An efficient algorithm for solving elliptic problems on percolation clusters

Chenlin GU

DMA/ENS, PSL Research University

Les probabilités de demain

May 13, 2020

Outline for section 1

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

Random conductance model

- $(\mathbb{Z}^d, E_d)$ standard d-dimension lattice, $Q_r := \left(-\frac{r}{2}, \frac{r}{2}\right)^d \cap \mathbb{Z}^d$.
- $\mathbf{a} : E_d \to [\Lambda^{-1}, \Lambda]$ with $\Lambda > 1$. $\{\mathbf{a}(e)\}_{e \in E_d}$ i.i.d. called random conductances.
- $-\nabla \cdot \mathbf{a} \nabla$ is the discrete divergence operator defined by

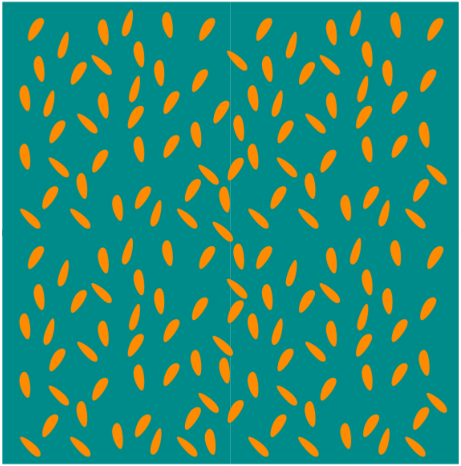
$$-\nabla \cdot \mathbf{a}\nabla u(x) := \sum_{y \sim x} \mathbf{a}(x,y)(u(x) - u(y)).$$

- Object: Find an algorithm to solve the elliptic Dirichlet problem quickly for **big** r,

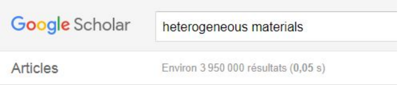
$$\begin{cases} -\nabla \cdot \mathbf{a}\nabla u = f & \text{in } \operatorname{int}(Q_r), \\ u = g & \text{on } \partial Q_r. \end{cases} \tag{1.1}$$

- Motivation: Modelisation in disordered medium and heterogeneous material.

Random stationary medium

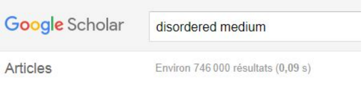


Question: What is the challenge in this problem?



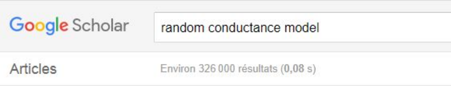
Google Scholar heterogeneous materials

Articles Environ 3 950 000 résultats (0,05 s)



Google Scholar disordered medium

Articles Environ 746 000 résultats (0,09 s)



Google Scholar random conductance model

Articles Environ 326 000 résultats (0,08 s)

Outline for section 2

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

Recap of some classical methods

Let us recall some classical method to solve this problem:

- Monte-Carlo Markov chain.
- Jacobi iterative method.
- Multigrid method.
- Homogenization.

Outline

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

MCMC

MCMC

For the case $f = 0$, the solution of Dirichlet problem is $u(x) = \mathbb{E}_x[g(X_\tau)]$ for $(X_n)_{n \geqslant 0}$ the Markov chain associated to the operator $-\nabla \cdot \mathbf{a}\nabla$ and τ the hitting time of the boundary.

- Advantages: Dimension free, easy to program.
- Disadvantages: It takes time if we want $u(x)$ for **all** $x \in \operatorname{int}(Q_r)$.

MCMC

MCMC

For the case $f = 0$, the solution of Dirichlet problem is $u(x) = \mathbb{E}_x[g(X_\tau)]$ for $(X_n)_{n\geqslant 0}$ the Markov chain associated to the operator $-\nabla \cdot \mathbf{a}\nabla$ and τ the hitting time of the boundary.

- Advantages: Dimension free, easy to program.
- Disadvantages: It takes time if we want $u(x)$ for **all** $x \in \operatorname{int}(Q_r)$.

MCMC

MCMC

For the case $f = 0$, the solution of Dirichlet problem is $u(x) = \mathbb{E}_x[g(X_\tau)]$ for $(X_n)_{n \geqslant 0}$ the Markov chain associated to the operator $-\nabla \cdot \mathbf{a} \nabla$ and τ the hitting time of the boundary.

- Advantages: Dimension free, easy to program.
- Disadvantages: It takes time if we want $u(x)$ for **all** $x \in \mathrm{int}(Q_r)$.

Outline

1 Motivation

2 Preliminary
- Monte-Carlo Markov chain
- Jacobi iterative method
- Multigrid method
- Homogenization

3 Algorithm

4 Numerical experience

5 Percolation setting

Jacobi iterative method

Jacobi iteration

- Jacobi iterative method = iteration of semigroup.
- $P(x,y) := \frac{\mathbf{a}(x,y)}{\sum_{z\sim x}\mathbf{a}(x,z)}, \tilde{f}(x) = f(x)/(\sum_{z\sim x}\mathbf{a}(x,z))$.
- We do iteration $u_0 = g$, $u_{n+1} = J(u_n, \tilde{f})$

$$\boxed{J(u_n, \tilde{f}) := Pu_n + \tilde{f}.} \tag{2.1}$$

- u is the unique solution of the equation $u = Pu + \tilde{f}$, $\lim_{n\to\infty} u_n = u$.

- Advantages: Easy to program and converges exponentially.
- Disadvantages: It takes time when Q_r is big: every iteration is a contraction $(1-\frac{1}{r^2})$, so at least $O(r^2)$ iterations.

Jacobi iterative method

Jacobi iteration

- Jacobi iterative method = iteration of semigroup.
- $P(x,y) := \frac{\mathbf{a}(x,y)}{\sum_{z\sim x}\mathbf{a}(x,z)}, \tilde{f}(x) = f(x)/(\sum_{z\sim x}\mathbf{a}(x,z))$.
- We do iteration $u_0 = g$, $u_{n+1} = J(u_n, \tilde{f})$

$$\boxed{J(u_n, \tilde{f}) := Pu_n + \tilde{f}.} \tag{2.1}$$

- u is the unique solution of the equation $u = Pu + \tilde{f}$, $\lim_{n\to\infty} u_n = u$.

- Advantages: Easy to program and converges exponentially.
- Disadvantages: It takes time when Q_r is big: every iteration is a contraction $(1 - \frac{1}{r^2})$, so at least $O(r^2)$ iterations.

Jacobi iterative method

Jacobi iteration

- Jacobi iterative method = iteration of semigroup.
- $P(x,y) := \frac{\mathbf{a}(x,y)}{\sum_{z\sim x}\mathbf{a}(x,z)}, \tilde{f}(x) = f(x)/(\sum_{z\sim x}\mathbf{a}(x,z))$.
- We do iteration $u_0 = g$, $u_{n+1} = J(u_n, \tilde{f})$

$$\boxed{J(u_n, \tilde{f}) := Pu_n + \tilde{f}.} \tag{2.1}$$

- u is the unique solution of the equation $u = Pu + \tilde{f}$, $\lim_{n\to\infty} u_n = u$.

- Advantages: Easy to program and converges exponentially.
- Disadvantages: It takes time when Q_r is big: every iteration is a contraction $(1 - \frac{1}{r^2})$, so at least $O(r^2)$ iterations.

Outline

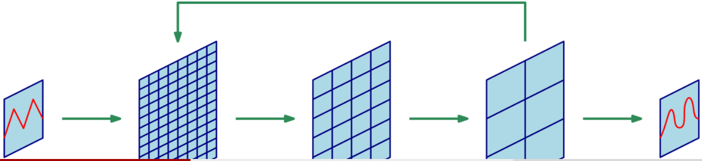
1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

Multigrid method

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.



Multigrid method

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.

- Advantages: Only $O(\log(r))$ iterations are required in Q_r.
- Disadvantages: $\mathbf{a}$ has to be constant.

Probabilistic interpretation: coarsened grid ≈ random walk with big step size.

Multigrid method

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.

- Advantages: Only $O(\log(r))$ iterations are required in Q_r.
- Disadvantages: $\mathbf{a}$ has to be constant.

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

Multigrid method

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.

- Advantages: Only $O(\log(r))$ iterations are required in Q_r.
- Disadvantages: $\mathbf{a}$ has to be constant.

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

Outline

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

Homogenized solution

Homogenized solution

- For r very big, one can use an effective conductance $\bar{\mathbf{a}}$ which is a constant matrix and the homogenized solution as an approximation

$$\begin{cases} -\nabla \cdot \bar{\mathbf{a}} \nabla \bar{u} = f & \text{in } \mathrm{int}(Q_r), \\ u = g & \text{on } \partial Q_r, \end{cases} \tag{2.2}$$

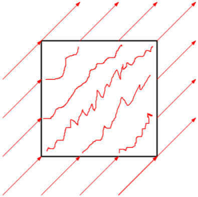
then we have

$$\|u - \bar{u}\|_{\underline{L}^2(Q_r)} := \left(\frac{1}{|Q_r|} \sum_{x \in Q_r} |u(x) - \bar{u}(x)|^2 \right)^{\frac{1}{2}} \leqslant o(r).$$

- $\bar{\mathbf{a}} \neq \mathbb{E}[\mathbf{a}]$ and can be solved much more quicker.

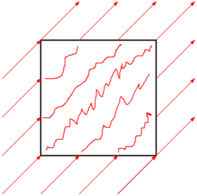
Homogenized solution

- Early work of Kozlov, Papanicolaou, Varadhan, Yurinski etc.
- Advantages: It is as fast as multigrid method.
- Disadvantages:
  - Loss of microscale information: Locally, $\bar{u}$ and u is never similar.
  - Limit of precision: For r fixed, $\frac{1}{r}\left\|u-\bar{u}\right\|_{\underline{L}^2(Q_r)} \simeq r^{-\frac{1}{2}}$ cannot be better.



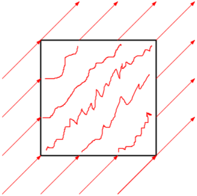
Homogenized solution

- Early work of Kozlov, Papanicolaou, Varadhan, Yurinski etc.
- Advantages: It is as fast as multigrid method.
- Disadvantages:
  - Loss of microscale information: Locally, $\bar{u}$ and u is never similar.
  - Limit of precision: For r fixed, $\frac{1}{r}\left\|u - \bar{u}\right\|_{\underline{L}^2(Q_r)} \simeq r^{-\frac{1}{2}}$ cannot be better.



Homogenized solution

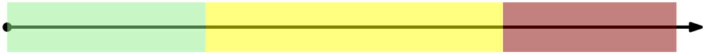
- Early work of Kozlov, Papanicolaou, Varadhan, Yurinski etc.
- Advantages: It is as fast as multigrid method.
- Disadvantages:
  - Loss of microscale information: Locally, $\bar{u}$ and u is never similar.
  - Limit of precision: For r fixed, $\frac{1}{r}\left\|u-\bar{u}\right\|_{\underline{L}^2(Q_r)} \simeq r^{-\frac{1}{2}}$ cannot be better.



Outline for section 3

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

Object of the algorithm



For r big, and for a high precision ???

Small r
a naive Jacobi interation.

Very very big r
beyond the capacity of mesh.
homogenized solution + multigrid.

Iterative algorithm

An iterative algorithm

- Initial guess $u_0 := g$.
- Solve the following equations with the null Dirichlet boundary condition:

$$\begin{cases} (\lambda^2 - \nabla \cdot \mathbf{a}\nabla) u_1 &= f + \nabla \cdot \mathbf{a}\nabla u_0 & \text{in } \operatorname{int}(Q_r), \\ -\nabla \cdot \bar{\mathbf{a}}\nabla \bar{u} &= \lambda^2 u_1 & \text{in } \operatorname{int}(Q_r), \\ (\lambda^2 - \nabla \cdot \mathbf{a}\nabla) u_2 &= (\lambda^2 - \nabla \cdot \bar{\mathbf{a}}\nabla)\bar{u} & \text{in } \operatorname{int}(Q_r), \end{cases} \tag{3.1}$$

- $\hat{u} := u_0 + u_1 + u_2$ 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

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

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

For any $s \in (0,2)$, *there exists a constant* $C(\Lambda, s, d)$ *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 λ that $\frac{1}{r} \ll \lambda \ll \frac{1}{\log^{\frac{1}{2}}(r)}$.
- Complexity $:= O(\log(r))$ iterations, very close to the one of multigrid.

Idea of proof

- 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 $(\lambda^2 - \nabla \cdot \mathbf{a} \nabla) u_2 = (\lambda^2 - \nabla \cdot \bar{\mathbf{a}} \nabla) \bar{u}$.
- The first order corrector: $\{\phi_{e_k}\}_{1 \leqslant k \leqslant d}$: $-\nabla \cdot \mathbf{a} \nabla (l_{e_k} + \phi_{e_k}) = 0$ in $\mathbb{Z}^d$.
- Two-scale expansion $w := \bar{u} + \sum_{k=1}^{d} (\mathcal{D}_{e_k} \bar{u}) \phi_{e_k}$.
- Key point: $\|\nabla (w - u)\|_{\underline{L}^2(Q_r)} \leqslant o(1)$.
- 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, Kuusi, Mourrat, Smart, Gloria, Neukamm and Otto etc.

Outline for section 4

1. Motivation
2. Preliminary
   - Monte-Carlo Markov chain
   - Jacobi iterative method
   - Multigrid method
   - Homogenization
3. Algorithm
4. Numerical experience
5. Percolation setting

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 $\varepsilon_n := \|f - (-\nabla \cdot \mathbf{a}\nabla u_n)\|_{\underline{L}^2(Q_r)}$.

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

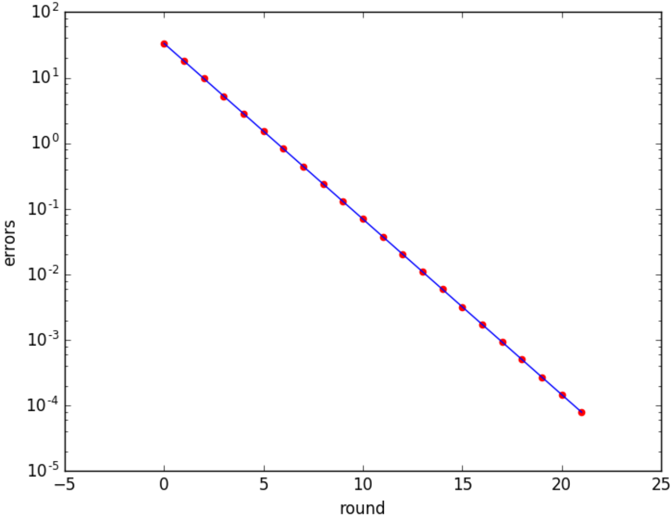


Figure: A numerical experience of the algorithm gives a very high precision of the solution.

Outline for section 5

1 Motivation

2 Preliminary

- Monte-Carlo Markov chain
- Jacobi iterative method
- Multigrid method
- Homogenization

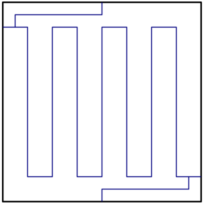
3 Algorithm

4 Numerical experience

5 Percolation setting

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 Q_r.
- 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

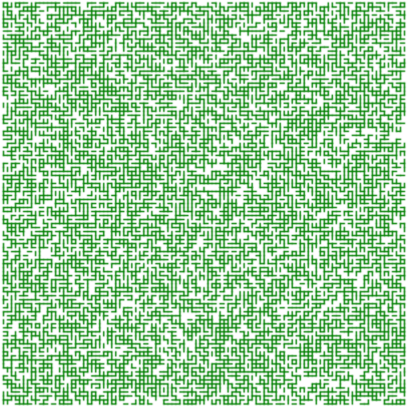


Figure: Can you tell all the connected components in the graph ?

Dirichlet problem on percolation cluster

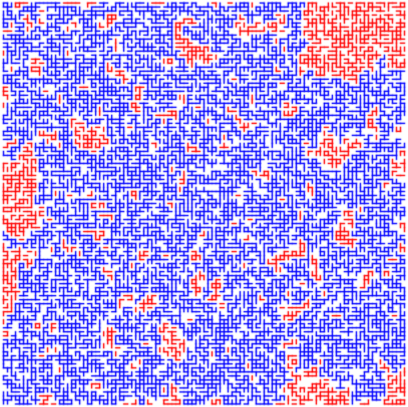
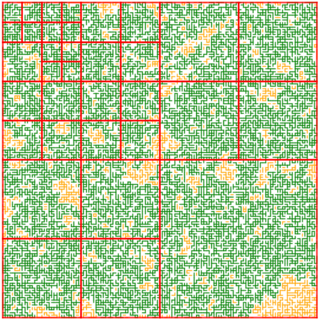


Figure: The cluster in blue is the maximal cluster in the cube

Calderón-Zygmund decomposition on cluster



A technique to decompose the cluster into small cubes so in every cube the behavior is good. See the work Armstrong and Dario (18) for regularity of harmonic function on the cluster, Dario (18+) for the corrector on the cluster, Dario and Gu (19+) for estimate of Green function on the cluster.

Calculate the corrector in a cube

- $d = 2$, size $= 256 \times 256$, $p = 0.6$, $\mathbf{a} \in \{0\} \cup [0.5, 1]$, $\lambda = 0.1$.
- $-\nabla \cdot \mathbf{a} \nabla \phi_{e_1, L} = \nabla \cdot \mathbf{a} e_1$ with null boundary condition.
- This example cannot be captured by homogenized solution.
- 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

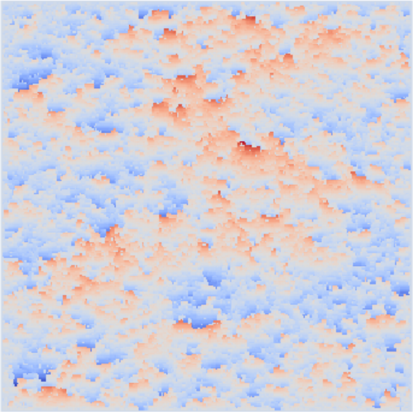


Figure: A simulation of the corrector on the maximal cluster of a cube 256×256.

Calculate the corrector in a cube

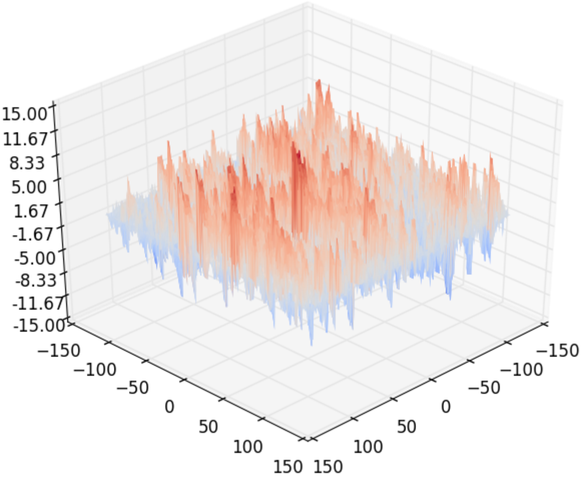
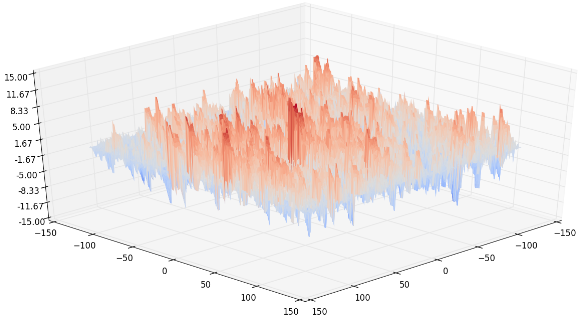


Figure: A simulation of the corrector on the maximal cluster of a cube 256×256.



Thank you for your attention.