# An iterative algorithm for the Dirichlet problem with random conductances

Chenlin GU

DMA/ENS

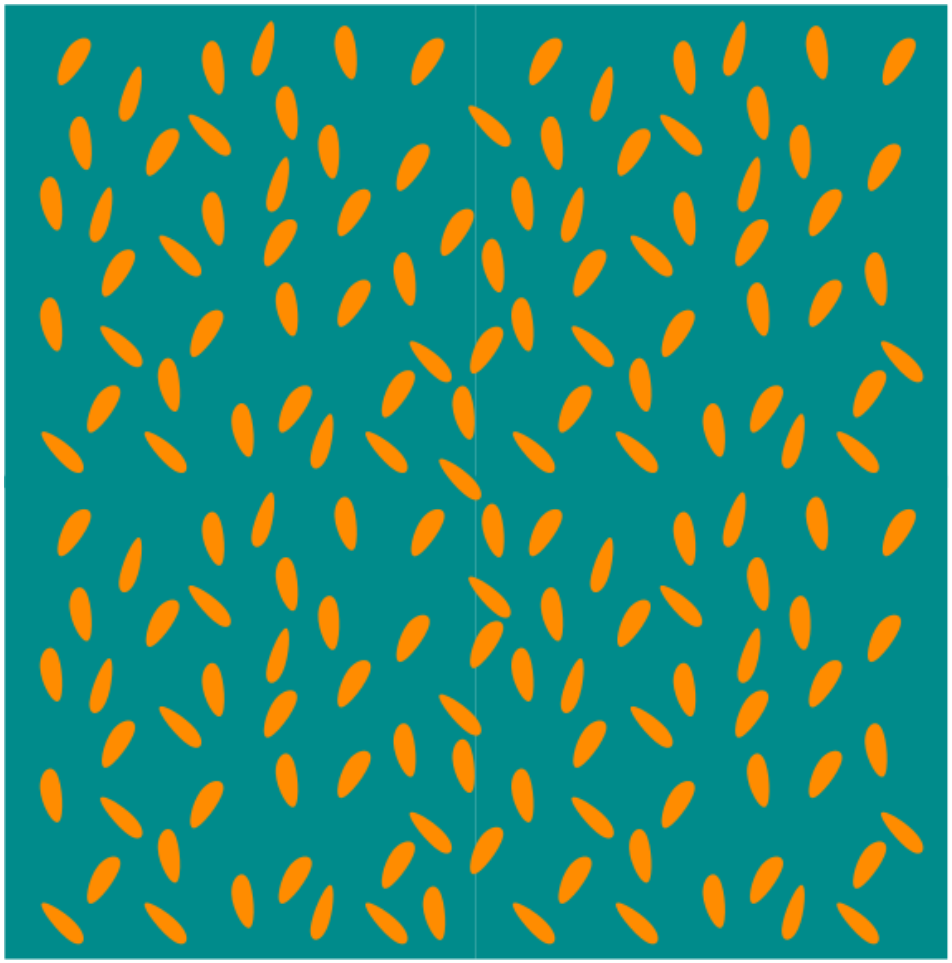
Journées de Probabilités 2019

June 25, 2019

# Outline

# Outline

# Random stationary medium



# Random conductance model

- $(\mathbb{Z}^d, E_d)$ standard d-dimension lattice.
- $\mathbf{a} : E_d \to [\Lambda^{-1}, \Lambda]$ with $\Lambda > 1$. $\{\mathbf{a}(e)\}_{e \in E_d}$ i.i.d. called *random conductances*.
- The *discrete divergence operator* $-\nabla \cdot \mathbf{a}\nabla$ is defined by

$$-\nabla \cdot \mathbf{a}\nabla u(x) := \sum_{y \sim x} \mathbf{a}(x,y)(u(x) - u(y)).$$

- $\square_m := \left(-\frac{3^m}{2}, \frac{3^m}{2}\right)^d \cap \mathbb{Z}^d$.
- Object : Find an algorithm to solve the elliptic Dirichlet problem quickly for big $m$,

$$\begin{cases} -\nabla \cdot \mathbf{a}\nabla u = f & \text{in } \operatorname{int}(\square_m), \\ u = g & \text{on } \partial\square_m. \end{cases} \tag{1.1}$$

# Naive method takes time when $m$ big

- Why could it be a problem ?
- Jacobi iterative method $\approx$ The stationary solution of random walk.
- Let $P(x,y) := \frac{\mathbf{a}(x,y)}{\sum_{z\sim x}\mathbf{a}(x,z)}$ be the semigroup generated by $\mathbf{a}$ and $\tilde{f}(x) = f(x)/(\sum_{z\sim x}\mathbf{a}(x,z))$.
- $u$ is the unique solution of the equation $u = Pu + \tilde{f}$.
- We do iteration $u_0 = g$, $u_{n+1} = J(u_n, \tilde{f})$ that

$$u_{n+1} = Pu + \tilde{f}.$$

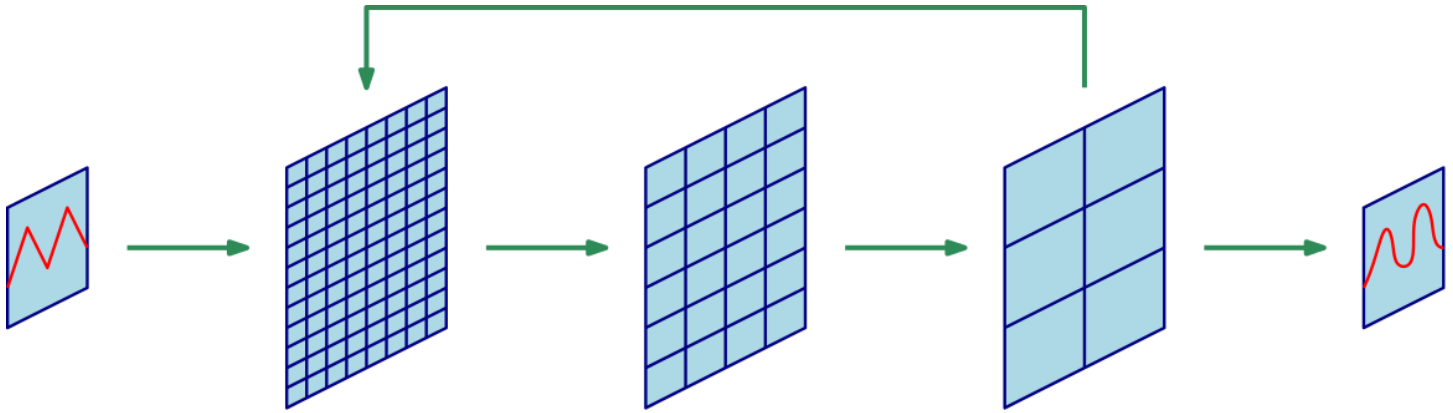
- $\lim_{n\to\infty} u_n = u$.
- When $\square_m$ is big, it takes time. At least $O(r^2)$ iterations for $r = 3^m$.

# Outline

# Multigrid method

Efficient method for $\mathbf{a} \equiv const.$ i.e. for the problem $-\Delta u = f$.

1. Try to solve $-\Delta u = f$, we do the Jacobi iteration and $u_1 = J^M(u_0, f)$.
2. $f_1 = f - (-\Delta u_1)$, coarsen the grid by 2, and $u_2 = J^{M/2}(0, f_1)$.
3. $f_2 = f_1 - (-\Delta u_2)$, coarsen the grid by 2, and $u_3 = J^{M/4}(0, f_2)$.
4. $\hat{u} = u_1 + u_2 + u_3$. Iterate this procedure and only $O(\log(r))$ iterations required.



Probabilistic interpretation : coarsened grid $\approx$ random walk with big step size.

# Homogenized solution

- Using the homogenized solution to approximate the real solution : Early work of Kozlov, Papanicolaou, Varadhan, Yurinski etc.
- When $m \to \infty$, we have an effective solution $\bar{u}$ for

$$\begin{cases} -\nabla \cdot \bar{\mathbf{a}} \nabla \bar{u} = f & \text{in } \operatorname{int}(\square_m), \\ u = g & \text{on } \partial \square_m, \end{cases} \tag{1.2}$$

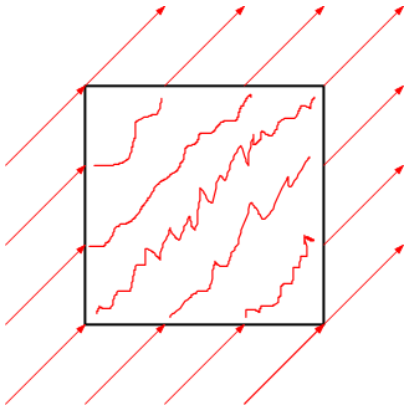
  with $\bar{\mathbf{a}}$ *effective conductance*.
- $\frac{1}{3^m} \|u - \bar{u}\|_{L^2(\square_m)} \xrightarrow{m \to \infty} 0$.
- $\bar{\mathbf{a}} \neq \mathbb{E}[\mathbf{a}]$.

# Homogenized solution

Several disadvantages :

- Loss of microscale information : Locally, $\bar{u}$ and $u$ is never similar.
- Limit of precision : $\frac{1}{3^m} \|u - \bar{u}\|_{L^2(\square_m)} \simeq 3^{-\frac{m}{2}}$. For $m$ fixed, using $\bar{u}$ can never go pass this precision.

Precise object : An algorithm quick but also with more precision.



# Outline

# Iterative algorithm

- The iterative algorithm proposed by Armstrong, Hannukainen, Kuusi, Mourrat.
- Start from an initial guess $u_0 := g$. One iteration is to solve the following equations with the null Dirichlet boundary condition:

$$\begin{cases} (\lambda^2 - \nabla \cdot \mathbf{a}\nabla)u_1 &= f + \nabla \cdot \mathbf{a}\nabla u_0 & \text{in } \mathrm{int}(\square_m), \\ -\nabla \cdot \bar{\mathbf{a}}\nabla \bar{u} &= \lambda^2 u_1 & \text{in } \mathrm{int}(\square_m), \\ (\lambda^2 - \nabla \cdot \mathbf{a}\nabla)u_2 &= (\lambda^2 - \nabla \cdot \bar{\mathbf{a}}\nabla)\bar{u} & \text{in } \mathrm{int}(\square_m), \end{cases} \tag{2.1}$$

  then let $\hat{u} := u_0 + u_1 + u_2$ and we put $\hat{u}$ in the place of $u_0$ to restart the iteration.
- All the three equations are easy to solve. The second one can be handled by the multigrid method, while the first and third one take less time ($O(\frac{1}{\lambda^2})$ iterations) thanks to the regularization.

# Main theorem

Let $\mathcal{Z} = \sup_{u_0, f, g} \frac{\|\nabla(\hat{u}-u)\|_{L^2(\square_m)}}{\|\nabla(u_0-u)\|_{L^2(\square_m)}}$, $\ell(\lambda) = 1$ for $d \geqslant 3$ and $\ell(\lambda) = \log^{\frac{1}{2}}(\lambda)$, and $r = 3^m$.

**Theorem ((Armstrong, Hannukainen, Kuusi, Mourrat 18)(Gu 19+))**

*For any $s \in (0,2)$, there exists a constant $C(m, \Lambda, s, d)$ such that for any $y > 0$*

$$\mathbb{P}[\mathcal{Z} \geqslant y] \leqslant \exp\left(-\left(\frac{y}{C\lambda^{\frac{1}{2}}\ell(\lambda)\log^{\frac{1}{s}}(r)}\right)^s\right).$$

- It suggests a practical choice of $\lambda$ that $\frac{1}{r} \ll \lambda \ll \frac{1}{\log(r)}$.
- Complexity $:= O(\log(r))$ iterations, very close to the one of multigrid.

# Outline

# Two-scale expansion

- Combing the first and second step of the iteration and we obtain $-\nabla \cdot \bar{\mathbf{a}} \nabla \bar{u} = -\nabla \cdot \mathbf{a} \nabla (u - u_0 - u_1)$.
- The third equation gives $(\lambda^2 - \nabla \cdot \mathbf{a} \nabla) u_2 = (\lambda^2 - \nabla \cdot \bar{\mathbf{a}} \nabla) \bar{u}$.
- Two-scale expansion $w := \bar{u} + \sum_{k=1}^{d} (\mathcal{D}_{e_k} \bar{u}) \phi_{e_k}$ with $\{\phi_{e_k}\}_{1 \leqslant k \leqslant d}$ *the first order corrector*.
- We have

$$|\hat{u} - u| = |u - (u_0 + u_1 + u_2)| \leqslant |(u - u_0 - u_1) - w| + |w - u_2|,$$

  so it suffices to know how close the two-scale expansion can be.
- First rigorous result in periodic homogenization : Allaire.
- Quantitative analysis in stochastic homogenization setting : Armstrong, Kussi, Mourrat, Gloria, Neukamm and Otto etc.

# Numerical experience

- $d = 2$, size $= 128 \times 128$, $\mathbf{a} \in \{\frac{1}{\sqrt{2}}, \sqrt{2}\}$ with law Bernoulli($\frac{1}{2}$).
- $f = 1$ and $g = 0$.
- $\lambda = 0.1$.
- The first 22 rounds of iteration give a convergence of errors $\varepsilon_n := \|f - (-\nabla \cdot \mathbf{a} \nabla u_n)\|_{\underline{L}^2(\square_m)}$.

$$\begin{aligned}\{\varepsilon_n\}_{1 \leqslant n \leqslant 22} = \{&34.43, 18.56, 9.99, 5.38, 2.89, 1.56, 0.84,\\ &0.45, 0.24, 0.13, 0.0709, 0.0382, 0.0206,\\ &0.0111, 0.0059, 0.0032, 0.0017, 0.0009,\\ &0.0005064, 0.0002730, 0.0001472, 7.94 \times 10^{-5}\}\end{aligned}$$

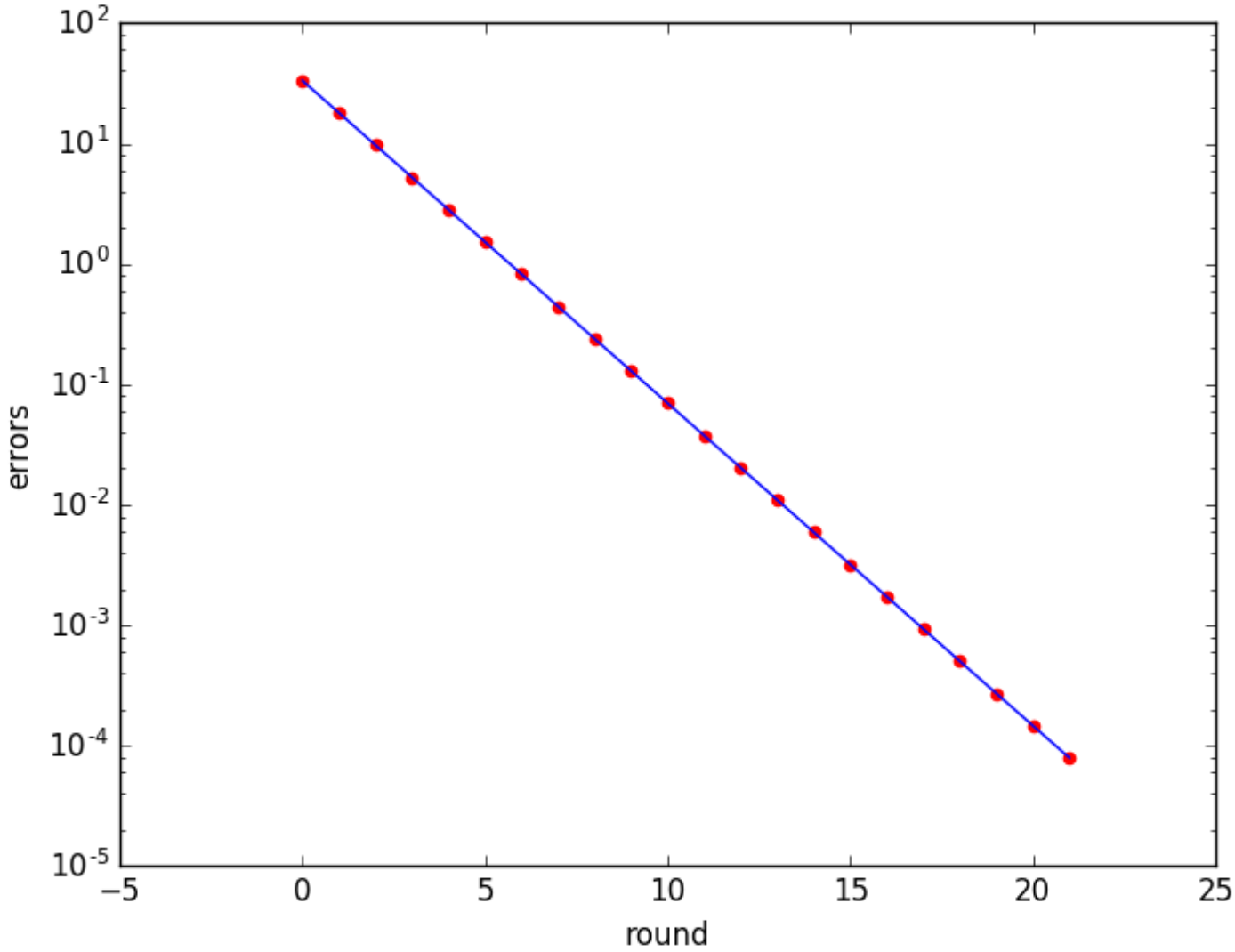
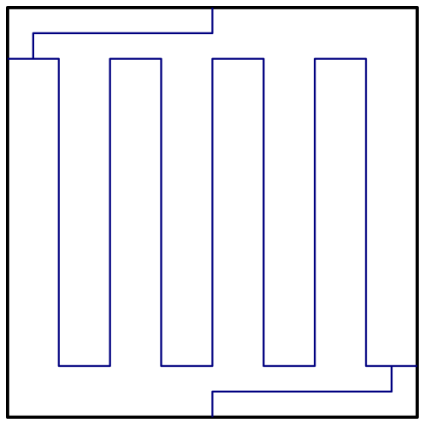



Figure: A numerical experience of the algorithm gives a very high precision of the solution.

# Outline

# Dirichlet problem on percolation cluster

- Apply the same algorithm on the same problem on percolation setting. (Gu 19+)
  - $\mathbf{a} : E_d \to \{0\} \cup [\Lambda^{-1}, 1]$.
  - $\mathbf{a}(e) > 0$ represents an *open bond* and $\mathbf{a}(e) = 0$ represents a *closed bond*.
  - Supercritical percolation $\mathbb{P}[\mathbf{a} \neq 0] = \mathfrak{p} > \mathfrak{p}_c(d)$.
  - Dirichlet problem on the maximal cluster in the cube $\square_m$.
- More technical : the random conductance also influences the domain of the solution, and the random graph structure is challenging for PDE analysis.



# Dirichlet problem on percolation cluster

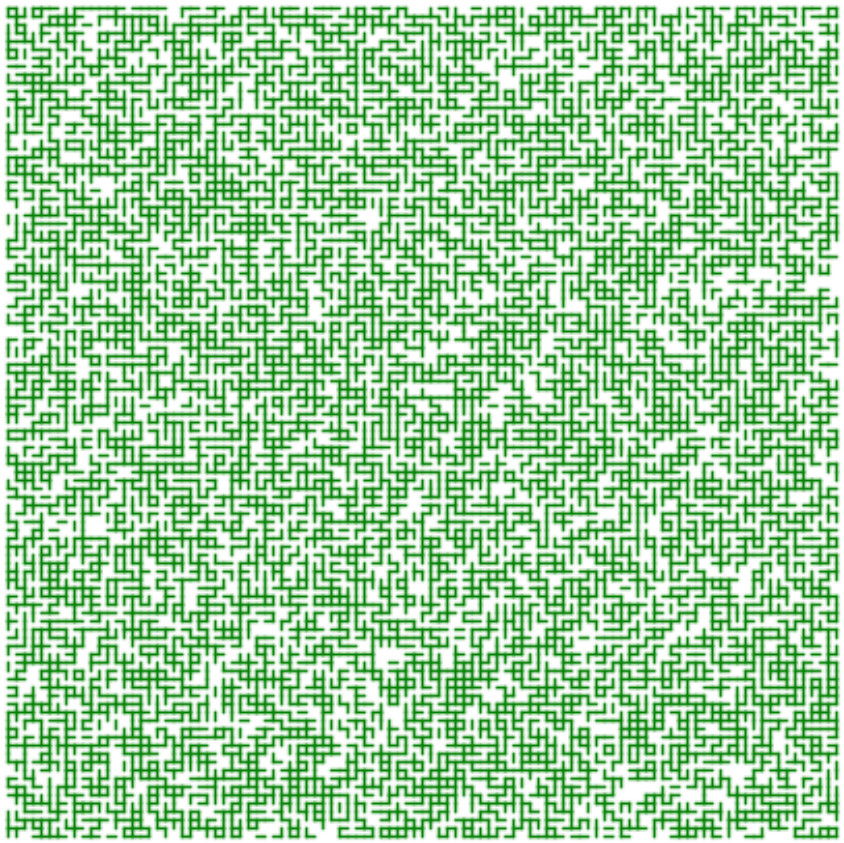


Figure: Can you tell all the connected components in the graph ?

# Dirichlet problem on percolation cluster

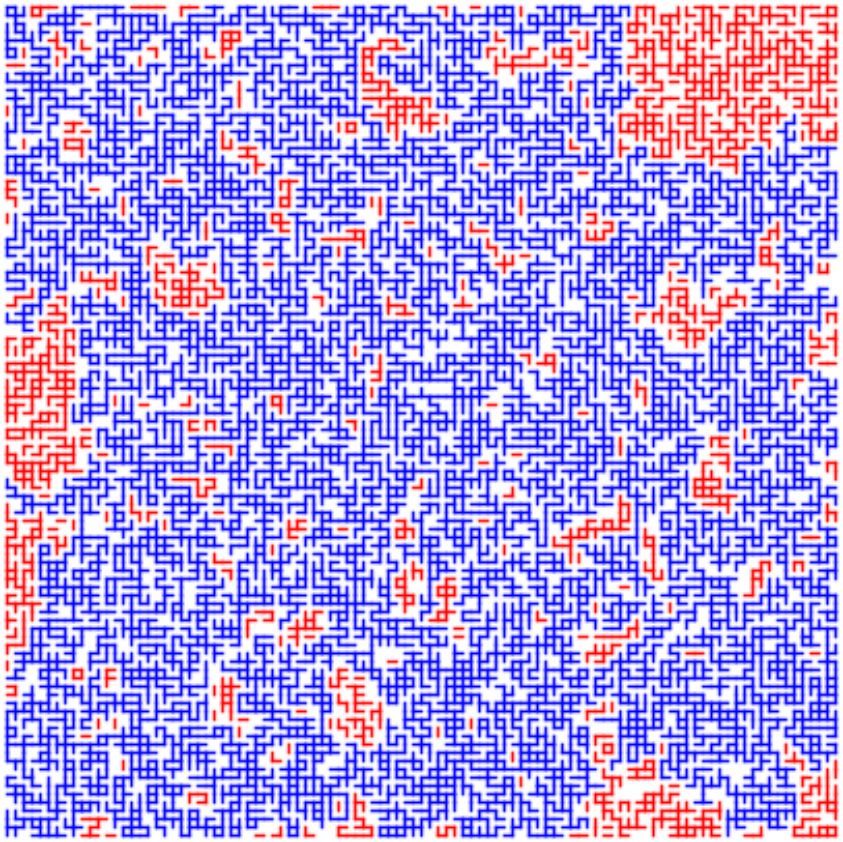


Figure: The cluster in blue is the maximal cluster in the cube

# Calculate the corrector in a cube

- $d = 2$, size $= 256 \times 256$, $p = 0.6$, $\mathbf{a} \in \{0\} \cup [0.5, 1]$, $\lambda = 0.1$.
- $-\nabla \cdot \mathbf{a} \nabla \phi_{e_1, L} = -\nabla \cdot \mathbf{a} \nabla l_{e_1}$ with null boundary condition.
- Initial error $\varepsilon_0 = 1.12085310602$.

# Calculate the corrector in a cube

| round | errors |
|---|---|
| 1 | 0.0282597982969 |
| 2 | 0.0126490361046 |
| 3 | 0.00707540548365 |
| 4 | 0.00435201077274 |
| 5 | 0.00282913420116 |
| 6 | 0.00190945842802 |
| 7 | 0.00132483912845 |
| 8 | 0.000939101476657 |

Figure: A table of errors

# Calculate the corrector in a cube

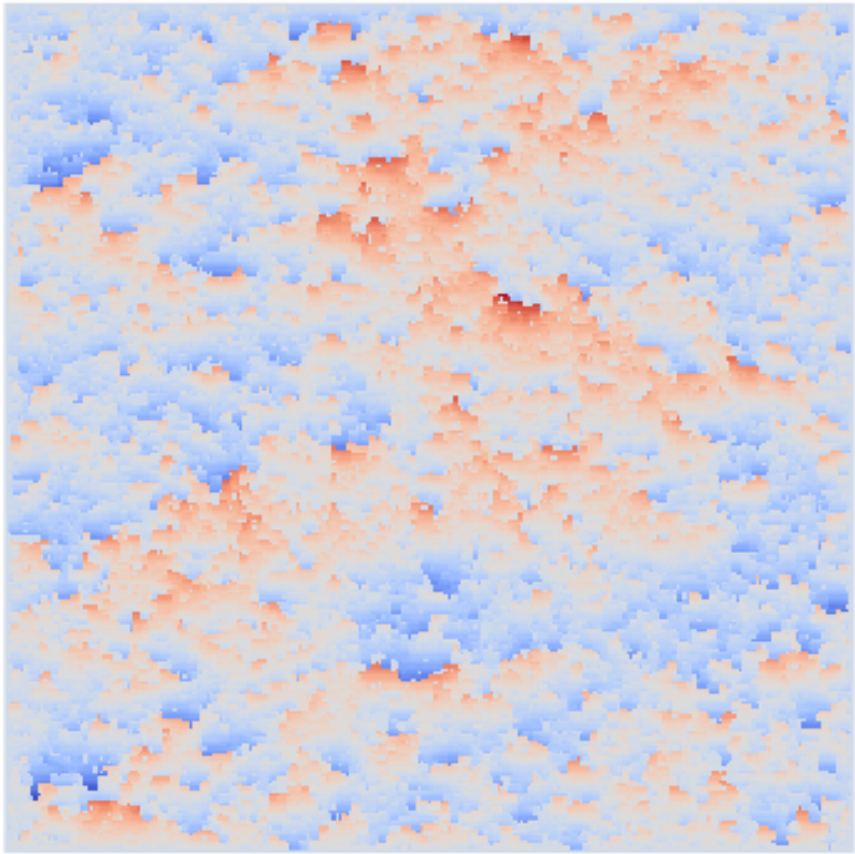


Figure: A simulation of the corrector on the maximal cluster of a cube $256 \times 256$.

# Calculate the corrector in a cube

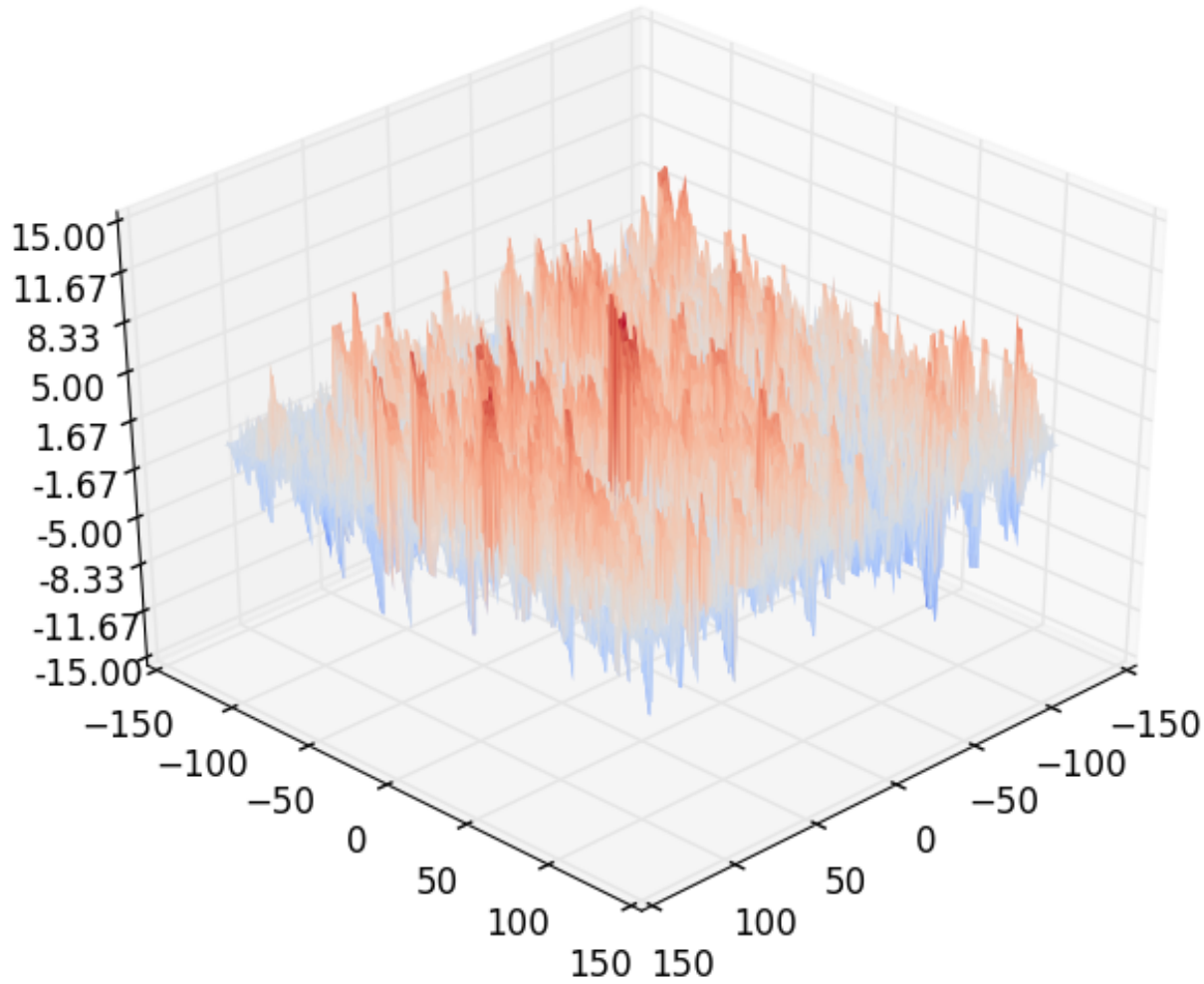
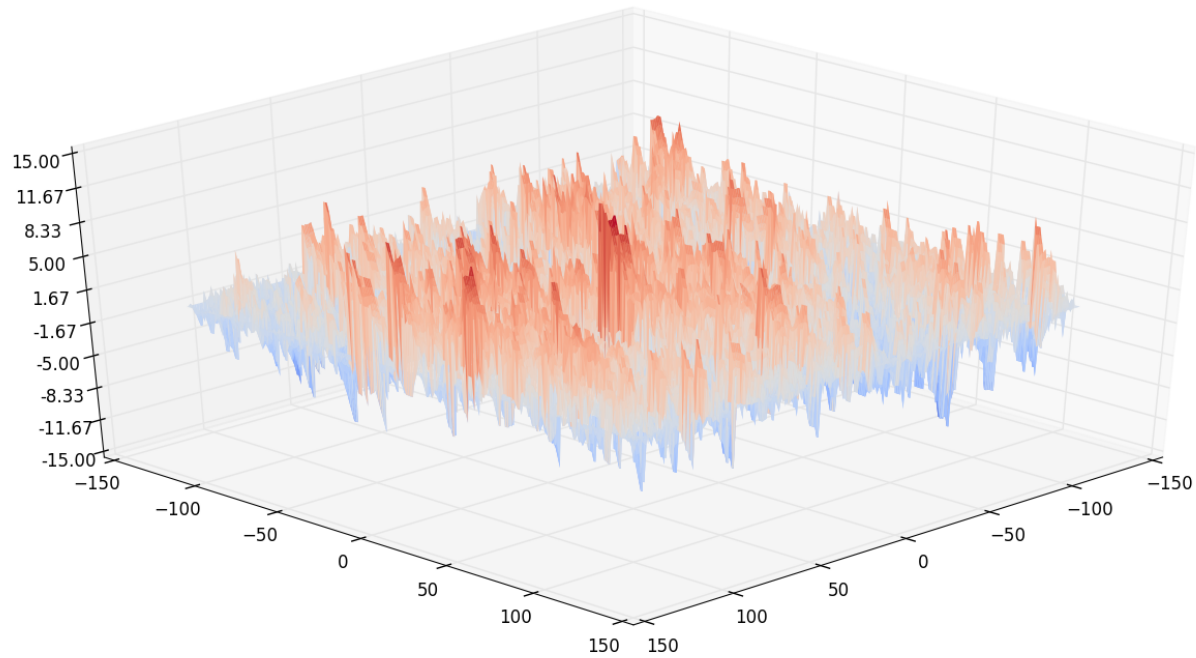



Figure: A simulation of the corrector on the maximal cluster of a cube $256 \times 256$.




# Thank you for your attention.