Efficient Green's-function approach to finding the currents in a random resistor network

Kang Wu and R. Mark Bradley
Department of Physics, Colorado State University, Fort Collins, Colorado 80523
(Received 11 August 1993)

Using Green's functions, we reformulate Kirchhoff's laws for a two-component random resistor network in which a fraction p of the resistors has conductance σ_- and the remainder have conductance σ_+. In this Green's-function formulation (GFF), the current correlation between any two resistors in the network is explicitly taken into account. The GFF yields a linear system equivalent to Kirchhoff's laws but with a smaller number of variables. In the dilute case ($p \ll 1$), the voltages can be calculated directly with very high speed using the GFF. For general p, a variety of algorithms can be used to solve the GFF linear system. We present the technical details of solving the GFF linear system using the conjugate gradient method (method A). Our extensive numerical work shows that method A consistently requires fewer iterations than solving Kirchhoff's laws directly using the conjugate gradient method (method B). For example, for a 128×128 grid with $p \geq 0.65$ and $\sigma_-/\sigma_+ \leq 10^{-4}$, the number of iterations needed to achieve a precision of 10^{-10} is more than 100 times smaller in method A than in method B.

PACS number(s): 02.70.−c, 64.60.Ak, 77.22.Jp, 05.70.Ln

I. INTRODUCTION

Kirchhoff's equations for a random medium must often be solved in statistical physics and materials science. The two-component random resistor network [1,2] is one of the most important problems of this kind, and is of enduring interest. This is because many materials are random and inhomogeneous, and this simple model captures the essential physics of many disordered media. In a two-component random resistor network, each resistor has conductance σ_- with probability p, or conductance σ_+ with probability $1-p$. (We may assume that $\sigma_+ \geq \sigma_-$ without loss of generality.) When $\sigma_-/\sigma_+ = 0$, the system undergoes a phase transition at the percolation threshold $p = p_c$. For example, if $\sigma_- = 0$, the network conductivity is zero for $p \leq p_c$, and is nonzero for $p > p_c$.

In real two-component mixtures, the ratio $h \equiv \sigma_-/\sigma_+$ is not zero, but it can be very small. For any finite nonzero h, there is no phase transition, and so this ratio plays a role similar to the external field in a ferromagnet [3,4]. A scaling form for the conductivity has been proposed that describes the development of the singularity as $h \to 0$ [3]. Similar scaling forms have been advanced for the cumulants of the macroscopic resistance fluctuation [5–7]. Because much more computing time was needed to perform simulations for small, nonzero h than for $h = 0$, numerical tests of these scaling forms were limited to relatively small lattices.

There is currently much interest in random media that are changed irreversibly by an applied electric field [8]. These so-called "breakdown problems" are a second example of a class of models in which Kirchhoff's laws for a random network must be solved. "Burn out" of random fuse networks [9–12], dielectric breakdown [10, 12–19], and the onset of superconductivity in granular superconductors [20–24] have all been studied using breakdown models. Because of their enormous complexity, these systems have been studied principally by Monte Carlo simulations. In these simulations, Kirchhoff's laws must be solved repeatedly, and most of the computing time is spent in solving these linear equations. The number of linear equations is proportional to the number of lattice sites in the system. Therefore, the amount of computing time needed increases rapidly with the system size, and, as a result, most simulations have been limited to relatively small systems. More efficient techniques for solving Kirchhoff's laws are desirable because they would allow us to more closely approach the infinite size limit.

To make these considerations more concrete, let us consider a specific breakdown problem for a moment. The model we shall consider is a variant of the model of dielectric breakdown introduced by Takayasu [13]. Every nearest-neighbor pair of sites in a $L \times L$ square lattice is connected by a resistor. Initially, each resistor either has conductance σ_+ with probability $1-p$, or conductance σ_- with probability p. The conductance σ_+ is much larger than σ_-, and σ_- is nonzero. Busbars are connected to the top and bottom of the network. For $p > p_c = \frac{1}{2}$, the network is initially an insulator with very low conductivity. A voltage is now applied across the two busbars, and its magnitude is steadily increased. When the voltage across a resistor with conductance σ_- reaches a specified threshold value, it irreversibly "breaks down"—it becomes a resistor with conductance σ_+. The voltage difference across the busbars is increased until they are connected by a path of resistors with conductance σ_+. The network as a whole has then broken down.

In this model, whenever the conductance of a resistor changes, the voltage at each site in the network must be calculated all over again. Typically, a large number of resistors in the network breaks down before the network as a whole becomes conducting. Therefore, a tremendous number of computational steps must be performed in a single simulation. This is a common feature of Monte

 ©1994 The American Physical Society

Carlo simulations of all breakdown processes.

In both the two-component random resistor network and in our model of dielectric breakdown, it is particularly important to be able to solve Kirchhoff's laws efficiently for a random resistor network with small nonzero values of h. Typically, an iterative process is used to solve Kirchhoff's equations for a random resistor network. In such an iterative process, we start with some initial value for the voltage at each site in the network. The voltages at the $(n+1)$th iteration are then calculated from their values at the nth iteration by some specified means. This process is repeated until the desired precision is reached. A large number of operations must be performed at each iteration. For example, for an $L \times L$ square grid, the number of operations that must be performed at each iteration is usually $O(L^2)$. The number of iterations required to reach a certain precision is, accordingly, an important gauge of the effectiveness of an iterative technique. The iterative numerical method most frequently used to solve Kirchhoff's equations is the conjugate-gradient method [25–28].

The number of operations required to solve Kirchhoff's equations using the conjugate-gradient method depends not only on the size of the lattice, but on the value of p as well. For example, for a two-component random resistor network with $\sigma_- = 0$, the number of iterations required to reach a given precision grows faster than the number of sites in the lattice at the percolation threshold $p = p_c$ [29,30]. This is called "critical slowing down." Critical slowing down occurs because at $p = p_c$ there are long-range correlations between sites in the resistor network.

The effects of critical slowing down can be reduced significantly by employing the multigrid method [31]. For example, it only takes 23 iterations on average to calculate the voltages to a precision of 10^{-12} in a 400×400 square grid at the percolation threshold using the multigrid method. In contrast, about 1300 iterations are required to solve Kirchhoff's equations for a 64×64 square grid at $p = p_c$ to a precision of 10^{-10} using the conjugate-gradient method [30].

Fourier acceleration [29,30] is a much simpler approach than the multigrid method for alleviating the effects of critical slowing down. In this approach, the long-range correlations in the resistor network at its critical point are taken into account through the use of a preconditioning matrix E [28]. For each pair of sites $(\mathbf{r},\mathbf{r}')$ in the lattice, there is an entry $E_{\mathbf{r},\mathbf{r}'}$ in the preconditioning matrix. In the Fourier acceleration approach of Batrouni and co-workers [29,30], $E_{\mathbf{r},\mathbf{r}'}$ is taken to be $|\mathbf{r}-\mathbf{r}'|^{-\phi}$, and the value of the exponent ϕ is chosen to optimize the performance. Numerical tests show that Fourier acceleration significantly improves performance for $p = p_c$ and $\sigma_- = 0$. For example, it only takes about 392 iterations to solve Kirchhoff's equations for 64×64 square grid with $p = \frac{1}{2}$, $\sigma_+ = 1$, and $\sigma_- = 0$ to a precision of 10^{-10} using the Fourier-accelerated conjugate-gradient method (FACG) [30]. In contrast, 1287 iterations are required to solve the same set of Kirchhoff equations using the conjugate-gradient method without Fourier acceleration [30].

Although the successes of the FACG method are impressive, the technique has some serious drawbacks. The optimal value of the exponent ϕ must be determined by repeated runs for different choices of the exponent. There are indications that the efficiency of the algorithm depends sensitively on the choice of ϕ [32], and so this optimization process could be rather time consuming. In addition, the best choice for ϕ depends on the values of p, L, and h. This means that in most simulations, the optimization process must be repeated several times if the FACG is to be close to optimal efficiency for all parameter values of interest.

The power-law form chosen for $E_{\mathbf{r},\mathbf{r}'}$ was motivated by the fact that the correlation function decays as a power law at the critical point. If $p - p_c$ is nonzero, the correlation function has an exponential cutoff at large distances [2]. Presumably, if the FACG is to be efficient away from the critical point, this would have to be taken into account. The preconditioning matrix would then depend on two parameters: ϕ and a cutoff length. In principle, both of these parameters would have to be adjusted to give optimal performance, a task that seems daunting indeed.

All the simulations of random resistor networks and breakdown problems performed to date are based on solving Kirchhoff's equations directly. In this paper, we show that the speed of convergence can be improved significantly by using an entirely different approach: We calculate the voltage at each site in the network using a Green's-function formulation (GFF) of Kirchhoff's laws instead of solving Kirchhoff's equations directly. The GFF yields a linear system of equations equivalent to Kirchhoff's equations; we refer to this system as the GFF linear system.

This approach has a number of attractive features. Many algorithms for the solution of systems of linear equations, e.g., the conjugate-gradient method and the multigrid method, can be used to solve the GFF linear system. Moreover, in the GFF, the current correlation between any two bonds in the network is explicitly taken into account, and this leads to a substantial increase in the speed of the convergence. Only one of the two kinds of resistors appears in the GFF linear equations. As a result, the dimension of the space in which we must search for the solution is reduced. Rounding errors drastically lower the speed of the conjugate-gradient method when h is small, but are not a problem for GFF-based algorithms in this regime. Finally, the GFF can be used for all values of p, L, and h.

To demonstrate the utility of the GFF, we show how it can be combined with the conjugate-gradient method. Our extensive numerical tests of this method show that it is a particularly efficient method for solving two-component random resistor networks with small nonzero values of h. As we noted earlier, this limit is of special importance in the study of two-component random resistor networks and in breakdown problems. Our numerical tests also show that when p is sufficiently small and $\sigma_- = 0$, it is more efficient to use the conjugate-gradient method to solve the GFF linear system than to employ it to solve Kirchhoff's laws directly.

The paper is organized as follows: In Sec. II, we con-

struct the required Green's function and derive the GFF linear system. In Sec. III, we show how the GFF can be combined with the conjugate-gradient algorithm, and test the efficiency of this method. Finally, in Sec. IV, we give our conclusions.

II. GREEN'S-FUNCTION FORMULATION OF THE RANDOM RESISTOR NETWORK

Consider a square grid in two dimensions with N_x sites in the x direction, and N_y sites in the y direction. Let the lattice spacing be a. We place a resistor on each bond in the lattice. The conductance of a resistor can be either σ_+ or σ_-, where $\sigma_+ \geq 0$ and $\sigma_- \geq 0$ are arbitrary but fixed. Busbars are placed on the top and the bottom of the network, and a voltage difference is applied across these two busbars. We let the voltage on the top busbar be V_0, and ground the bottom busbar. Periodic boundary conditions are imposed in the x direction. Let $V(\mathbf{r})$ be the voltage at the point $\mathbf{r}$ in the network. The boundary conditions are then

$$V(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a}=V_0, \quad V(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{y}}=0}=0 , \tag{1a}$$

and

$$V(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{x}}=0}=V(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{x}}=N_x a} . \tag{1b}$$

Here $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are unit vectors in the x and y directions, respectively.

Throughout the paper, we will call a bond in the y direction a vertical bond, and a bond in the x direction a horizontal bond. If a bond connects the point (x,y) to the point $(x+a,y)$, we refer to this bond as a horizontal bond located at (x,y). Similarly, if a bond connects (x,y) and $(x,y+a)$, we refer to this bond as a vertical bond located at (x,y). To uniquely specify a bond we must give both a coordinate and a direction. We will use the vector $\mathbf{s}$ to distinguish the two bonds located at the point (x,y). This vector can only take on two values: $\mathbf{s}=\mathbf{s}_y\equiv a\hat{\mathbf{y}}$ for a vertical bond, and $\mathbf{s}=\mathbf{s}_x\equiv a\hat{\mathbf{x}}$ for a horizontal bond. We label a bond located at $\mathbf{r}$ with direction $\mathbf{s}$ by $(\mathbf{r},\mathbf{s})$. In order to express several relationships succinctly, we define the translation operator $\hat{l}_{\mathbf{s}}$ and the operator $\hat{\rho}_{\mathbf{s}}(\mathbf{r})$ as follows:

$$\hat{l}_{\mathbf{s}}\mathbf{r}\equiv\mathbf{r}+\mathbf{s}, \quad \hat{l}_{-\mathbf{s}}\mathbf{r}\equiv\mathbf{r}-\mathbf{s}, \quad \hat{\rho}_{\mathbf{s}}(\mathbf{r})\equiv\hat{l}_{\mathbf{s}}-1 ,$$

and

$$\hat{\rho}_{-\mathbf{s}}(\mathbf{r})\equiv\hat{l}_{-\mathbf{s}}-1 ,$$

where the parameter $\mathbf{r}$ in $\hat{\rho}_{\mathbf{s}}(\mathbf{r})$ and $\hat{\rho}_{-\mathbf{s}}(\mathbf{r})$ indicates that the operators $\hat{l}_{\mathbf{s}}$ and $\hat{l}_{-\mathbf{s}}$ act on $\mathbf{r}$. If $I_{\mathbf{s}}(\mathbf{r})$ is the current passing through the bond $(\mathbf{r},\mathbf{s})$ in the direction $\mathbf{s}$, then

$$I_{\mathbf{s}}(\mathbf{r})=\sigma(\mathbf{r},\mathbf{s})[V(\mathbf{r})-V(\hat{l}_{\mathbf{s}}\mathbf{r})]=-\sigma(\mathbf{r},\mathbf{s})\hat{\rho}_{\mathbf{s}}(\mathbf{r})V(\mathbf{r}) , \tag{2}$$

where $\sigma(\mathbf{r},\mathbf{s})$ is the conductance of the bond $(\mathbf{r},\mathbf{s})$.

Using the notation just introduced, Kirchhoff's laws can be written

$$\sum_{\mathbf{s}}[\sigma(\mathbf{r},\mathbf{s})\hat{\rho}_{\mathbf{s}}(\mathbf{r})+\sigma(\mathbf{r}-\mathbf{s},\mathbf{s})\hat{\rho}_{-\mathbf{s}}(\mathbf{r})]V(\mathbf{r})=0 . \tag{3}$$

Here $\mathbf{r}$ can be any site in the network except a site on the busbars: in other words, $\mathbf{r}\cdot\hat{\mathbf{y}}$ cannot be zero or $(N_y-1)a$. Equation (3), together with the boundary conditions (1), uniquely determines the voltage $V(\mathbf{r})$. The solution to this system of linear equations is also the minimum of the function f subject to the constraints given in Eq. (1), where

$$f\equiv\tfrac{1}{2}\sum_{\mathbf{r}}\sum_{\mathbf{s}}\{\sigma(\mathbf{r},\mathbf{s})[\hat{\rho}_{\mathbf{s}}(\mathbf{r})V(\mathbf{r})]^2 + \sigma(\mathbf{r}-\mathbf{s},\mathbf{s})[\hat{\rho}_{-\mathbf{s}}(\mathbf{r})V(\mathbf{r})]^2\} \tag{4}$$

and the summation $\sum_{\mathbf{r}}$ runs over all the sites in the network. To solve Kirchhoff's laws numerically, we could start with Eq. (3) and use over-relaxation, for example. Alternatively, we could begin with Eq. (4) and use the conjugate-gradient method to find the minimum of the function f. In both Eqs. (3) and (4), only nearest-neighbor pairs of sites are directly coupled. This reduces the speed of convergence in both the over-relaxation method and the conjugate-gradient method. For example, in the over-relaxation method, a change in the value of $V(\mathbf{r})$ will propagate to the site $\mathbf{r}'$ only after $(|x-x'|+|y-y'|)/a$ iterations. To overcome this problem, we reformulate Kirchhoff's laws using Green's functions.

We begin by dividing $V(\mathbf{r})$ into two parts: we set $V(\mathbf{r})=V^0(\mathbf{r})+V'(\mathbf{r})$. We shall take $V^0(\mathbf{r})$ to be the solution of the network in which each resistor has conductance σ_+, and which is subject to the boundary conditions (1). Therefore, $V^0(\mathbf{r})=V_0 y/[(N_y-1)a]$. The function $V'(\mathbf{r})$ is subject to the boundary conditions

$$V'(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{y}}=0}=V'(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a}=0$$

and

$$V'(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{x}}=0}=V'(\mathbf{r})|_{\mathbf{r}\cdot\hat{\mathbf{x}}=N_x a} .$$

We will show that $V'(\mathbf{r})$ can be expressed in terms of a set of Green's functions $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$. Our first step in this process will be to define and construct these Green's functions.

Consider a network in which all the resistors have conductance σ_+. We impose periodic boundary conditions along the x direction, and set the voltage to zero at all points on the two busbars. For convenience, we connect each pair of points $(n_x a,0)$ and $(n_x a,(N_y-1)a)$ with a resistor with unit conductance, where $n_x=0,1,\ldots,N_x-1$. Obviously, the presence of these extra resistors will not affect the voltages in the network once the boundary conditions are specified. We now inject a current σ_+ into the network at $\mathbf{r}_0=(x_0,y_0)$, and draw the same amount of current out at the point $\mathbf{r}_0+\mathbf{s}_0$. Here $\mathbf{r}_0$ and $\mathbf{s}_0$ are arbitrary. The Green's function $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ is defined to be the voltage at the point $\mathbf{r}$ due to this dipole current source. The current is conserved at all points except $\mathbf{r}_0$, $\mathbf{r}_0+\mathbf{s}_0$, and the points on the busbars. Using this fact, we obtain

$$\sum_{\mathbf{s}}[\hat{\rho}_{\mathbf{s}}(\mathbf{r})+\hat{\rho}_{-\mathbf{s}}(\mathbf{r})]U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)=\hat{\rho}_{-\mathbf{s}_0}(\mathbf{r})\delta(\mathbf{r},\mathbf{r}_0) . \tag{5}$$

Here $\mathbf{r}$ can be any point in the network except a point on the busbars, where $\mathbf{r}\cdot\hat{\mathbf{y}}=0$ or $\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a$. The value of the function U on the busbars is determined by the boundary conditions

$$U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=0}=U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a}=0\ , \tag{6a}$$

and

$$U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=0}=U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=N_x a}\ . \tag{6b}$$

It is useful to define a second Green's function, $G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$. We let $G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ be the current passing through the bond $(\mathbf{r},\mathbf{s})$ in the direction $\mathbf{s}$ due to the current dipole source located on the bond $(\mathbf{r}_0,\mathbf{s}_0)$. Clearly, the relation between the Green's functions $G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ and $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ is

$$G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)=-\hat{\rho}_{\mathbf{s}}(\mathbf{r})U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)\ . \tag{7}$$

G must satisfy the boundary conditions

$$G_{\mathbf{s}_x\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a}=G_{\mathbf{s}_y\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=(N_y-1)a}=0\ , \tag{8a}$$

and

$$G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=0}=G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=N_x a}\ . \tag{8b}$$

The boundary conditions (6a) make the solution of Eq. (5) somewhat complicated. It is simplest if we first construct the Green's function $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$, which is the solution of

$$\sum_{\mathbf{s}}[\hat{\rho}_{\mathbf{s}}(\mathbf{r})+\hat{\rho}_{-\mathbf{s}}(\mathbf{r})]u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)=\hat{\rho}_{-\mathbf{s}_0}(\mathbf{r})\delta(\mathbf{r},\mathbf{r}_0) \tag{9}$$

subject to periodic boundary conditions along both the x and y directions. Explicitly,

$$u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=0}=u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{x}}=N_x a} \tag{10a}$$

and

$$u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=0}=u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)|_{\mathbf{r}\cdot\hat{\mathbf{y}}=N_y a}\ . \tag{10b}$$

The boundary condition (10b) can be realized by connecting the points $(n_x a,0)$ and $(n_x a,(N_y-1)a)$ by a resistor with conductance σ_+, where $n_x=0,1,\ldots,N_x-1$. In Eq. (9), $\mathbf{r}$ can be any point in the network, and $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ is the voltage at the point $\mathbf{r}$ due to the dipole current source located at $(\mathbf{r}_0,\mathbf{s}_0)$ when periodic boundary conditions apply in both the x and y directions.

As previously, we define the Green's function $g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ to be

$$g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)=-\hat{\rho}_{\mathbf{s}}(\mathbf{r})u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)\ . \tag{11}$$

The Green's function $g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ is the current passing through the bond $(\mathbf{r},\mathbf{s})$ in the direction $\mathbf{s}$ due to the dipole current source located at the bond $(\mathbf{r}_0,\mathbf{s}_0)$.

Equations (9) and (10) are easily solved using Fourier transforms. The solution is

$$u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)=\tfrac{1}{4}\sum_{\mathbf{k}}\frac{1-\exp(-i\mathbf{k}\cdot\mathbf{s}_0)}{\sin^2(k_x a/2)+\sin^2(k_y a/2)}\times\exp[i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_0)]\ , \tag{12}$$

where $k_x=2\pi n_x/N_x a$, $k_y=2\pi n_y/N_y a$, $n_x=0,1,\ldots,N_x-1$, and $n_y=0,1,\ldots,N_y-1$. Using the definition of g [Eq. (11)] and Eq. (12), we obtain

$$g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)=\tfrac{1}{4}\sum_{\mathbf{k}}\frac{[1-\exp(-i\mathbf{k}\cdot\mathbf{s}_0)][1-\exp(i\mathbf{k}\cdot\mathbf{s})]}{\sin^2(k_x a/2)+\sin^2(k_y a/2)}\times\exp[i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}_0)]\ . \tag{13}$$

Equations (12) and (13) give us the Green's functions for the voltage and current with periodic boundary conditions in both the x and y directions. Both $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ and $g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ depend on $\mathbf{r}-\mathbf{r}_0$ but not on $\mathbf{r}+\mathbf{r}_0$. This is a result of the translational symmetry of the network. Later we will use this feature to reduce the number of operations needed in the numerical calculations. The Fourier transform of $g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ is

$$g_{\mathbf{s}\mathbf{s}_0}(\mathbf{k})\equiv\frac{1}{N_x N_y}\sum_{\mathbf{r}}g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},0)\exp(-i\mathbf{k}\cdot\mathbf{r})\ .$$

Using Eq. (13), we obtain

$$g_{\mathbf{s}\mathbf{s}_0}(\mathbf{k})=\frac{1}{4}\frac{[1-\exp(-i\mathbf{k}\cdot\mathbf{s}_0)][1-\exp(i\mathbf{k}\cdot\mathbf{s})]}{\sin^2(k_x a/2)+\sin^2(k_y a/2)}\ . \tag{14}$$

We shall now construct the Green's functions U and G using u and g. Notice that the Green's function $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ is a solution of Eq. (5), but does not satisfy the boundary condition (6a). However, any linear combination of the function $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ with the functions $u(\mathbf{r},\mathbf{r}',\mathbf{s}')$ will be a solution of Eq. (5) if $\mathbf{r}'\cdot\hat{\mathbf{y}}=(N_y-1)a$. To obtain $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$, we form a linear combination of the function $u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ and the functions $u(\mathbf{r},\mathbf{r}',\mathbf{s}')$ in such a way that Eq. (6a) is satisfied. In other words, we inject current into and draw current from the sites in the top and bottom rows of the network in such a way that the currents passing through the bonds $(n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}},\mathbf{s}')$ are zero, where $n_x=0,1,\ldots,N_x-1$ and $\mathbf{s}'$ can take on the values $\mathbf{s}_x$ and $\mathbf{s}_y$. The Green's function $G_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ also can be obtained from a linear combination of the function $g_{\mathbf{s}\mathbf{s}_0}(\mathbf{r},\mathbf{r}_0)$ and the functions $g_{\mathbf{s}\mathbf{s}'}(\mathbf{r},\mathbf{r}')$. This combination has the same form as for $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$. Explicitly,

$$U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)=u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)+\sum_{\mathbf{s}'}\sum_{n_x=0}^{N_x-1}t_{\mathbf{s}'\mathbf{s}_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)\times u(\mathbf{r},n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}},\mathbf{s}') \tag{15}$$

and

$$G_{ss_0}(\mathbf{r},\mathbf{r}_0)=g_{ss_0}(\mathbf{r},\mathbf{r}_0)+\sum_{s'}\sum_{n_x=0}^{N_x-1}t_{s's_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)\times g_{ss'}(\mathbf{r},n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}}) , \tag{16}$$

where the coefficients $t_{s's_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)$ are determined by Eq. (8a). We substitute Eq. (16) into Eq. (8a) and obtain

$$g_{ss_0}(n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}},\mathbf{r}_0)+\sum_{s'}\sum_{n_x'=0}^{N_x-1}t_{s's_0}(n_x' a\hat{\mathbf{x}},\mathbf{r}_0)g_{ss'}((n_x-n_x')a\hat{\mathbf{x}},0)=0 , \tag{17}$$

where $n_x=0,1,\ldots,N_x-1$. We define $\tilde{t}_{ss_0}(k_x,\mathbf{r}_0)$ as follows:

$$\tilde{t}_{ss_0}(k_x,\mathbf{r}_0)=\frac{1}{N_x}\sum_{n_x=0}^{N_x-1}t_{ss_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)\exp(-ik_x n_x a) . \tag{18}$$

In Fourier space, Eq. (17) becomes

$$T_{ss_0}(k_x,\mathbf{r}_0)+N_x\sum_{s'}\tilde{t}_{s's_0}(k_x,\mathbf{r}_0)T^0_{ss'}(k_x)=0 , \tag{19}$$

where

$$T_{ss_0}(k_x,\mathbf{r}_0)\equiv\sum_{k_y}g_{ss_0}(\mathbf{k})\exp(-i\mathbf{k}\cdot\mathbf{r}_0)\exp[ik_y(N_y-1)a]$$

and

$$T^0_{ss_0}(k_x)\equiv T_{ss_0}(k_x,(N_y-1)a\hat{\mathbf{y}}) .$$

The solution of Eq. (19) is

$$\tilde{t}_{ss_0}(k_x,\mathbf{r}_0)=-\frac{1}{N_x}\frac{T^0_{s^*s^*}(k_x)T_{ss_0}(k_x,\mathbf{r}_0)-T^0_{s_xs_y}(k_x)T_{s^*s_0}(k_x,\mathbf{r}_0)}{T^0_{s_xs_x}(k_x)T^0_{s_ys_y}(k_x)-T^0_{s_xs_y}(k_x)T^0_{s_xs_y}(k_x)} , \tag{20}$$

where

$$s^*=\begin{cases}s_x, & \text{when } s=s_y\\ s_y, & \text{when } s=s_x\end{cases} .$$

Because of the translational symmetry of the network in the x direction, the function $t_{ss_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)$ can be written $t_{ss_0}(0,\mathbf{r}_0-n_x a\hat{\mathbf{x}})$. We will use this feature of $t_{ss_0}(n_x a\hat{\mathbf{x}},\mathbf{r}_0)$ in the numerical calculations. Using Eqs. (15), (16), and (18), we obtain

$$U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)=u(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)+\sum_{s'}\sum_{n_x,l=0}^{N_x-1}\tilde{t}_{s's_0}\left[\frac{2\pi}{N_x a}l,\mathbf{r}_0\right]u(\mathbf{r},n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}},\mathbf{s}')\exp\left[i\frac{2\pi}{N_x}n_x l\right] \tag{21}$$

and

$$G_{ss_0}(\mathbf{r},\mathbf{r}_0)=g_{ss_0}(\mathbf{r},\mathbf{r}_0)+\sum_{s'}\sum_{n_x,l=0}^{N_x-1}\tilde{t}_{s's_0}\left[\frac{2\pi}{N_x a}l,\mathbf{r}_0\right]g_{ss'}(\mathbf{r},n_x a\hat{\mathbf{x}}+(N_y-1)a\hat{\mathbf{y}})\exp\left[i\frac{2\pi}{N_x}n_x l\right] . \tag{22}$$

Together, Eqs. (20)–(22) give explicit solutions for the Green's functions U and G [33].

Using Eqs. (20) and (22), one can prove that

$$G_{ss_0}(\mathbf{r},\mathbf{r}_0)=G_{s_0s}(\mathbf{r}_0,\mathbf{r}) . \tag{23}$$

This symmetry will also be used in the numerical calculations. The value of $G_{s_is_j}(\mathbf{r}_i,\mathbf{r}_j)$ depends on both $\mathbf{r}_i$ and $\mathbf{r}_j$: it does not only depend on $\mathbf{r}_i-\mathbf{r}_j$ alone. In this respect, $G_{s_is_j}(\mathbf{r}_i,\mathbf{r}_j)$ differs from $g_{s_is_j}(\mathbf{r}_i,\mathbf{r}_j)$. However, there is translational symmetry in the x direction, i.e., $G_{s_is_j}(\mathbf{r}_i,\mathbf{r}_j)$ does not depend on x_i+x_j. This symmetry will be used to reduce the number of operations needed in the numerical calculations.

So far, we have constructed the Green's function $G_{ss_0}(\mathbf{r},\mathbf{r}_0)$ for a network of identical resistors of conductance σ_+. We are now ready to deal with the case in which some bonds have conductance σ_- and the remainder have conductance σ_+. We first consider the simplest case, in which all the bonds have conductance σ_+ except one bond located at $(\mathbf{r}_0,\mathbf{s}_0)$ with conductance σ_-. We apply a voltage V_0 across the busbars so that the boundary conditions are given by Eq. (1). The equation determining the voltage $V(\mathbf{r})$ in the network is

$$\sum_s[\hat{\rho}_s(\mathbf{r})+\hat{\rho}_{-s}(\mathbf{r})]V(\mathbf{r})=\Omega^{-1}[\delta(\mathbf{r},\mathbf{r}_0)\hat{\rho}_{s_0}(\mathbf{r})+\delta(\mathbf{r},\hat{l}_{s_0}\mathbf{r}_0)\hat{\rho}_{-s_0}(\mathbf{r})]V(\mathbf{r}) , \tag{24}$$

where $\Omega\equiv(1-h)^{-1}$. We assume that

$$V(\mathbf{r})=V^0(\mathbf{r})+\epsilon U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0) ,$$

where the coefficient ϵ is to be determined. Substituting

$V(\mathbf{r})$ into Eq. (24) and using Eqs. (5) and (7), we obtain

$$\Omega\epsilon+G_{\mathbf{s}_0\mathbf{s}_0}(\mathbf{r}_0,\mathbf{r}_0)\epsilon=-v_0\mathbf{s}_0\cdot\hat{\mathbf{y}} , \tag{25}$$

where $v_0\equiv V_0/(N_y-1)$. This equation yields the value of the coefficient ϵ. With this choice of ϵ, current is conserved and the boundary conditions (1) are satisfied. $V(\mathbf{r})$ is thus the correct solution for the voltage.

Because $V(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$ is a superposition of the functions $V^0(\mathbf{r})$ and $U(\mathbf{r},\mathbf{r}_0,\mathbf{s}_0)$, Eq. (24) holds for all $\mathbf{r}$ except $\mathbf{r}=\mathbf{r}_0$ and $\mathbf{r}=\mathbf{r}_0+\mathbf{s}_0$ for arbitrary choices of ϵ. This means that the current is conserved at all the sites in the network except these two sites for any value of ϵ. If the value of ϵ given in Eq. (25) is not chosen, the current leak at the point $\mathbf{r}_0$ is $\Delta I(\epsilon,\mathbf{r}_0)$, where

$$\Delta I(\epsilon,\mathbf{r}_0)=(\sigma_+-\sigma_-)[\Omega\epsilon+G_{\mathbf{s}_0\mathbf{s}_0}(\mathbf{r}_0,\mathbf{r}_0)\epsilon+v_0\mathbf{s}_0\cdot\hat{\mathbf{y}}] .$$

Similarly, the current leak at $\mathbf{r}_0+\mathbf{s}_0$ is $-\Delta I(\epsilon,\mathbf{r}_0)$.

We now turn to the general case, in which there are n bonds with conductance σ_-. The ith of these bonds will be labeled $(\mathbf{r}_i,\mathbf{s}_i)$, where $i=1,2,\ldots,n$. The rest of the bonds have conductance σ_+. The equation for the potential function $V(\mathbf{r})$ is

$$\sum_{\mathbf{s}}[\hat{\rho}_{\mathbf{s}}(\mathbf{r})+\hat{\rho}_{-\mathbf{s}}(\mathbf{r})]V(\mathbf{r})$$
$$=\Omega^{-1}\left\{\sum_{i=1}^{n}[\delta(\mathbf{r},\mathbf{r}_i)\hat{\rho}_{\mathbf{s}_i}(\mathbf{r})+\delta(\mathbf{r},\hat{l}_{\mathbf{s}_i}\mathbf{r}_i)\hat{\rho}_{-\mathbf{s}_i}(\mathbf{r})]\right\}V(\mathbf{r}) . \tag{26}$$

As for the network with a single bond of conductance σ_-, we assume that $V(\mathbf{r})$ may be written

$$V(\mathbf{r})=V^0(\mathbf{r})+\sum_{i=1}^{n}\epsilon_i U(\mathbf{r},\mathbf{r}_i,\mathbf{s}_i) . \tag{27}$$

We will show shortly that such a representation of the potential always exists. Substituting this equation into Eq. (26), we find that the ϵ_i's satisfy the linear system

$$\Omega\epsilon_i+\sum_{j=1}^{n}G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)\epsilon_j=-v_0\mathbf{s}_i\cdot\hat{\mathbf{y}} , \tag{28}$$

where $i=1,2,\ldots,n$. When $n=1$, Eq. (28) reduces to Eq. (25). If we find a set of ϵ_i's that satisfy Eq. (28), then $V(\mathbf{r})$ will be the correct solution for the voltage. For an arbitrary set of ϵ_i's, the current leak $\Delta I(\{\epsilon_i\},\mathbf{r}_j)$ at the point $\mathbf{r}_j$ is

$$\Delta I(\{\epsilon_i\},\mathbf{r}_j)=(\sigma_+-\sigma_-)\left[\Omega\epsilon_j+\sum_{k=1}^{n}G_{\mathbf{s}_j\mathbf{s}_k}(\mathbf{r}_j,\mathbf{r}_k)\epsilon_k+v_0\mathbf{s}_j\cdot\hat{\mathbf{y}}\right] . \tag{29}$$

In numerical calculations, once the values of the ϵ_i's have been obtained, the potential at any point in the network can be calculated using Eq. (27).

Equation (28) is the GFF linear system that we set out to derive. To understand its physical import, we begin by examining Eq. (27). $V^0(\mathbf{r})$ is the voltage distribution generated by the applied external voltage when all bonds have conductance σ_+. The Green's function $U(\mathbf{r},\mathbf{r}_i,\mathbf{s}_i)$ is the voltage distribution created by injecting a unit current at $\mathbf{r}_i$ and drawing the same current out at $\mathbf{r}_i+\mathbf{s}_i$. This is a Green's function for the uniform network in which all resistors have conductance σ_+, and there is no applied voltage. Thus, in Eq. (27), the voltage is decomposed into a portion coming from the applied external voltage, and a linear superposition of voltages arising from unit dipole current sources. The effect of the bonds with conductance σ_- is mimicked by replacing them by resistors with conductance σ_+ in parallel with dipole current sources.

Why does a representation of the potential of the form (27) always exist? For simplicity, consider the case $n=1$, so that there is a single resistor with conductance σ_-. This resistor can be replaced by a resistor with conductance σ_+ in parallel with a current source. The current source causes a fixed current $\sigma_+\epsilon$ to flow in the direction opposite to the current through the resistor. When ϵ is chosen appropriately, this current compensates for the increased current that flows through the resistor as a result of its increased conductance. Thus the conductance σ_- can be replaced by a conductance σ_+ and a dipole current source. Clearly this discussion can be generalized to arbitrary values of n. The GFF linear system (28) specifies the appropriate strengths for the dipole current sources in this case.

We have developed the GFF for the simplest case in which there are only two types of resistors. However, our treatment is readily generalized to networks with an arbitrary number of resistor types. Suppose that there are n bonds with conductances different from σ_+. The ith of these bonds has conductance σ_i and will be labeled $(\mathbf{r}_i,\mathbf{s}_i)$, where $i=1,2,\ldots,n$. The rest of the bonds have conductance σ_+. The only modification to Eq. (28) that is necessary is that Ω must be replaced by Ω_i, where $\Omega_i\equiv(1-\sigma_i/\sigma_+)^{-1}$.

In contrast to Kirchhoff's laws [Eq. (3)], the couplings between sites are not restricted to nearest-neighbor pairs in Eq. (28). Instead, ϵ_i and ϵ_j are coupled through the Green's function $G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)$. In an iterative solution of Eq. (28), the change in ϵ_i at the nth iteration will propagate to all the other ϵ_j's at the next [$(n+1)$th] iteration. The changes in the other ϵ_j's due to the change in ϵ_i are determined by the Green's function $G_{\mathbf{s}_j\mathbf{s}_i}(\mathbf{r}_j,\mathbf{r}_i)$. Because of this long-range coupling, we expect the speed of convergence in an iterative calculation of the ϵ_i's using Eq. (28) to be much higher than in an iterative calculation of $V(\mathbf{r})$ using Kirchhoff's laws directly. This expectation is borne out by the numerical work described in Sec. III.

III. SOLUTION OF THE GFF LINEAR SYSTEM USING THE CONJUGATE-GRADIENT METHOD

The GFF linear system (28) can be solved using a variety of numerical methods. In this section, we will use the conjugate-gradient method to solve Eq. (28) numeri-

cally. We call this method A. For comparison, we will also solve Kirchhoff's equations directly using the conjugate-gradient method (method B). In our numerical calculations, the boundary conditions are given by Eq. (1), and we apply a voltage difference $V_0=N_y$ between the two busbars. The conductance of each resistor in the network is either σ_- with probability p, or $\sigma_+=1$ with probability $1-p$. We will first give the details of implementing method A, and then show that it is usually much faster than method B.

Because the number of variables in Eq. (3) is $N_x(N_y-2)$, it is impractical to solve Eq. (3) directly by computing the inverse matrix for all but the smallest of networks. On the other hand, the number of variables in Eq. (28) is n, the number of resistors with conductance σ_-. To solve Eq. (28) directly, we must calculate the inverse matrix of a $n\times n$ matrix. This calculation requires $O(n^3)$ operations [34]. For small n, this scheme will be very efficient. When n is close to the total number of the resistors in the network b, we can still solve Eq. (28) directly by making the following changes in Eq. (28): first, we replace the summation over all the resistors with conductance σ_- by a summation over all the resistors with conductance σ_+, and then we replace Ω by $\Omega'=(1-h^{-1})^{-1}$. After these replacements, there will be only $b-n$ variables in the new form of Eq. (28). It requires $O((b-n)^3)$ operations to calculate the inverse matrix of this $(b-n)\times(b-n)$ matrix, and when n is close to b, this is practicable.

When both n and $b-n$ are large, it is inefficient to solve Eq. (28) directly. In this case, other methods should be used. Using the symmetry property (23) of the Green's function $G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)$, it can be shown readily that the solution of Eq. (28) is the minimum of the function F, where

$$F\equiv\frac{1}{2}\sum_{i=1}^{n}\Omega\epsilon_i^2+\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)\epsilon_i\epsilon_j+v_0\sum_{i=1}^{n}\mathbf{s}_i\cdot\hat{\mathbf{y}}\epsilon_i\ . \tag{30}$$

We shall employ the conjugate-gradient method to find the minimum of the function F.

To calculate the function F directly for a given set of ϵ_i's, we must perform $O(n^2)$ operations. As long as the number of iterations needed to reach the minimum of the function F is less than $O(n)$, using Eq. (30) to calculate the ϵ_i's will be more efficient than inverting Eq. (28) directly. However, the number of operations will still be large for very large N. Actually, as we shall see, we can calculate the function F in Fourier space using fast Fourier transforms (FFT's). This allows us to compute F with $O(N_xN_y\log_2(N_xN_y))$ operations [35] for a given set of ϵ_i's. Typically, n is on the order of pN_xN_y, and so this will be more efficient than computing F directly. Accordingly, for large n, we shall calculate the function F in Fourier space instead of real space.

We begin by explaining how the conjugate-gradient method is used to solve the GFF linear system (for a general discussion of the conjugate-gradient method, see Ref. [26]). At step 0, we choose some initial set of $\epsilon_i^{(0)}$'s, and calculate the derivative $q_i^{(0)}$ of the function F with respect to ϵ_i at this point for $i=1,2,\ldots,n$. The $q_i^{(0)}$'s are obtained from the relation

$$q_i^{(0)}=\Omega\epsilon_i^{(0)}+\sum_{j=1}^{n}G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)\epsilon_j^{(0)}+v_0\mathbf{s}_i\cdot\hat{\mathbf{y}}\ .$$

We also put $Q_i^{(0)}=q_i^{(0)}$ for $i=1,2,\ldots,n$. Now consider the situation at the mth step. The $\epsilon_i^{(m)}$'s, $q_i^{(m)}$'s, and $Q_i^{(m)}$'s have been computed at this stage. We now calculate

$$P_i^{(m)}\equiv\sum_{j=1}^{n}G_{\mathbf{s}_i\mathbf{s}_j}(\mathbf{r}_i,\mathbf{r}_j)Q_j^{(m)} \tag{31}$$

for $i=1,2,\ldots,n$ and

$$l^{(m)}\equiv-\left[\sum_{i=1}^{n}Q_i^{(m)}(\Omega Q_i^{(m)}+P_i^{(m)})\right]^{-1}\times\left[\sum_{i=1}^{n}q_i^{(m)}q_i^{(m)}\right]\ . \tag{32}$$

The point $\{\epsilon_i^{(m+1)}\}$ at the $(m+1)$th iteration, and the derivatives of F with respect to the ϵ_i's at this point (the $q_i^{(m+1)}$'s) are found using

$$\epsilon_i^{(m+1)}=\epsilon_i^{(m)}+l^{(m)}Q_i^{(m)} \tag{33}$$

and

$$q_i^{(m+1)}=q_i^{(m)}+l^{(m)}(\Omega Q_i^{(m)}+P_i^{(m)})\ . \tag{34}$$

Finally, the $Q_i^{(m+1)}$'s are computed using the relation

$$Q_i^{(m+1)}=q_i^{(m+1)}+t^{(m)}Q_i^{(m)}\ , \tag{35}$$

where

$$t^{(m)}\equiv\left[\sum_{i=1}^{n}q_i^{(m)}q_i^{(m)}\right]^{-1}\left[\sum_{i=1}^{n}q_i^{(m+1)}q_i^{(m+1)}\right]\ . \tag{36}$$

We repeat this procedure until

$$\frac{1}{n}\left[\sum_{i=1}^{n}(q_i^{(m)})^2\right]^{1/2}\le\Delta\ ,$$

where Δ is the desired precision. Using Eqs. (29) and (30), we see that this condition can be written

$$\frac{1}{n(\sigma_+-\sigma_-)}\left\{\sum_{j=1}^{n}[\Delta I(\{\epsilon_i\},\mathbf{r}_j)]^2\right\}^{1/2}\le\Delta\ . \tag{37}$$

Therefore, the stopping criterion (37) limits the root-mean-square current leak in the network.

Performing the summations in Eqs. (32)–(37) only requires $O(n)$ operations for each iteration. The most time-consuming part of each iteration is to calculate the $P_i^{(m)}$'s defined in Eq. (31). This requires $O(n^2)$ operations if the sums are carried out directly. Provided n is not too large, the $P_i^{(m)}$'s can be calculated using Eq. (31), but for large n a different approach is needed.

In order to reduce the number of operations needed to evaluate the $P_i^{(m)}$'s using Eq. (31), we carry out the calcu-

lation in Fourier space. We divide the resistors with conductance σ_- into two groups. The first group contains all the horizontal resistors, and we use the variable $\mathbf{h}_i$ to specify the location of the ith horizontal resistor. The other group consists of the vertical resistors, and the variable $\mathbf{v}_i$ will be used to specify the location of the ith resistor in this set. We let n_h and n_v be the number of horizontal resistors and vertical resistors with conductance σ_-, respectively. The total number of the resistors with conductance σ_- is $n_h+n_v=n$. We use $Q^{(m)}_{s_x}(\mathbf{h}_i)$ and $P^{(m)}_{s_x}(\mathbf{h}_i)$ to represent the values of $Q_i^{(m)}$ and $P_i^{(m)}$ for a horizontal resistor located at the position $\mathbf{h}_i$. Similarly, $Q^{(m)}_{s_y}(\mathbf{v}_i)$ and $P^{(m)}_{s_y}(\mathbf{v}_i)$ are the values of $Q_i^{(m)}$ and $P_i^{(m)}$ for a vertical resistor located at $\mathbf{v}_i$. We define

$$\tilde{Q}^{(m)}_{s_x}(\mathbf{k})\equiv\frac{1}{N_xN_y}\sum_{i=1}^{n_h}Q^{(m)}_{s_x}(\mathbf{h}_i)\exp(-i\mathbf{k}\cdot\mathbf{h}_i)\ , \tag{38}$$

$$\tilde{Q}^{(m)}_{s_y}(\mathbf{k})\equiv\frac{1}{N_xN_y}\sum_{i=1}^{n_v}Q^{(m)}_{s_y}(\mathbf{v}_i)\exp(-i\mathbf{k}\cdot\mathbf{v}_i)\ , \tag{39}$$

and

$$\tilde{t}_{ss'}(\mathbf{k})\equiv\frac{1}{N_xN_y}\sum_{n_x=0}^{N_x-1}\sum_{n_y=0}^{N_y-1}t_{ss'}(n_xa\hat{\mathbf{x}},n_ya\hat{\mathbf{y}})\times\exp[-i\mathbf{k}\cdot(n_xa\hat{\mathbf{x}}+n_ya\hat{\mathbf{y}})]\ . \tag{40}$$

Using the fact that $t_{ss_0}(n_xa\hat{\mathbf{x}},\mathbf{r}_0)$ can be written $t_{ss_0}(0,\mathbf{r}_0-n_xa\hat{\mathbf{x}})$ and Eqs. (13), (14), (18), and (22), we obtain

$$P^{(m)}_{s_i}(\mathbf{r}_i)=N_xN_y\sum_{\mathbf{k}}\sum_{s'}\tilde{P}^{(m)}_{s}(\mathbf{k})\exp(i\mathbf{k}\cdot\mathbf{r}_i)\ , \tag{41}$$

where

$$\tilde{P}^{(m)}_{s}(\mathbf{k})=g_{s_is'}(\mathbf{k})[\tilde{Q}^{(m)}_{s'}(\mathbf{k})+e^{-i(N_y-1)k_ya}J^{(m)}_{s'}(k_x)]$$

and

$$J^{(m)}_{s'}(k_x)=N_x\sum_{s''}\sum_{k_y}\tilde{t}_{s's''}(\mathbf{k})\tilde{Q}^{(m)}_{s''}(-\mathbf{k})\ . \tag{42}$$

Note that $J^{(m)}_{s}$ depends on k_x alone and that it involves a single sum over k_y. This leads to a tremendous reduction in the number of operations required at each iteration and makes method A practical. This reduction is a direct consequence of the translational symmetry in the x direction. If it were not for this symmetry, it would take $O(N_x^2N_y^2)$ operations to evaluate $\tilde{P}^{(m)}_{s}(\mathbf{k})$, instead of just $O(N_xN_y)$ operations.

To calculate the $P^{(m)}_{s_i}(\mathbf{r}_i)$'s using Eq. (41), we first must calculate $\tilde{Q}^{(m)}_{s_x}(\mathbf{k})$ and $\tilde{Q}^{(m)}_{s_y}(\mathbf{k})$ from Eqs. (38) and (39) using FFT's. Next, we calculate the $J^{(m)}_{s'}(k_x)$'s using Eq. (42). This takes $O(N_xN_y)$ operations. Finally, the $P^{(m)}_{s_i}(\mathbf{r}_i)$'s are calculated from Eq. (41) using another FFT. Each FFT requires $O(N_xN_y\log_2(N_xN_y))$ operations. Therefore, the number of operations required to calculate the $P^{(m)}_{s_i}(\mathbf{r}_i)$'s is of order $N_xN_y\log_2(N_xN_y)$. If we use Eqs. (38)–(41) to calculate the $P^{(m)}_{s_i}(\mathbf{r}_i)$'s, we must perform FFT's of $Q^{(m)}_{s_x}(\mathbf{h}_i)$, $Q^{(m)}_{s_y}(\mathbf{v}_i)$, $P^{(m)}_{s_x}(\mathbf{k})$, and $P^{(m)}_{s_y}(\mathbf{k})$ separately. Four FFT's must therefore be done for each iteration if we use these equations.

Actually, the number of FFT's needed for each iteration can be reduced to two using the fact that $Q^{(m)}_{s_x}(\mathbf{r}_h)$, $Q^{(m)}_{s_y}(\mathbf{r}_v)$, and $P^{(m)}_{s_i}(\mathbf{r}_i)$ are real. To see how this is done, let

$$Q^{(m)}(\mathbf{r})\equiv\sum_{j=1}^{n_h}Q^{(m)}_{s_x}(\mathbf{h}_j)\delta(\mathbf{r},\mathbf{h}_j)+i\sum_{j=1}^{n_v}Q^{(m)}_{s_y}(\mathbf{v}_j)\delta(\mathbf{r},\mathbf{v}_j)\ . \tag{43a}$$

and

$$P^{(m)}(\mathbf{r})\equiv\sum_{j=1}^{n_h}P^{(m)}_{s_x}(\mathbf{h}_j)\delta(\mathbf{r},\mathbf{h}_j)+i\sum_{j=1}^{n_v}P^{(m)}_{s_y}(\mathbf{v}_j)\delta(\mathbf{r},\mathbf{v}_j)\ . \tag{43b}$$

Here $\mathbf{r}$ is an arbitrary point in the network. $\delta(\mathbf{x},\mathbf{y})$ is one if $\mathbf{x}=\mathbf{y}$ and is zero otherwise. Note that

$$P^{(m)}_{s_x}(\mathbf{h}_i)=\mathrm{Re}[P^{(m)}(\mathbf{h}_i)] \tag{44}$$

and

$$P^{(m)}_{s_y}(\mathbf{v}_i)=\mathrm{Im}[P^{(m)}(\mathbf{v}_i)]\ , \tag{45}$$

where $\mathrm{Re}(z)$ is the real part of the variable z and $\mathrm{Im}(z)$is its imaginary part. Similar relations apply to $Q^{(m)}_{s_x}(\mathbf{h}_i)$ and $Q^{(m)}_{s_y}(\mathbf{v}_i)$.

We are now ready to explain how the computation is performed. Let $\tilde{Q}^{(m)}(\mathbf{k})$ and $\tilde{P}^{(m)}(\mathbf{k})$ be the Fourier transforms of $Q^{(m)}(\mathbf{r})$ and $P^{(m)}(\mathbf{r})$, respectively. Once we obtain $\tilde{Q}^{(m)}(\mathbf{k})$ from $Q^{(m)}(\mathbf{r})$ using a FFT, we calculate $\tilde{Q}^{(m)}_{s_x}(\mathbf{k})$ and $\tilde{Q}^{(m)}_{s_y}(\mathbf{k})$ using the following formulas:

$$\mathrm{Re}[\tilde{Q}^{(m)}_{s_x}(\mathbf{k})]=\tfrac{1}{2}\mathrm{Re}[\tilde{Q}^{(m)}(\mathbf{k})+\tilde{Q}^{(m)}(-\mathbf{k})], \tag{46}$$

$$\mathrm{Im}[\tilde{Q}^{(m)}_{s_x}(\mathbf{k})]=\tfrac{1}{2}\mathrm{Im}[\tilde{Q}^{(m)}(\mathbf{k})-\tilde{Q}^{(m)}(-\mathbf{k})]\ , \tag{47}$$

$$\mathrm{Re}[\tilde{Q}^{(m)}_{s_y}(\mathbf{k})]=\tfrac{1}{2}\mathrm{Re}[\tilde{Q}^{(m)}(\mathbf{k})-\tilde{Q}^{(m)}(-\mathbf{k})]\ , \tag{48}$$

and

$$\mathrm{Im}[\tilde{Q}^{(m)}_{s_y}(\mathbf{k})]=\tfrac{1}{2}\mathrm{Im}[\tilde{Q}^{(m)}(\mathbf{k})+\tilde{Q}^{(m)}(-\mathbf{k})]\ . \tag{49}$$

After $J^{(m)}_{s'}(k_x)$ is calculated using Eq. (42), we calculate $\tilde{P}^{(m)}(\mathbf{k})$ using

$$\tilde{P}^{(m)}(\mathbf{k})=\sum_{s'}\{g_{s_xs'}(\mathbf{k})[\tilde{Q}^{(m)}_{s'}(\mathbf{k})+J^{(m)}_{s'}(k_x)]\}+i\sum_{s'}\{g_{s_ys'}(\mathbf{k})[\tilde{Q}^{(m)}_{s'}(\mathbf{k})+J^{(m)}_{s'}(k_x)]\}\ . \tag{50}$$

$P^{(m)}(\mathbf{r})$ is obtained by performing a FFT on $\tilde{P}^{(m)}(\mathbf{k})$. Finally, $P^{(m)}_j$ is computed using the definition of $P^{(m)}(\mathbf{r})$. Thus two FFT's must be performed for each iteration,

one to obtain the function $\tilde{Q}^{(m)}(\mathbf{k})$ and the other to find $P^{(m)}(\mathbf{r})$.

Let us summarize the steps in our algorithm. Both $g_{ss'}(\mathbf{k})$ and $\tilde{t}_{ss'}(\mathbf{k})$ are calculated before the simulation begins using Eqs. (14), (20), and (40). The main task in the conjugate-gradient method is to calculate the $P_i^{(m)}$'s defined in Eq. (31). In order to reduce the number of operations in the calculation of the $P_i^{(m)}$'s, we use the following procedure: After we obtain $Q_i^{(m)}$ using Eqs. (34)–(36), we calculate the function $Q^{(m)}(\mathbf{r})$ using Eq. (43a). The functions $\tilde{Q}^{(m)}(\mathbf{k})$ is obtained by performing a FFT on $Q^{(m)}(\mathbf{r})$. Next, $\tilde{Q}_{s'}^{(m)}(\mathbf{k})$ is calculated from $\tilde{Q}^{(m)}(\mathbf{k})$ using Eqs. (46)–(49). We then calculate $J_{s'}^{(m)}(k_x)$ and $\tilde{P}^{(m)}(\mathbf{k})$ using Eqs. (42) and (50), and perform a FFT on the function $\tilde{P}^{(m)}(\mathbf{k})$ to obtain $P^{(m)}(\mathbf{r})$. Finally, we use Eqs. (44) and (45) to calculate the $P_i^{(m)}$'s. All of these steps require $O(N_xN_y)$ operations except the two FFT's, which require $O(N_xN_y\log_2(N_xN_y))$ operations. In contrast, if Eq. (31) is used to compute the $P_i^{(m)}$'s directly, $O(p^2N_x^2N_y^2)$ operations are needed.

In order to sensibly compare methods A and B, we must adopt the same stopping criterion in both methods. The criterion we use limits the current leak on each site in the network. Let ΔI_i be the current leak at the site i. The stopping criterion is

$$\left[\sum_i (\Delta I_i)^2\right]^{1/2} \leq \delta ,$$

where the sum runs over all the sites in the network, and δ is the desired precision. This criterion is essentially the same as Eq. (37). We required a precision of $\delta = 10^{-10}$ in all of our calculations. This precision is much higher than that typically used in simulations. Since the behavior of the convergence is different for $\sigma_- = 0$ and $\sigma_- > 0$, we will discuss these two cases separately.

Let $N_A(N,p,\sigma_-)$ and $N_B(N,p,\sigma_-)$ be the average number of iterations required to reach the desired precision using methods A and B, respectively, for a given set of values of the lattice size $N \equiv N_xN_y$, p, and σ_-. In the numerical calculations, we computed $N_A(N,p,\sigma_-)$ and $N_B(N,p,\sigma_-)$ for the lattice sizes 16×16, 32×32, 64×64, and 128×128, and averaged over 200, 100, 50, and 10 configurations, respectively. For each lattice size and value of σ_-, we calculated $N_A(N,p,\sigma_-)$ and $N_B(N,p,\sigma_-)$ for the values of p ranging from 0.05 to 0.95 in steps of 0.05. In order to compare the overall performance of methods A and B, we define $M(N,\sigma_-)$ to be

$$M(N,\sigma_-) = \frac{\bar{N}_B(N,\sigma_-)}{\bar{N}_A(N,\sigma_-)} ,$$

where

$$\bar{N}_A(N,\sigma_-) \equiv \frac{1}{p_0}\int_0^{p_0} N_A(N,p,\sigma_-)dp$$

and

$$\bar{N}_B(N,\sigma_-) \equiv \frac{1}{p_0}\int_0^{p_0} N_B(N,p,\sigma_-)dp .$$

Here p_0 takes on the value 1 for $\sigma_- > 0$, and is $\frac{1}{2}$ for $\sigma_- = 0$. $\bar{N}_A(N,\sigma_-)$ is the number of iterations required to reach the desired precision using method A averaged over the interval $p \in (0,p_0)$, and $\bar{N}_B(N,\sigma_-)$ is defined similarly.

First consider the case $\sigma_- = 0$. Figures 1 and 2 show N_A and N_V vs p for several different lattice sizes. Both N_A and N_B grow larger as p increases, and reach a maximum at $p = 0.5$. This increase as the critical point is approached is a result of critical slowing down.

Critical slowing down occurs because as p approaches p_c, the geometry of the conducting clusters becomes increasingly more complex. Method A does not contain information about the fractal geometry of the conducting clusters, and so it suffers from critical slowing down. If we combined the multigrid method [31] with the GFF, critical slowing down would presumably be all but eliminated. Alternatively, for p close to p_c, the number of iterations could be reduced by deleting isolated clusters of conducting bonds.

Figure 3 shows the ratio N_B/N_A for several different lattices sizes. For all the values of p and N we studied, N_B/N_A was greater than 1. Thus method A requires fewer iterations than method B to reach the specified precision. Furthermore, even though method A does suffer from critical slowing down, its effect is less dramatic than in method B. For example, for $N = 128\times128$ and $p = \frac{1}{2}$, method B requires about five times more iterations than method A.

A log-log plot of the ratio $M(N,0)$ vs the size of the lattice N (Fig. 4) shows that $M(N,0)$ is an increasing function of the lattice size N, and that $M(N,0)$ reaches the value 7.12 when $N = 128\times128$. From Fig. 4, we see that $\log_{10}M(N,0)$ is approximately linear in $\log_{10}N$ and hence $M(N,0) \approx cN^x$ for large N. This allows us to use the exponent x to compare the performance of methods A and B for large N. A linear least-squares fit to the data in Fig. 4 yields $x = 0.253\pm0.003$ and $c = 0.61\pm0.01$. Since x is

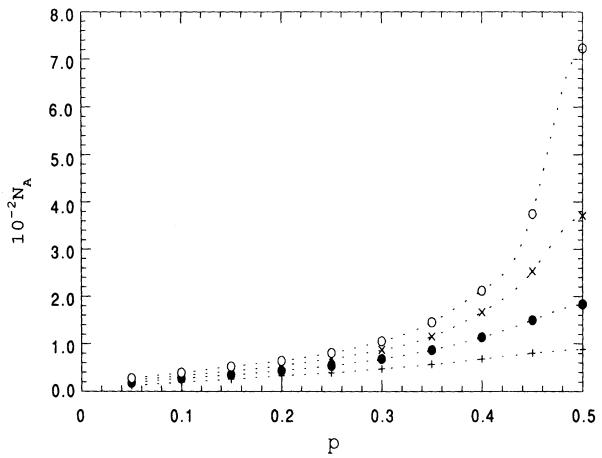

FIG. 1. The number of iterations required to reach the precision $\delta = 10^{-10}$ in method A (N_A) is plotted vs p for several different lattice sizes. In each case $\sigma_- = 0$. The dashed lines are merely guides to the eye. The lattice sizes are 16×16 (+), 32×32 (●), 64×64 (×), and 128×128 (○).

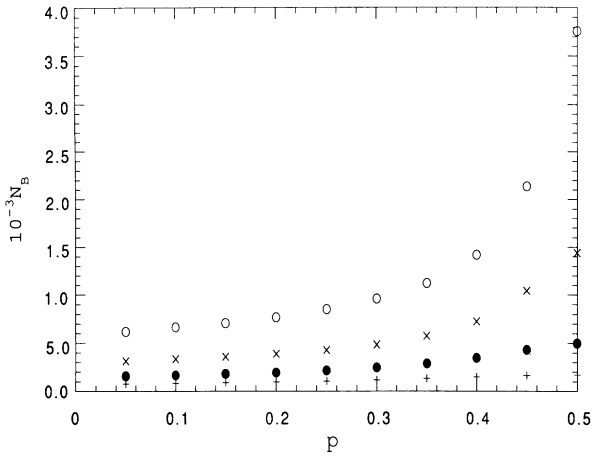

FIG. 2. The number of iterations required to reach the precision $\delta=10^{-10}$ in method B (N_B) is plotted vs p for several different lattice sizes. In each case $\sigma_-=0$. N_B was calculated for the same set of the configurations as used in Fig. 1. The lattice sizes are 16×16 (+), 32×32 (●), 64×64 (×), 128×128 (○).

greater than zero, the bigger the lattice size is, the more we gain using method A.

We now turn to the case $\sigma_->0$. Our numerical results show that as σ_- approaches zero, the value of N_B increases rapidly. When σ_- is small but nonzero, the number of iterations needed to attain the desired precision is much larger than when $\sigma_-=0$. This can be seen in Fig. 5, which shows N_B for a lattice of size 128×128, and for several values of σ_-. This figure is to be compared with the data for the 128×128 lattice with $\sigma_-=0$ shown in Fig. 2.

Let us contrast this with the efficiency of method A for small nonzero σ_-. The value of Ω in Eq. (30) tends to 1 as σ_- approaches zero. Therefore, as σ_- approaches zero, N_A converges to its value for $\sigma_-=0$. Figure 6 shows N_A vs p for several different values of σ_- for the lattice size 128×128. The curves for $\sigma_-=10^{-2}$, 10^{-3},

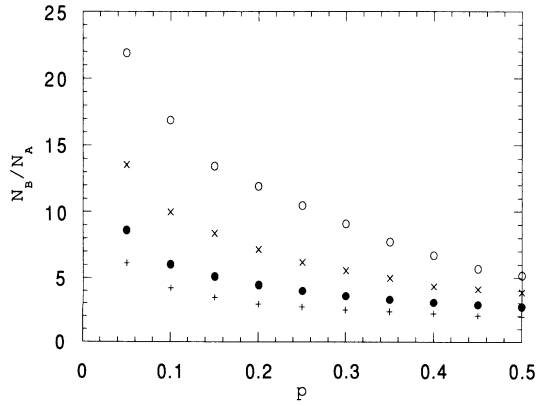

FIG. 3. The ratio N_B/N_A as a function of p for the lattice sizes 16×16 (+), 32×32 (●), 64×64 (×), and 128×128 (○). In each case, $\sigma_-=0$.

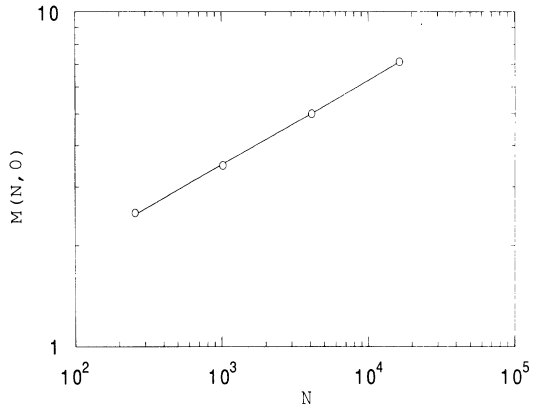

FIG. 4. A log-log plot of $M(N,0)$ vs the lattice size N. The solid line is a linear least-squares fit to the data.

and 10^{-4} are successively closer to the curve for $\sigma_-=0$. Thus method A requires far fewer iterations than method B to reach the desired precision when σ_- is small but nonzero. Figure 7 shows N_B/N_A vs p for several values of σ_- for the 128×128 lattice. In the region $0.5\le p\le1$, the number of iterations needed is dramatically reduced by using method A. For example, for $N=128\times128$, $\sigma_-=10^{-5}$, and $p=0.8$, the value of N_B/N_A exceeds 200.

Figure 8 is a log-log plot of $M(N,\sigma_-)$ vs the lattice size N for several different values of σ_-. Again, $\log_{10}M(N,\sigma_-)$ is approximately linear in $\log_{10}N$, and so $M(N,\sigma_-)\cong cN^x$ for sufficiently large N. Figure 9 shows the values of x for several different values of σ_-. The exponent x appears to have a minimum value of $x_{\min}\cong0.4$; certainly all values of x exceed 0.34. We have seen that method B becomes increasingly inefficient as σ_- approaches zero. To see why this is so, we begin by writing Eq. (4) in matrix form:

$$f=\tfrac{1}{2}\mathbf{v}^T A\mathbf{v}\ , \tag{51}$$

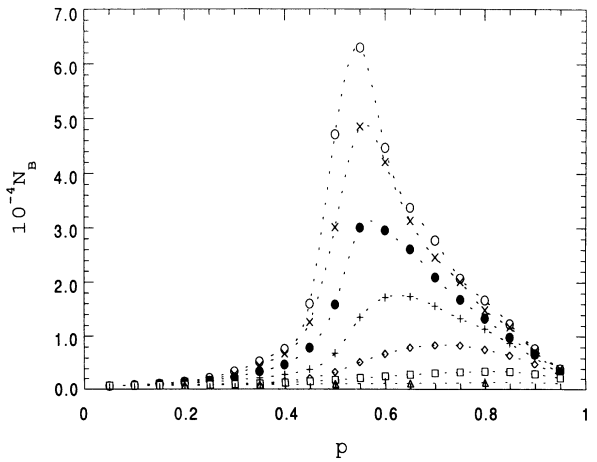

FIG. 5. A plot of N_B vs p for $\sigma_-=10^{-1}$ (△), 10^{-2} (□), 10^{-3} (◇), 10^{-4} (+), 10^{-5} (●), 10^{-6} (×), and 10^{-7} (○). In each case, the lattice size is 128×128. The dashed lines are guides to the eye.

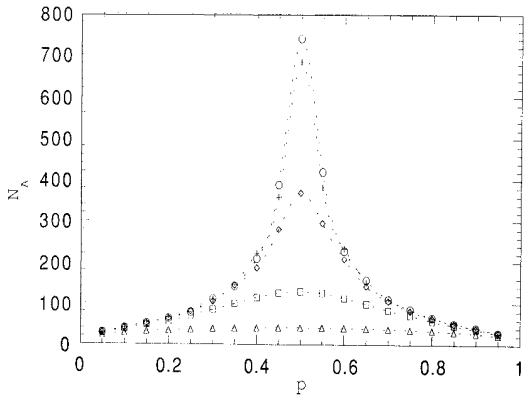

FIG. 6. A plot of N_A vs p for $\sigma_- = 10^{-1}$ (△), 10^{-2} (□), 10^{-3} (◇), 10^{-4} (+), and 0 (○). In each case, the lattice size is 128×128. The dashed lines are guides to the eye.

where $\mathbf{v} \equiv (V(\mathbf{r}_1), V(\mathbf{r}_2), \ldots, V(\mathbf{r}_N))^T$. The $N \times N$ matrix A has components $A(\mathbf{r},\mathbf{r}')$ given by

$$A(\mathbf{r},\mathbf{r}') \equiv \begin{cases} -\frac{1}{2}\sum_{\mathbf{r}''} \sigma(\mathbf{r},\mathbf{r}''), & \text{if } \mathbf{r}=\mathbf{r}' \\ \frac{1}{2}\sigma(\mathbf{r},\mathbf{r}'), & \text{if } \mathbf{r}' \text{ is a nearest neighbor of } \mathbf{r} \\ 0, & \text{otherwise} . \end{cases}$$

Here $\sigma(\mathbf{r},\mathbf{r}')$ is defined to be the conductance of the resistor connecting sites $\mathbf{r}$ and $\mathbf{r}'$, and the $\mathbf{r}''$ are the nearest neighbors of site $\mathbf{r}$.

There are two factors that affect the efficiency of method B when σ_- is small. Let R be the ratio of the largest eigenvalue to the smallest eigenvalue of the matrix A. If R is large, then the convergence to the solution in method B is slow, while if R is close to 1, the convergence is rapid [28]. When h is small, R will be large. As a result, method B converges slowly in this regime.

The second factor that limits the efficiency of method B is rounding errors. In the absence of rounding errors, method B takes at most N iterations to reach the minimum of the function f [27]. The proof of this assertion is based the following fact: Let $\mathbf{v}^{(k)}$ be the value of $\mathbf{v}$ obtained at the kth iteration, and let $\mathbf{d}^{(k)}$ be the derivative of f at that point. $\mathbf{v}^{(k)}$ is the minimum of the function f in the subspace

$$\{\mathbf{v} | \mathbf{v} = \mathbf{v}^{(0)} + c_0\mathbf{d}^{(0)} + c_1 A\mathbf{d}^{(0)} + \cdots + c_k A^k \mathbf{d}^{(0)}\} ,$$

where $c_0, c_1, \ldots, c_k$ are arbitrary real constants. For $k = N$, the subspace becomes the entire space. Thus the number of iterations required to find the solution is less than or equal to N. In a real computation, rounding errors are always present, and method B takes many more than N iterations to reach the minimum of f. This shows that method B is very sensitive to the effect of rounding errors. The efficiency of method B depends a great deal on how precisely the vectors $\mathbf{d}^{(0)}$, $A\mathbf{d}^{(0)}, \ldots, A^k\mathbf{d}^{(0)}$ are calculated. When h is small, the relative precision achieved in calculating these vectors is low. Moreover, the value of $\mathbf{v}^{(i)}$ is used in computing $\mathbf{v}^{(i+1)}$. As a result, rounding errors are compounded during the course of the computation.

So far, we have seen that method A saves a significant number of iterations in solving for the current flow in a

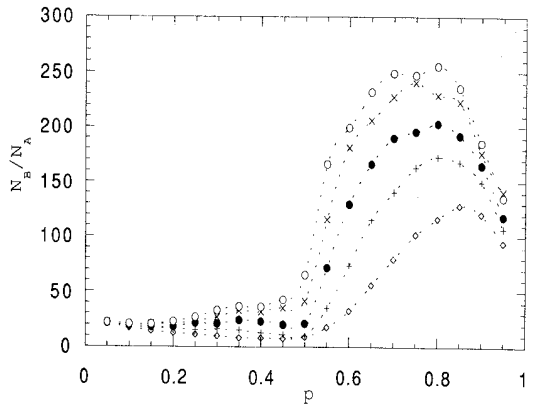

FIG. 7. N_B/N_A is plotted vs p for $\sigma_- = 10^{-3}$ (◇), 10^{-4} (+), 10^{-5} (●), 10^{-6} (×), and 10^{-7} (○). In each case, the lattice size is $N = 128 \times 128$. The dashed lines are guides to the eye.

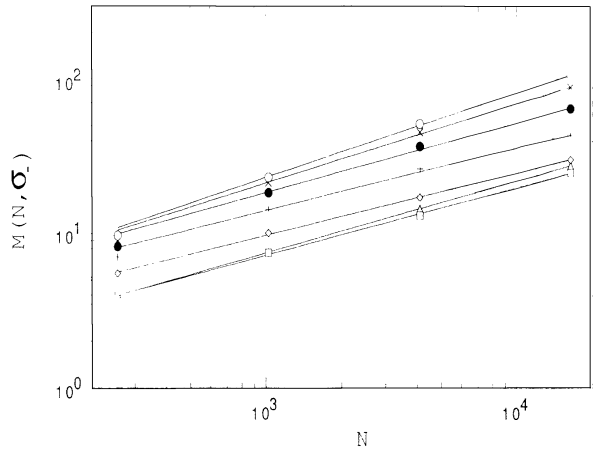

FIG. 8. A log-log plot of $M(N,\sigma_-)$ vs the lattice size N for $\sigma_- = 10^{-1}$ (△), 10^{-2} (□), 10^{-3} (◇), 10^{-4} (●), 10^{-5} (+), 10^{-6} (×), and 10^{-7} (○). The solid lines are linear least-squares fits to the data.

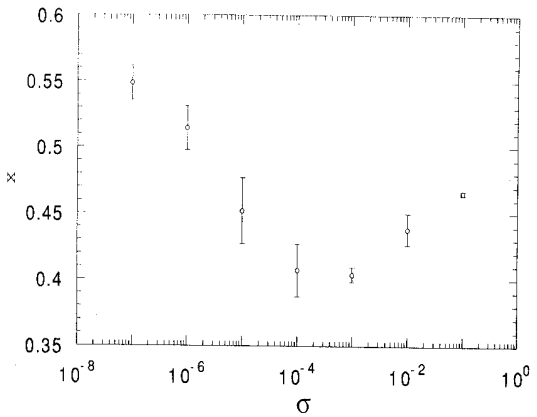

FIG. 9. A semilog plot of the exponent x vs σ_-.

resistor network. However, each iteration in method A requires more operations than in method B, i.e., the overhead is greater in method A. To fully evaluate the efficiency of the two methods, we must consider both the number of iterations and the number of operations at each iteration.

Let us first consider the overhead in method A. Since multiplications are much more time consuming than additions on most processors, we shall only take multiplications into account in determining the overhead. To calculate the $P_i^{(m)}$'s, two FFT's must be carried out. Each of these take $2N \log_2 N$ multiplications [35]. Considering the symmetries of $\tilde{P}^{(m)}(\mathbf{k})$ and $J_{s'}^{(m)}(k_x)$, we find that $12N$ multiplications must be performed to calculate the $\tilde{P}^{(m)}(\mathbf{k})$'s using Eq. (50). Thus, if we compute the $P_i^{(m)}$'s using FFT's, the total number of multiplications needed is $4N \log_2 N + 12N$. This is the largest source of overhead in method A. On average, the remaining number of multiplications needed is $10pN$. The average number of multiplications needed per iteration in method A is $4N \log_2 N + 10pN + 12N$.

The number of multiplications needed for each iteration in method B is $10N$. Let $\eta(N,p,\sigma_-)$ be the ratio of the average CPU time needed to calculate the currents in the network using method B to the corresponding time in method A. We have

$$\eta(N,p,\sigma_-) \simeq \frac{10}{4\log_2 N + 10p + 12} \left[\frac{N_B(N,p,\sigma_-)}{N_A(N,p,\sigma_-)} \right] .$$

If the value of $\eta(N,p,\sigma_-)$ is greater than 1, then method A is more efficient than method B.

We first consider the case $\sigma_- = 0$. In Fig. 10, η is plotted vs p for several different lattice sizes. This figure shows that method A is not always more efficient than method B. However, for a given value of N, method A is always more efficient when p is sufficiently small. Moreover, for a given p, method A is the more efficient of the two methods for sufficiently large lattice sizes. The greater efficiency of method A for small p stems from the fact that the GFF linear system is typically quite small in this regime. For p close to 0.5 and for small lattices, the greater overhead in method A makes it less efficient than method B.

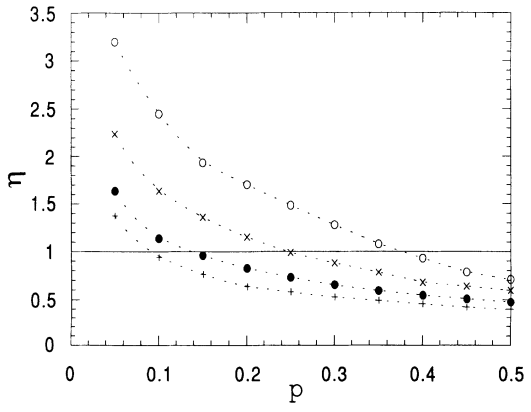

FIG. 10. The relative efficiency η as a function of p for the lattice sizes 16×16 (+), 32×32 (●), 64×64 (×), and 128×128 (○). In each case, $\sigma_- = 0$. The dashed lines are guides to the eye.

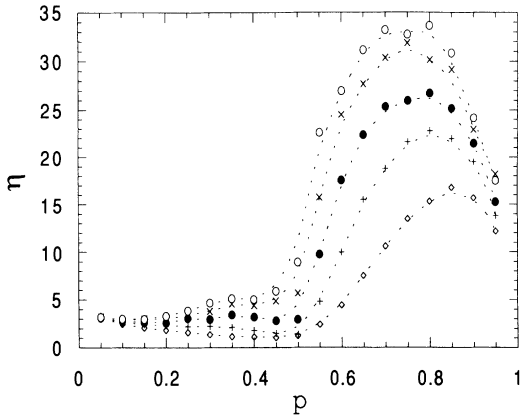

FIG. 11. η is plotted vs p for $\sigma_- = 10^{-3}$ (◇), 10^{-4} (+), 10^{-5} (●), 10^{-6} (×), and 10^{-7} (○). In each case, the lattice size is $N = 128\times128$. The dashed lines are guides to the eye.

Now let us turn to the case in which $\sigma_- > 0$. Figure 11 shows η as a function of p for several different values of σ_-. The lattice size N is 128×128. For p less than 0.4, the relative efficiency η is greater than, 1 although in all cases it is less than 6. In this regime, method A is the more efficient of the two algorithms, although not overwhelmingly so. As p is increased beyond 0.5, however, method A becomes vastly more efficient than method B. For example, for $N = 128\times128$, $\sigma_- = 10^{-7}$, and $p = 0.7$, method B requires more than 30 times more CPU time than method A. This is of particular interest, since in many simulations of breakdown processes, Kirchhoff's laws must be repeatedly solved for random resistor networks in which the majority of the bonds have a small nonzero conductance σ_-.

IV. CONCLUSIONS

In this paper, Green's functions were used to reformulate Kirchhoff's equations for random resistor networks with two types of resistors. In this Green's function formulation (GFF), the current correlation between any pair of resistors in the network is explicitly taken into account. The GFF yields a linear system equivalent to Kirchhoff's laws, but with a smaller number of variables.

A variety of algorithms can be used to solve the GFF linear system, including iterative methods like the conjugate-gradient method. Since the conjugate-gradient method is frequently employed in simulations of random resistor networks, we used this method to solve the GFF linear system. We call this algorithm method A. Our extensive numerical work showed that in two important regimes, method A is considerably more efficient than solving Kirchhoff's laws directly using the conjugate-gradient method (method B).

We compared the efficiency of methods A and B by solving for the voltages in a two-component random resistor network in which a fraction p of the resistors has

conductance σ_-, and the remainder has conductance $\sigma_+=1$. In all cases, the computation was terminated once a precision of 10^{-10} had been achieved. For a 128×128 grid with $\sigma_-=0$, the number of iterations needed in method B was 5–22 times greater than in method A (the relative efficiency of the two methods depends on the value of p). This reduction in the number of iterations is to some degree offset by the greater overhead in method A. We found that method A is more efficient than method B when p is sufficiently small or N is sufficiently large. Method A is more efficient than method B when σ_- is small but nonzero for all values of p and lattice sizes N we studied. Method A is particularly efficient when $0<\sigma_-\ll\sigma_+$ and $p>p_c$. For example, for a 128×128 square grid with $p=0.8$ and $\sigma_0/\sigma_+=10^{-7}$, method B required roughly 250 times more iterations than method A to reach the desired precision. Even when overhead is taken into account, method A is more than 30 times more efficient than method B. Method A therefore provides an effective means of testing scaling forms for the resistance fluctuations in a two-component random resistor network with $0<\sigma_-/\sigma_+\ll1$.

The high efficiency of method A stems in part from the fact that a change in the voltage at one site propagates to all other sites in the network in a single iteration. This is also true of the Fourier-accelerated conjugate-gradient (FACG) method. However, our GFF approach is fundamentally different than the FACG method. In the FACG technique, long-range couplings are introduced through the use of a preconditioning matrix E with entries $E(\mathbf{r},\mathbf{r}')$ that are proportional to $|\mathbf{r}-\mathbf{r}'|^{-\phi}$. The exponent ϕ is chosen to optimize the performance. This optimization is accomplished through repeated numerical tests of the efficiency achieved for different choices of ϕ. There are indications that the efficiency depends sensitively on the value of ϕ [32].

Method A has no adjustable parameters, and so no optimization is needed. In addition, we have shown that our technique delivers exceptional performance for small σ_-, particularly when p is greater than p_c. Although the FACG method has not been tested in this regime, we expect it to suffer from much the same problems as the conjugate-gradient method when σ_- is small.

Critical showing down occurs in method A as the percolation threshold is approached. However, the multigrid method has been shown to substantially reduce the effects of critical slowing down. If the multigrid method [31] were used to solve the GFF linear system, critical slowing down would presumably be all but eliminated.

Although we tested method A on random resistor networks with uncorrelated disorder, our technique can equally well be applied to nets with correlated disorder. In particular, it can be used to study breakdown in random resistor networks. As we noted in Sec. I, random resistor networks in which a majority of the bonds have a small nonzero conductance σ_- arise frequently in simulations of breakdown models. We have shown that method A is a vastly more efficient method of solving this type of problem than the traditional method of solution (method B). Since method A is more efficient than method B and yet is not difficult to implement, we hope that method A will eventually supplant method B in future work on breakdown in resistor networks.

Our tests of method A were confined to two-component random resistor networks in two dimensions. However, method A is readily generalized to networks with many different types of resistors, and applies in any space dimension.

In the present paper, we considered random resistor networks exclusively. However, a GFF can be developed for random networks of Hookean springs, following almost the same procedure as for random resistor networks [36]. We expect that simulations of random elastic networks based on the solution of the resultant GFF linear system will be very efficient. This subject will be dealt with in detail in our future work.

ACKNOWLEDGMENTS

We would like to thank Paul Beale, B. J. Buchalter, and Mark Gyure for interesting discussions. This work was supported by NSF Grant No. DMR-9100257.

[1] S. Kirkpatrick, Rev. Mod. Phys. **45**, 574 (1973).
[2] D. Stauffer, *Introduction to Percolation Theory* (Taylor & Francis, London, 1985).
[3] J. P. Straley, J. Phys. C **9**, 783 (1976).
[4] D. C. Hong, H. E. Stanley, A. Coniglio, and A. Bunde, Phys. Rev. B **33**, 4564 (1986).
[5] L. de Arcangelis and A. Coniglio, J. Stat. Phys. **48**, 935 (1987).
[6] R. R. Tremblay, G. Albinet, and A. M. S. Tremblay, Phys. Rev. B **43**, 11 546 (1991).
[7] R. R. Tremblay, G. Albinet, and A. M. S. Tremblay, Phys. Rev. B **45**, 755 (1992).
[8] *Disorder and Fracture,* edited by J. C. Charmet, S. Roux, and E. Guyon (Plenum, New York, 1990).
[9] L. de Arcangelis, S. Redner, and H. J. Herrmann, J. Phys. (Paris) Lett. **46**, L585 (1985).
[10] P. M. Duxbury, P. D. Beale, and P. L. Leath, Phys. Rev. Lett. **57**, 1052 (1986).
[11] P. M. Duxbury, P. L. Leath, and P. D. Beale, Phys. Rev. B **36**, 367 (1987).
[12] M. Söderberg, Phys. Rev. B **35**, 352 (1987).
[13] H. Takayasu, Phys. Rev. Lett. **54**, 1099 (1985).
[14] B. K. Chakrabarti, K. K. Bardhan, and P. Ray, J. Phys. C **20**, L57 (1987).
[15] S. S. Manna and B. K. Chakrabarti, Phys. Rev. B **36**, 4078 (1987).
[16] P. D. Beale and P. M. Duxbury, Phys. Rev. B **37**, 2785 (1988).
[17] D. R. Bowman and D. Stroud, Phys. Rev. B **40**, 4641 (1989).
[18] M. F. Gyure and P. D. Beale, Phys. Rev. B **40**, 9533 (1989).

[19] M. F. Gyure and P. D. Beale, Phys. Rev. B **46**, 3736 (1992).
[20] R. M. Bradley, D. Kung, S. Doniach, and P. N. Strenski, J. Phys. A **20**, L911 (1987).
[21] R. M. Bradley, D. Kung, P. N. Strenski, and S. Doniach, Physica B **152**, 282 (1988).
[22] P. L. Leath and W. Tang, Phys. Rev. B **39**, 6485 (1989).
[23] W. Xia and P. L. Leath, Phys. Rev. Lett. **63**, 1428 (1989).
[24] E. L. Hinrichsen, S. Roux, and A. Hansen, Physica C **167**, 433 (1990).
[25] M. R. Hestenes and E. Stiefel, Natl. Bur. Stand. J. Res. **49**, 409 (1952).
[26] G. H. Golub and C. F. van Loan, *Matrix Computations* (North Oxford, London, 1986), p. 516.
[27] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer, New York, 1980), p. 572.
[28] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems* (Academic, London, 1984), Chap. 1.
[29] G. G. Batrouni, A. Hansen, and M. Nelkin, Phys. Rev. Lett. **57**, 1336 (1986).
[30] G. G. Batrouni and A. Hansen, J. Stat. Phys. **52**, 747 (1988).
[31] R. G. Edwards, J. Goodman, and A. D. Sokal, Phys. Rev. Lett. **61**, 1333 (1988).
[32] M. F. Gyure (private communication).
[33] Equation (21) actually gives U only up to an overall additive constant. The value of this constant is obtained from the boundary condition (6a) in our numerical work.
[34] G. H. Golub and C. F. van Loan, *Matrix Computations* (Ref. [26]), Chap. 3.
[35] D. F. Elliot and K. R. Rao, *Fast Transforms: Algorithms, Analyses, Applications* (Academic, Orlando, 1982).
[36] K. Wu and R. M. Bradley (unpublished).