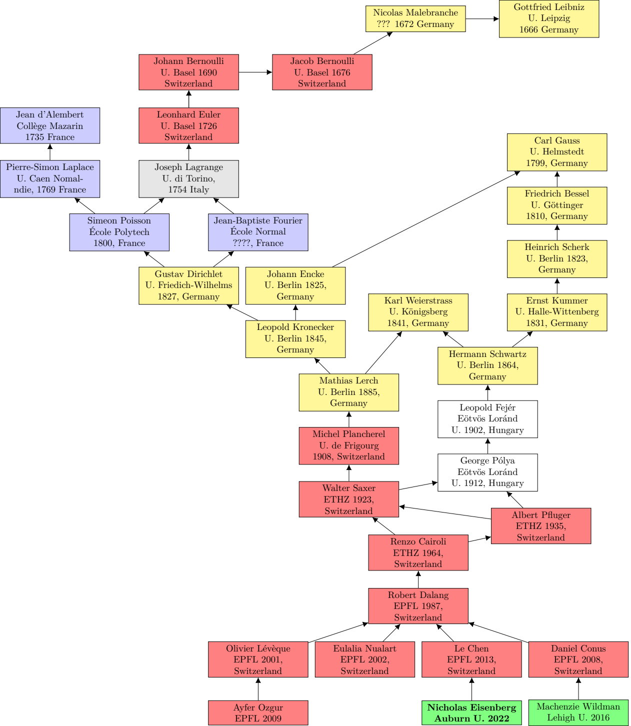
Two publications:

Chen, L., Eisenberg, N., *Stoch. Partial Differ. Equ. Anal. Comput.*, 2023

Chen, L., Eisenberg, N., *J. Theoret. Probab. (pending revision, preprint arXiv:2209.04771)*, 2022

Now 2022 – now: Postodc at Department of Energy (Data Scientist)


A space-time white noise, dB, is the distributional derivative of a Brownian sheet, B(t, x).
White noise expansion DaPrato [2]



Join our family !

Doing a Ph.D. in this exciting field?

1. Prelim courses:

   Math 7000/7010: Applied Mathematics

   Math 7200/7210: Real Analysis

   Math 7800/7810: Probability

   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:

   Math 7000/7010: Applied Mathematics

   Math 7200/7210: Real Analysis

   Math 7800/7810: Probability

   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:

   Math 7000/7010: Applied Mathematics

   Math 7200/7210: Real Analysis

   Math 7800/7810: Probability

   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:
   Math 7000/7010: Applied Mathematics
   Math 7200/7210: Real Analysis
   Math 7800/7810: Probability
   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:
   Math 7000/7010: Applied Mathematics
   Math 7200/7210: Real Analysis
   Math 7800/7810: Probability
   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:
   Math 7000/7010: Applied Mathematics
   Math 7200/7210: Real Analysis
   Math 7800/7810: Probability
   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:
   Math 7000/7010: Applied Mathematics
   Math 7200/7210: Real Analysis
   Math 7800/7810: Probability
   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

Doing a Ph.D. in this exciting field?

1. Prelim courses:
   Math 7000/7010: Applied Mathematics
   Math 7200/7210: Real Analysis
   Math 7800/7810: Probability
   Math 7830/7840: Stochastic Processes

2. Learn programming, Linux, Python, GitHub, Git, Vim, etc.

3. Motivations !

4. Sufficient NSF funding supports for GRA

A Haiku for the talk

by OpenAI's GPT-4.0

(Accessed on 2023/11/01)

Surface patterns form,
Fluctuations tell the tale,
KPZ unveils.

Acknowledgment:

* ChatGPT (Code Interpreter)
* Midjourney
* Vim, RIP *Bram Moolenaar* (1961 – 2023-08-03)