CRITICAL SLOW DOWN EFFECT OF CONVERGENCE ON THE FORMAL ORTHOGONAL POLYNOMIALS METHOD

Reçu le 02/01/2005 – Accepté le 23/05/2005

A. SARI
S. MAHAMMED
Laboratoire de Physique Théorique, Département de Physique, Faculté des Sciences, Université Aboubekr Belkaïd, Tlemcen.

N. GHOUALI
A. BENCHAIB
Département de Mathématiques, Faculté des Sciences, Université Aboubekr Belkaïd, Tlemcen.

Résumé

Le ralentissement critique de convergence (CSDE) est l'obstacle majeur dans les simulations numériques de grands systèmes physiques. Dans notre cas pour obtenir la distribution de courants électriques sur un réseau de résistances aléatoires, au seuil de percolation, par la méthode de relaxation de Jacobi, le nombre d'itérations nécessaire pour que le système relaxe vers son état permanent croît plus vite que le volume de ce dernier. Dans cet article, nous décrivons la méthode des polynômes orthogonaux formels et commentons nos résultats. Nous montrons que le principal avantage de celle-ci est qu'elle ne possède pas de CSDE. Cependant il apparaît un nouveau type de ralentissement dû à la difficulté croissante de trouver un vecteur initial y.

***Mots clés**: filtration ; critique lente ; relaxation numérique ; polygone de forme orthogonale*

Abstract

The critical Slow down Effect (CSDE) is the major obstacle in the large scale numerical simulations of physical systems. In our case, to obtain the current distribution on a random resistor network at the percolation threshold, by the Jacobi relaxation method, the number of iterations needed for the system to relax to its steady state grows faster than the volume. In this paper, we describe technical details on the formal orthogonal polynomials method and comment our results. We show that the main advantage of this method is that there is not CSDE. However it appears a new type of slow down by the difficult choice of an initial vector y.

***Keywords**: percolation, critical slow down, numerical relaxation, formal orthogonal polynomials.*

ملخص

الإعاقة الاحراجية للتقارب في الطرق العددية بالمتكررات تطرح أمر دو شأن في حساب كميات فزيائية للمنظمات التي توجد قرب احد مفرداتها. أنشأت عدة طرق عددية تمكن التخلص من تلك الإعاقة و لكن عيبها الأكبر هو أنها تختص بتقدير مميزة معينة و بهذا الأمر تكون غير شاملة. في هذا المقال قمنا بدراسة حول طريقة أخرى تستخدم متعدد المخارج المتعامدة و يصح أن نقول في أمرها أنها شاملة و بدون إعاقة. فعلا يوجد نظرية قوية التي تشير إلى أن المتكررات تبقى محدودة إلى قياس المصفوفة المستخدمة، و لكن وجدنا أن الاختيار الغير المميز للشعاع " " يعرقل مردود الطريقة حيث يتشكل نوع جديد من الإعاقة الاحرجية للتقارب.

<u>الكلمات المفتاحية</u>الترشيح،الإعاقة الإحراجية، استرخاء عددي، عجيد أضلع قائم.

The study of physical systems with critical phenomena, has gained interest these last decades [1-4]. It is well established that systems, with random structure, have properties where scaling laws lead to exponents depending strongly on the dimension of the space; and where the microscopic details are irrelevant [5]. For example, we can cite the diffusion in particular lattice [6], the electrical transport on a random resistor network [7], or the mass transport in a porous media [5, 6].

In such systems the evaluating of physical quantities using methods of numerical simulation based on relaxation algorithms, is confronted to the problem of CSDE. This is a serious problem in the computations of physical quantities on structures close to singularities. There are numerical methods that allow to avoid this CSDE, however, their inconvenience is to be specific to a given property [7-10].

The main purpose of this work is to apply a method based on the formal orthogonal polynomials (or Lanczos-type algorithm), which seems enough general and avoids CSDE.

In exact arithmetic, the number of iterations, to obtain the solution, remains limited by the matrix size. Unfortunately, we will see that the choice of an arbitrary vector "y", satisfying the orthogonally conditions, restricts the efficiency of this method. The purpose of the present paper is to expose technical details on a new method based on formal orthogonal polynomials ideas. This method was used already Friedrichs *et al.* [11] to determine $P_0(t)$, the probability of returning to the initial position at time, for a walker on percolating systems.

In this paper we apply the method to the bond percolation on two-dimensional square lattice, and where is written, at each site, discretised Poisson's equation[30; 31]. The size L of square lattice is times the lattice constant a (a=1). There are two types of links connecting neighbouring sites: they have either unit conductance with probability p equation[30; 31]. The size L of square lattice is times the lattice constant a (a=1). There are two types of links connecting neighbouring or zero conductance (link hqing been cut) with propability (1-p).

We can rewrite this as:

$$p(r_{ij}) = (1-p)\delta(r_{ij}) - p\delta(r_{ij}-1), \qquad (1)$$

Let a matrix A, where A_{ij}, matrix element, represent the link value between sites i and j, V_i the voltage on the i-th site, and the elements of the vector I are the constraints on each node. We apply unit voltage between two bars put on opposite edges of the network, and periodical condition on the other edges. Then the steady state of the system is giving by equation:

$$AV = I \qquad (2)$$

To solve the above equation, we can write the diffusion equation:

$$dV/dT = AV-I \qquad (3)$$

and the steady state solution of this equation gives the solution of equation (2). The discrete form of equation (3) is the well know relaxation process in numerical analysis:

$$V^{(k+1)} = A.V^{(k)} - I, \qquad (4)$$

At the percolation threshold, the system not relax to its steady state with a number of iterations growing like the volume but like $L^{d_f+d_w}$, where d_f is the fractal dimension of the backbone of percolating cluster and d_w is the random walk dimension [12], and it is easy to see that $d<d_f+d_w$.

The similarity between diffusion and relaxation processes enables to understand qualitatively the CSDE. On damaged network, at the percolation threshold, during its spreading out, the random walk is subject to forced crossings, dead arms and winding paths. Consequently, when the system size grows, the convergence to the solution in equation (4) becomes slower.

When we study the transport in disordered systems, the analytical techniques: effective medium theories [13], continuous-time random walks [14], series expansions [14], renormalization-group techniques [16] and scaling arguments [17], are not sufficient to understand their properties. In fact, in much case, these methods have provided only qualitative descriptions.

An other hand Monte Carlo simulations [18] give formally good solutions, unfortunately averages must be performed over time and cluster configurations, and thus the CSDE can occur. There exist methods which avoid the CSDE, like matrix transfer methods [7] or Fourier acceleration methods [19], but they are ad hoc processes.

In this paper, we present an approach based on theory of formal orthogonal polynomials [20]. A priori, this method seems exempt from the CSDE.

This paper is set out as follows. In section II, the method is discussed in detail. The results of calculations on the percolating cluster are presented in section III, compared with relaxation method and then analyzed. We conclude in section IV with a brief discussion of significance of our results, also we introduce a qualitative representation.

ORTHOGONAL POLYNOMIALS METHOD (OPM)

Initially the theory of formal orthogonal polynomials appear in the papers of Lanczos [21, 22]. In exact arithmetic, the Lanczos algorithm would terminate at the most (L^2)-th recursion [23]. The Lanczos algorithm is based on the constructing of a sequence of orthogonal polynomials:

If AV=I is a n linear equations system, where $A \in \Re^{\nu\xi\nu}$ is symmetric and positive definite, $V \in \Re^{\nu}$ and $I \in \Re^{\nu}$, and considers the functional Φ(V) defined by : $\Phi(V) = \frac{1}{2}V^T AV - X^T I$, it follows that $V= A^{-1}I$ is the unique minimizer of Φ(V) [10]. If V_0 and *y* are two nonzero vectors in ∇^{ν}, one way to produce a sequence $\{V_k\}$ that converges to V is to generate a sequence of vectors $\{r_k\}$ defined by

$$r_k = I - AV_k = r_0 - A^{-1}.(V_k - V_0) = \text{span}\,(1, A, ..., A^k).r_0\,,$$

$$= r_0 + \alpha_1 A r_0 + ... + \alpha_k A^k r_0 = P_k(A).r_0$$

where :

$$r_0 = I - AV_0$$

and let

$$\text{span}\,(r_0, Ar_0, ... A^{k-1}r_0) \perp \text{span}\,(y, A^T y, ..., A^{T(k-1)}y),$$

where A^T denotes the transpose of *A*.

This othogonality condition can be write as:

$$(A^{T^i}y, r_k) = 0 \text{ for } i = 0,...,k-1.$$

Knowing P_k, r_k can be computed by :

$$r_k = P_k(A)r_0\,,$$

and *V-k* is obtained by:

$$V_k = V_0 - R_{k-1}(A)r_0$$

where :

$$P_k(\xi) = 1 + \xi R_{k-1}(\xi)\,.$$

This is Lanczos method [24].

Table 1: For the OPM and JRM, the number of iterations per realization needed to reach an accuracy ≤10^{-7}.

Size	78		136	
P	OPM	JRM	OPM	JRM
0.9	41	79	52	120
0.8	46	139	64	149
0.7	49	182	77	Div.
0.6	54	Div.(*)	80	Div.
0.55	60	Div.	65	Div.
0.52	56	Div.	75	Div.

(*) The abbreviation Div. means that the number of iterations have much more great than the system size.

The family $\{P_k\}$ also satisfies a three-term recurrence relationship, which can be write as :

$$P_{k+1}(\xi) = P_k(\xi) - \beta_k \xi Q_k(\xi)$$

where : $Q_k(\xi) = P_k(\xi) + \alpha_k Q_{k-1}(\xi)$

with : $\alpha_k = -c(\xi U_{k-1}P_k)/c(\xi U_{k-1}Q$

and : $\beta_k = c(U_k P_k)/c(\xi U_k Q_k)$.

{Ui} is an arbitrary family of polynomials such that $\forall i, U_i$ has exactly the degree *i*, and c is the linear functional on the space of polynomials defined as :

$$c(\xi^i P(\xi)) = \sum_{j=0}^{k} \alpha_j (y, A^{i+j}) r_0 = \sum_{j=0}^{k} \alpha_j (A^{T^i}y, A^j) r_0\,.$$

There exists various possibilities for computing recursively the vectors r^k $=P_k(A)r_0$ [25]. They give rise to different algorithms. For our application, we shall use the method so-called biconjugate gradient method, or the BIOMIN method.

We give the algorithm that we have used to study the current distribution of random resistor networks in two-dimensions at the percolation threshold :

```
- Choose $V_0$, y
- Set $r_0 = I - AV_0 = p_0$; $\tilde{r}_0 = \tilde{p}_0 = y$
- For k = 0, ….; compute :
        $\beta_k = (\tilde{r}_k, r_k)/(\tilde{p}_k, Ap_k)$
        $V_{k+1} = V_k + \beta_k p_k$
        $r_{k+1} = r_k - \beta_k Ap_k$
        $\tilde{r}_{k+1} = \tilde{r}_k - \beta_k A^T \tilde{p}_k$
- If $r_{k+1} \neq 0$, then compute :
    $\alpha_{k+1} = (\tilde{r}_{k+1}, r_{k+1})/(\tilde{r}_k, r_k)$
    $p_{k+1} = r_{k+1} - \alpha_{k+1} p_k$
    $\tilde{p}_{k+1} = \tilde{r}_{k+1} - \alpha_{k+1} \tilde{p}_k$
- Else give the result.
```

NUMERICAL RESULTS AND DISCUSSION

At the percolation threshold, the conductance of the system is due only to the incipient infinite cluster, and consequently this reduces the size of the matrix A. We tested the OPM and compared it to the Jacobi relaxation method (JRM) [23, 26], for 2-dimensional lattices of size 78, and 136. Table 1 shows the number of iterations needed for OPM and JRM, with different rate of damage (1-p) = q = 0.1, 0.2, 0.3, 0.4, 0.45, and 0.48,. The main advantage of the JRM is that the code is very simple. However, according to table 1, the convergence may be very slow compare to the OPM.

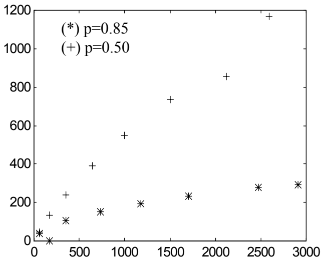

Figure 1: Plot of the number of iteration versus matrix size, for p=0.50 (*) and p=0.85 (+). The using method is OPM.

The number of iterations N, which gives a solution with accuracy 10^{-7}, is evaluate on matrix ranged in size from 44 to 2650. In order to avoid important fluctuations in the values, we will keep only matrix in a small range-size for each given size of the network (averaging over 20 configurations).

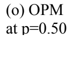

Figure 2: Plot of the number of iteration versus matrix size, at p=0.50 and for the two methods OPM and JRM.

The OPM is applied at p =p$_c$ =0.5 (percolation threshold) and p=0.85. In figure 1 the graphs for the numbers of iterations versus size of matrix for p=0.5 and p=0.85 are shown. We observe that the number of iterations increases strongly with the size (10^3 times as big in our case) when p=0.5. But, at the percolation threshold, the OPM is faster than JRM as seen in figure 2.

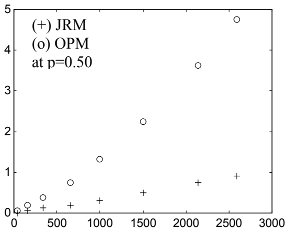

Figure 3: The time iteration (in arbitrary unit) versus matrix size, at p=0.50 and for the two methods OPM and JRM.

We want to answer at the question: the rapidity of the OPM is due to time of one iteration or to number of iterations? In figure 3 we notice that the time of one iteration versus the matrix-size for the OPM exhibits increasing more than for the JRM, while we have an increase relatively slower in the iteration-number (see figure 2). Therefore, the iteration-number behaviour versus the size of the matrix is the main cause for the rapidity of the OPM. These observations are corroborated by the plot in figure 4, which shows the variation of iteration-number per unit of size with the size of the matrix. As seen in this figure, the variation decreases exponentially.

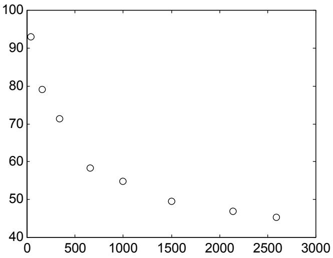

Figure 4: The iteration number per unit matrix size versus the size of matrix at p=0.50 and for the OPM.

A second manifestation of the OPM is represented in figure 5. In fact the vector *y* must verify the linear independence of {y, A^Ty, …, $A^{T(n-1)}$y}, and we have any indication to determine *y*.

Figure 5: The number of "bad y" versus the size of the matrix. The results are averaged over 100 trials.

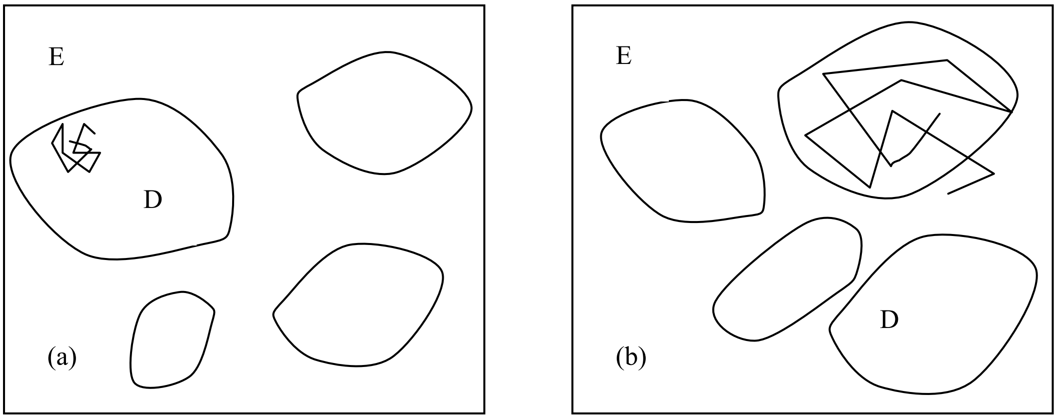

Figure 6: Qualitatively, we represent the space of polynomials (E) and the domains (D), non convexe, in solid curve, the regions where the polynomials are orthogonal. The round off error in computation generate a random walk in this space (broken line). When the size of the network is small the random walk has little steps (a), when the size of the network is large big steps (b).

In our algorithm, we choice randomly the vector *y*. In this latest figure, we show that when the size of the network increases, firstly, statistically the rapidity with which we obtain a "good" *y* increases, and this decreases rapidly. We think that there is two reasons to explain this behaviour. In computer, we use a finite-precision arithmetic, which introduces round off error. On the other hand, when the size of the network increases, there is a parcelling up of the set of vectors *y* checking the orthogonally conditions.

That being the case, qualitatively, we assume that when we start with a good y, because of round off error a new y is obtained. We can represent this by a random walk in formal-polynomials-space When the size of network increases the jump of the walker is more wide, and this effect increases the probability that "y" falls out the convergence domain (see figure 6).

REFERENCES

[1] R. Kopelman, J. Stat. Phys. 42, 185 (1986).

[2] W. D. Dozier, J. M. Drake, and J. Klafter, Phys. Rev. Lett. 56, 197 (1986).

[3] P. Meakin, Phys. Rev. A34, 710 (1986).

[4] R. Rammal, C. Tannous, P. Breton, and A.-M. S. Tremblay, Phys. Rev. Lett. 54, 1718 (1985).

[5] see, for example, K. Huang, "Statistical Mechanics" (J. Wiley & Sons, 2nd ed. 1987).

[6] S. Havlin and D. Ben-Avraham, Adv. Phys. 36, N°. 6, 695 (1987)

[7] M. Barthélémy, S.V. Buldyrev, S. Havlin and H.E. Stanley, arXiv:cond-mat/0002252v2, 23 Feb 2000.

[8] J. Adler, A. Aharony, R. Blumenfeld, A. B. Harris, and Y. Meir, Phys. Rev. B 47, 10, 5770 (1993).

[9] H. Tanino, A. Kuprin, and H. Deai, Phys. Rev. B 53, 4, 1937 (1996).

[10] B. Derrida, J. G. Zabolitzky, J. Vannimenus, and D. Stauffer, J. Stat. Phys 36, Nos. 1/2, 31 (1984).

[11] C.J. Lobb, and D. J. Frank, J. Phys. C 12, L827 (1979).

[12] J. Hoshen, and R. Kopelman, Phys. Rev. B 14, 3428 (1976).

[13] H. Sher, and M. Lax, Phys. Rev. B 7, 4491 (1973).

[14] M. S. Friedrichs, and R. A. Friesner, Phys. Rev. B 37, No. 1, 308 (1988).

[15] L. de Arcangelis, S. Redner, and A. Coniglio, Phys. Rev. B 31, 4725 (1985).

[16] R. Orbach, J. Stat. Phys. 36, 735 (1984).

[17] A. Blumen, G. Zumofen, and J. Klafter, J. Luminescence 45, 90 (1990).

[18] R. Fisch, and A. B. Harris, Phys. Rev. B 18, 416 (1978).

[19] J.-P. Hovi, and A. Aharony, in Proc. Of the Nonlinear and Random Processes, Taipei,
18-24 july, p. 68, ed. C.-K. Hu and K.-T Leung, Elsevier (1995).

[20] T. Nakayama, and K. Yakubo, ReV. of Modern Phys. 66, No 2, 381 (1994).

[21] J.-S. Wang, and R.H. Swendsen, Physica A 167, 565 (1990).

[22] G. G. Batrouni, A. Hansen, and M. Nelkin, Phys. Rev. Lett. 57, No. 11, 1336 (1986).

[23] C. Lanczos, J. Rs. Natl. Bur. Stand. 45, 255, (1950).

[24] C. Lanczos, J. Rs. Natl. Bur. Stand. 49, 33, (1952).

[25] C. Brezinski, M. Redivo Zaglia, and H. Sadok, Numer. Math. 83, 53, (1999).

[26] G.H. Golub, and C. F. Van Loan, Matrix Computations, ed. The Johns Hopkins Univ. Pres., (1989).

[27] C. Brezinski, M. Redivo Zaglia, and H. Sadok, Numer. Math. 83, 53 (1999).

[28] C. Brezinski , and H. Sadok, Lanczos type algorithm for systems of linear equations, Publication ANO – 272, may 1992, USTL (Lille-France).

[29] O. Axelsson, Lectures notes in Mathematics 572, Springer-Verlag, Berlin, Heudelberg, New-York, pp. 2-51 (1977).

[30] C. Penneta, L. Reggiani, Gy. Trefán and E. Alfinito, arXiv.cond-mat/0202268 v1, 15 Feb 2002.

[31] O; Stenull and H.-K. Jansen, arXiv.cond-mat/0110560 v1 26 Oct 2001.