SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2012 Vol. 55 No. 4: 938–950
doi: 110.1007/s11432-011-4479-1

Localized relaxation theory of circuits and its applications in electro-thermal analyses

LUO ZuYing[1]*, ZHAO GuoXing[1], GORDON Joseph A.[2] & TAN Sheldon X.-D.[2]

[1]*College of Information Science and Technology, Beijing Normal University, Beijing* 100875*, China;*
[2]*Department of Electrical Engineering, University of California, Riverside, CA* 92521*, USA*

Received January 29, 2010; accepted April 8, 2011; published online January 2, 2012

Abstract In the high-performance IC design with increasing design complexity, it is a very important design content to efficiently analyze IC parameters. Thus, the electro-thermal (ET) analyses including power/ground (P/G) analysis and thermal analysis are hot topics in today's IC research. Since ET analysis equation has a sparse, positive definite and strictly diagonally dominant coefficient-matrix, we prove that the ET analysis has the advantage of locality. Owing to this advantage, localized relaxation method is formally proposed, which has the same accuracy as the global relaxation done with the constraint of the same truncation error limitation. Based on the localized relaxation theory, an efficient and practical localized successive over-relaxation algorithm (LSOR2) is introduced and applied to solve the following three ET analysis problems. (1) Single-node statistical voltage analysis for over-IR-drop nodes in P/G networks; (2) single-node statistical temperature analysis for hot spots in 3D thermal analysis; (3) fast single open-defect analysis for P/G networks. A large amount of experimental data demonstrates that compared with the global successive over-relaxation (SOR) algorithm, LSOR2 can speed up 1–2 orders of magnitudes with the same accuracy in ET analyses.

Keywords integrated circuit, electro-thermal analysis, SOR algorithm, power/ground network

Citation Luo Z Y, Zhao G X, Gordon J A, et al. Localized relaxation theory of circuits and its applications in electro-thermal analyses. Sci China Inf Sci, 2012, 55: 938–950, doi: 10.1007/s11432-011-4479-1

1 Introduction

Driven by performance hunger, the manufacture technology and architecture of high-end chip design are improved constantly. As a result, high performance ICs are becoming more and more complex. How to speed up character analysis in IC design is an eternal topic in electronic design automation (EDA) research [1,2]. Furthermore, the ET analyses including P/G analysis and thermal analysis are a hot EDA topic [1–13].

With technology improvement and frequency increasing, power consumption is becoming one of major design constraints. In order to cut down power consumption in the low-power design, supply voltage should be decreased, which means that the margin of P/G networks is also compacted. Due to large magnitude and high variation frequency of working currents in high-end chips, P/G networks suffer from escalating IR drops and Ldi/dt drops. Since supply voltage directly influences circuit delay, voltage drop of P/G network is limited into a very small margin. Thus engineers have to frequently optimize the P/G

*Corresponding author (email: luozy@bnu.edu.cn)

grid in design process, which means that P/G analyses are needed to solve the nodal voltage vectors for the trial P/G networks [4–11]. Meanwhile high power consumption of high-end chips produces a large amount of heat. If chip package is not carefully designed for heat dissipation, high temperature and hot spots may lower chip performance and reliability. Therefore thermal analyses, especially three-dimension (3D) full-chip thermal analyses, are needed in chip design [12]. Due to employing a general electro-thermal equation for their formulation, P/G analyses and 3D thermal analyses are totally regarded as the electro-thermal (ET) analyses in IC design [3].

Many researchers proposed a lot of efficient ET analysis methods for this hot EDA topic including such notable ones as incomplete cholesky decomposition conjugate gradient (ICCG) algorithm [4], successive over-relaxation (SOR) algorithm [5], multi-grid (MG) algorithm [6], random walk (RW) algorithm [7,8], equivalent circuit algorithm [9,10]. With IC technology scaling into nanometer regime, process variation (PV) is so visible that IC deterministic design evolves into statistical design named design for manufacturing (DFM) [2]. And some statistical P/G analysis methods are also presented to deal with PV influences in [8,10–12], including the statistical equivalent circuit method [10] and the single-node SOR method(SNSOR) [11,12] proposed by authors. SNSOR can efficiently and accurately solve the voltage and temperature variations in over-IR-drop nodes and hot spots. SNSOR is 1–2 orders of magnitudes faster than the traditional global ICCG method and global SOR method (GSOR). Based on SNSOR, we further proposed a single open-defect P/G analysis method (SDSOR) [13].

Based on our solid ET analysis researches [2–4,9–13], especial SNSOR algorithm to analyze single-node voltage variations for over-IR-drop nodes [11], here we deduce the localized theory about ET analyses from the point of math view. Since ET analysis equation has the sparse coefficient matrix of positive definite character, this paper proves that the ET analysis has the advantage of locality and owing to this advantage, localized relaxation is as accurate as the global relaxation with the constraint of the same truncation error limitation. Furthermore, this paper proposes an efficient and practical localized successive over-relaxation algorithm (LSOR2). Over SNSOR proposed by ourselves, LSOR2 shows some advantages such as theory completion and generality, low complexity, definite accuracy, wide applications.

At the last part of the paper, LSOR2 is used to solve the following three ET analysis problems.

(1) Single-node statistical voltage analysis for over-IR-drop nodes in P/G networks.

(2) Single-node statistical temperature analysis for hot spots in 3D thermal analysis.

(3) Fast single open-defect analysis for P/G networks.

A large amount of experimental data demonstrate that compared with the global successive over-relaxation (SOR) algorithm, LSOR2 can speed up 1–2 orders of magnitudes with the same accuracy in ET analyses. LSOR2 is more efficient and more accurate than RW [7,8], another localized P/G analysis algorithm. Compared with present circuit relaxation technology, the localized relaxation theory and algorithm proposed in the paper not only improve the SOR method but also enrich present theory on circuit relaxation, which means that our work is an important theoretic supplement on present analysis theory of large-scale electric grid.

2 Research background

2.1 ET analysis equation

P/G analysis and 3D thermal analysis employ the same formulation [3].

$$GX = B. \tag{1}$$

This formulation is known as the ET analysis equation. Each term in the equation has similar definition in P/G analysis and 3D thermal analysis. In P/G analyses [3–11], B and G are the current vector and conductance matrix, respectively, while X is the nodal voltage vector. With the known B and G, Eq. (1) is used to solve the corresponding X. As B and X are replaced by I and V, Eq. (1) is transformed into the standard P/G analysis formulation as follows:

$$GV = I. \tag{2}$$

In 3D thermal analysis [12], B and G are the power density vector and thermal conductance matrix, respectively, while X is the nodal temperature vector. With the known B and G, Eq. (1) is used to solve the corresponding X. As B and X are replaced by P and T, Eq. (1) is transformed into the standard 3D thermal analysis formulation as follows:

$$GT = P. \tag{3}$$

Since P/G analysis and 3D thermal analysis employ the same ET analysis equation, they are classified as the ET analysis [3], which means that all P/G analysis methods can be directly used in 3D thermal analysis [3,12]. The coefficient matrix G is a five-diagonal matrix in P/G analysis, and is a seven-diagonal matrix in 3D thermal analysis.

2.2 Gauss-Seidel and SOR iterative method

Gauss-Seidel iterative method can be used to solve Eq. (1) as follows [5]:

$$x_i^{(k+1)} = \frac{1}{\sum_{j\in N_i} g_{i,j}} \left(b_i + \sum\nolimits_{j\in N_i} (g_{i,j} x_j^{(k)}) \right), \tag{4}$$

where $x_j^{(k)}$ is jth nodal voltage (or temperature) after kth relaxation iteration, b_i is jth nodal input stimulus of absorption current (or power density), N_i is the set of all neighbor nodes of node i, and $g_{i,j}$ is interconnecting conductor (thermal conductor) between node i and j.

Successive over-relaxation (SOR) iterative method is another method to solve Eq. (1)

$$x_i^{(k+1)} = (1-\omega)\, x_i^{(k)} + \frac{\omega}{\sum_{j\in N_i} g_{i,j}} \left(b_i + \sum\nolimits_{j\in N_i} (g_{i,j} x_j^{(k)}) \right), \tag{5}$$

where ω is the relaxation factor to be optimized to make the convergence faster. The optimal value ω_{opt} can be calculated with

$$\omega_{\text{opt}} = \frac{2}{1+\sqrt{1-\rho(E-D^{-1}G)^2}}, \tag{6}$$

where D is the diagonal matrix of G, E is the identity matrix, and ρ is the spectral radius function of $E - D^{-1}G$. There is also a one-dimension searching method for finding ω_{opt} [11]. The SOR method can be easily coded into global SOR iterative algorithm (GSOR) shown in Figure 1.

2.3 Single-node SOR iterative algorithm (SNSOR)

Ref. [11] first proposed the efficient single-node SOR relaxation algorithm (SNSOR) shown in Figure 2. SNSOR simulates the wave transmission process, in which one relaxation iteration is regarded as a vibration-wave transmission. The problematic node q is input 1A stimulus while the other nodes are non-stimuli, which means q is the vibration source. During the wave transmission process, edge nodes with large amplitude evolution δ are upgraded into internal nodes if δ is greater than a pre-defined value $\varepsilon_2 \ll 1$. And then, all un-relaxed neighbor nodes of new-upgraded internal nodes are marked as the edge nodes and are added into the relaxation set, which means that the wave transmits and ends at edges nodes of $\delta < \varepsilon_2$. An internal node can be marked as the relaxation-ending node if its amplitude evolution δ is less than a pre-defined value $\varepsilon_3 \ll 1$. As for an edge node, if its all neighbor nodes are relaxation-ending nodes, it is defined as an isolated vibration node and can be marked as the relaxation-ending node. As the maximum vibration amplitude $\max \delta < \varepsilon_1$, a pre-defined relaxation limitation, SNSOR ends, which means that the wave transmission is at the dynamic balance state.

Compared with the traditional GSOR, SNSOR is a localized relaxation algorithm and shows the advantages of high accuracy and low complexity [11]. But since SNSOR is the simple simulation about the natural vibration-wave transmission, it does not have the theory completion. Thus we focus on theoretically inducing a localized analysis algorithm of the theory completion with as high efficiency as SNSOR, but with wider applications.

Global SOR algorithm (GSOR)

Step 1: Initialization

1.1) Input coefficient matrix G and stimulus vector B.

1.2) Input initial vector $X^{(0)}$, generally assign all nodes with supply voltage V_{DD} or ambient temperature T_a.

1.3) Assign the truncation error ε, and assign $k = 0$.

Step 2: $(k+1)$th relaxation iteration.

2.1) If all nodes are relaxed, go to Step 3; else, sequentially relax the next node i.

2.2) Use Eq. (5) to calculate $x_i^{(k+1)}$ and $\delta = |x_i^{(k)} - x_i^{(k+1)}|$.

2.3) If $\delta > \varepsilon$, then $S_i^{(k+1)} = 1$; else $S_i^{(k+1)} = 0$.

2.4) Go to Step 2.1.

Step 3: End judging

If $S_i^{(k+1)} = 0$ for all nodes i, end the relaxation and output $X^{(k)}$;

Else let $k = k + 1$ and turn to Step 2.

Figure 1 GSOR algorithm flowchart.

Single-node SOR algorithm (SNSOR)

Step 1: Initiation

1.1) Input coefficient matrix G and stimulus vector B.

1.2) Input initial vector $X^{(0)}$, generally assign all nodes with supply voltage V_{DD} or ambient temperature T_a.

1.3) Assign the maximum truncation error ε_1, standard truncation error $\varepsilon_2 = 0.1\varepsilon_1$, and minimum truncation error $\varepsilon_3 = 0.01\varepsilon_1$.

1.4) Reset relaxation mark flag1[i] for all nodes (0 means un-relaxed node, 1 means edge node, 2 means internal node), and reset relaxation ending mark flag2[i] (0 means not ending, 1 means relaxation ending).

1.5) Add problematic node q with 1A stimulus in initial relaxation set$A^{(0)}$ and set the nodal relaxation mark flag1[q]=1, which means q is an edge node. Then assign $k = 0$ and the maximum relaxation improvement value max$\delta = 0.0$.

Step 2: The $(k+1)$th relaxation iteration

2.1) If all nodes in $A^{(k)}$ have been relaxed, turn to Step 3; else, sequentially take a node i from $A^{(k)}$ and go to the next step.

2.2) Use Eq. (5) to compute $x_i^{(k+1)}$ and the relaxation improvement value $\delta = |x_i^{(k)} - x_i^{(k+1)}|$, refresh max δ with δ.

2.3) If flag1[i]=2 and $\delta < \varepsilon_3$, assign flag2[i]=1, which means marking node i as the relaxation ending node.

2.4) If flag1[i]=1 and $\delta \leqslant \varepsilon_2$, check the relaxation isolated state of i. If i is a relaxation isolated node, mark i as the relaxation ending node.

2.5) If flag1[i]=1 and $\delta > \varepsilon_2$, assign flag1[i]=2, which means upgrading i as an internal node, and upgrade its all un-relaxed neighbor nodes as edge nodes and add them in $A^{(k)}$.

2.6) Return to Step 2.1.

Step 3: Refresh all pre-conditions for the next relaxation iteration

Add all un-ending nodes in $A^{(k)}$ in $A^{(k+1)}$ and then, reset flag1 and flag2 marks for all relaxation ending nodes in $A^{(k)}$.

Step 4: End judging

If max $\delta \leqslant \varepsilon_1$, end SNSOR and output $X^{(k)}$; else, assign $k = k + 1$, max $\delta = 0.0$, and turn to Step 2.

Figure 2 SNSOR algorithm flowchart [11].

3 Localized SOR analysis theory

3.1 Localized SOR theory

In the ET analysis equation (1), coefficient matrix G is of the following characters according to the Kirchhoff current law (KCL) [5].

Theorem 1. Matrix G in ET analysis equation Eq. (1) satisfies

1) G is symmetry.

2) G's diagonal elements are positive.

3) G is weakly diagonal dominanted.

4) G is positive definite.

In other words, G is a weakly diagonal dominanted and positive definite large-scale sparse matrix. According to Theorem 1, there is exactly one unique solution for the ET analysis equation (1).

As G and B are locally modified in P/G grid optimizations or ECO designs, it is inefficient to directly use global algorithms such as GSOR [5] and ICCG [4] to solve Eq. (1). On the other hand, SNSOR [11, 12] and single open-defect SOR algorithm (SDSOR) apply the locality property of ET analysis [10] to localized relax P/G grids for low algorithm complexity. In the localized relaxation for a test case of N nodes, we assume only $m(m \ll N)$ nodes of 1A stimulus and the other nodes without any stimulus. And these m nodes consist of the initial relaxation set $A^{(0)}$. The formulation expresses the local stimuli

$$GX = \sum\nolimits_{q \in A^{(0)}} B_q. \tag{7}$$

Here $B_q = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^{\mathrm{T}}$ is the input vector in which only node q is under 1A stimulus and the other nodes are under no stimuli.

Theorem 2. Solving Eq. (7) by Formula (5) satisfies the following properties:

1) for any i, if for all $j \in N_i, |x_j^{(k)} - x_j^{(k-1)}| \leqslant \varepsilon$, then $|x_i^{(k+1)} - x_i^{(k)}| \leqslant (2\omega + 1)\varepsilon$,

2) for any i, if for all $j \in N_i, x_j^{(k-1)} = x_j^{(k)}$, then $x_i^{(k+1)} = x_i^{(k)}$.

Proof.

1) By $x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{\sum_{j \in N_i} g_{i,j}}(b_i + \sum_{j \in N_i}(g_{i,j}x_j^{(k)}))$, we get

$$\begin{aligned}\left|x_i^{(k+1)} - x_i^{(k)}\right| &= \left|(1-\omega)\left(x_i^{(k)} - x_i^{(k-1)}\right) + \frac{\omega}{\sum_{j \in N_i} g_{i,j}}\left(\sum\nolimits_{j \in N_i} g_{i,j}\left(x_j^{(k)} - x_j^{(k-1)}\right)\right)\right| \\ &\leqslant |1-\omega|\left|x_i^{(k)} - x_i^{(k-1)}\right| + \frac{\omega}{\sum_{j \in N_i} g_{i,j}}\left(\sum\nolimits_{j \in N_i} g_{i,j}\left|x_j^{(k)} - x_j^{(k-1)}\right|\right) \\ &\leqslant (1+\omega)\varepsilon + \frac{\omega}{\sum_{j \in N_i} g_{i,j}}\left(\sum\nolimits_{j \in N_i} g_{i,j}\varepsilon\right) = (2\omega+1)\varepsilon.\end{aligned}$$

2) It is a direct conclusion of 1).

In one word, according to Theorem 2, ET analysis is of locality property under the local stimuli.

3.2 Elementary localized SOR algorithm (LSOR1) and its accuracy analysis

Directly according to Theorem 2, we propose an elementary localized SOR algorithm (LSOR1) to solve Eq. (7). And the following flowchart is used to describe LSOR1.

Theorem 3. LSOR1 with truncation error ε is as accurate as GSOR with truncation error $(2\omega + 1)\varepsilon$.

Proof. We prove by induction of iteration steps k.

1) If $k = 0$, then $x_i^{(k)} = V_{DD}$(or Ta). It is obviously correct.

2) Assume that the result of kth iteration of LSOR1 algorithm is as accurate as the GSOR with truncation error $(2\omega + 1)\varepsilon$.

3) Then for the $(k + 1)$th iteration

a) if $S_i^{(k)} = 1$, LSOR1 computes by Eq. (5), which means LSOR1 gets the same result as GSOR does;

Localized SOR algorithm 1 (LSOR1)

Step 1: Initiation.

1.1) Input coefficient matrix G and stimulus vector B.

1.2) Input initial vector $X^{(0)}$, generally assign all nodes with supply voltage V_{DD} or ambient temperature T_a.

1.3) m nodes under 1A stimulus consist of initial relaxation set $X^{(0)}$.

1.4) Let $S_i^{(0)} = 1$ for $i \in X^{(0)}$ and $S_i^{(0)} = 0$ for $i \notin X^{(0)}$.

1.5) Assign truncation error ε and let $k = 0$.

Step 2: The $(k+1)$th relaxation iteration.

If $S_i^{(k)} = 1$, use Eq. (5) to compute $x_i^{(k+1)}$; else, assign $x_i^{(k+1)} = x_i^{(k)}$.

Step 3: Refresh the pre-conditions for the next relaxation iteration

3.1) For all i, let $S_i^{(k+1)} = 0$.

3.2) For all i, if $|x_i^{(k+1)} - x_i^{(k)}| > \varepsilon$, then let $S_i^{(k+1)} = 1$ and $S_j^{(k+1)} = 1, j \in N_i$.

Step 4: End judging

If $S_i^{(k+1)} = 0$ for all nodes i, end the relaxation and output $X^{(k)}$;

Else, let $k = k + 1$ and turn to Step 2.

Figure 3 LSOR1 algorithm flowchart.

b) if $S_i^{(k)} = 0$, LSOR1 computes by setting $x_i^{(k+1)} = x_i^{(k)}$, while GSOR still computes by Eq. (5) and the result is denoted as $\overline{x}_i^{(k+1)}$. $S_i^{(k)} = 0$ guarantees implies $j \in N_i, |x_j^{(k-1)} - x_j^{(k)}| \leqslant \varepsilon$. In addition with Theorem 2, we get

$$\left|x_i^{(k+1)} - \bar{x}_i^{(k+1)}\right| = \left|x_i^{(k)} - \bar{x}_i^{(k+1)}\right| \leqslant (2\omega + 1)\varepsilon,$$

which means that the truncation error of $(k+1)$th iteration is less than $(2\omega + 1)\varepsilon$.

For the GSOR algorithm with truncation error ε, the accuracy is guaranteed by the following theorem.

Theorem 4. The accuracy of GSOR algorithm

1) The GSOR alorithm is numerically stable for the coefficient matrix satisfying the condition of Theorem 1; furthermore, the accuracy is guaranteed by (see [14])

$$\overline{\lim}_{k\to\infty}|x_k - \alpha| \leqslant \varepsilon c\|\alpha\| cond(A),$$

where α is the theoretical solution, $cond(A)$ is the conditional number of matrix A and c is a constant.

2) If the truncation error is a uniform random variable with expectation ε, then the error of GSOR algorithm is (see [15])

$$r = \frac{1}{\omega} A^{-1}\varepsilon.$$

Combining Theorem 3 and Theorem 4, we reach the following conclusions: 1) LSOR1 algorithm has similar accuracy to GSOR algorithm; 2) LSOR1 has theoretical foundation of its own accuracy, showing that the LSOR1 algorithm is effective and reliable.

3.3 Improvement on localized SOR algorithm

The LSOR1 algorithm mentioned above shows its weakness on slow convergence and inefficiency though it is intuitive and easy to understand. Based on our previous researches [11–13], Figure 4 presents an improved and practical localized SOR algorithm (LSOR2).

Over LSOR1, LSOR2 has following advantages.

(1) In Step 2.3, if $\delta = |x_i^{(k)} - x_i^{(k+1)}| > \varepsilon$ for node i, add its un-relaxed neighbor node $j(j \in N_i)$ in $A^{(k)}$, which may lengthen the relaxation set and reduce the number of relaxation iterations.

(2) In Step 3, reset $S_i^{(k)}$ for node $i \in A^{(k)}$ and then set $S_j^{(k+1)} = 1$ for node $j \in A^{(k+1)}$, which may avoid the LSOR1's global operation on reset relaxation marks for all nodes.

(3) In Step 4, end-judging condition that $A^{(k+1)}$ is empty may avoid the LSOR1's global operation on checking if $S_i^{(k+1)} = 0$ for all nodes.

As a result, the improved LSOR2 algorithm highlights the locality property and increases the algorithm efficiency.

LSOR2 algorithm

Step 1: Initiation.

1.1) Input coefficient matrix G and stimulus vector B.

1.2) Input initial vector $X^{(0)}$, generally assign all nodes with supply voltage V_{DD} or ambient temperature T_a.

1.3) m nodes under 1A stimulus consist of initial relaxation set $X^{(0)}$.

1.4) Let $S_i^{(0)} = 1$ for $i \in X^{(0)}$ and $S_i^{(0)} = 0$ for $i \notin X^{(0)}$

1.5) Assign truncation error ε and $k = 0$.

Step 2: The $(k+1)$th relaxation iteration.

2.1) If all nodes in $A^{(k)}$ have been relaxed, turn to Step 3; else, sequentially take. a node i from A and go to the next step.

2.2) Use Eq. (5) to compute $x_i^{(k+1)}$ and the relaxation improvement value $\delta = |x_i^{(k)} - x_i^{(k+1)}|$.

2.3) If $\delta > \varepsilon$, add i into $A^{(k+1)}$ and then, to its neighbor node $j \in N_i$, if $S_j^{(k)} = 0$, set $S_j^{(j)} = 1$ and add j into $A^{(k)}$.

2.4) Return to Step 2.1.

Step 3: Refresh the pre-conditions for the next relaxation iteration.

Reset $S_i^{(k)}$ for node $i \in A^{(k)}$ and then, set $S_i^{(k)} = 1$ for node $i \in A^{(k+1)}$.

Step 4: End judging.

If $A^{(k+1)}$ is empty, end the relaxation and output $X^{(k)}$;

Else, let $k = k + 1$ and turn to Step 2.

Figure 4 LSOR2 algorithm flowchart.

3.4 Experimental verification on superiority of localized SOR algorithms

In order to verify the superiority of localized SOR algorithms, we do comparison experiments with present GSOR [5] and SNSOR [11], elementary LSOR1 and practical LSOR2 proposed by this work. For accuracy comparison, we assume that all algorithms have the same truncation error 10^{-9}. According to the definition in SNSOR flowchart, the standard truncation error $\varepsilon_2 = 0.1\varepsilon_1 = 10^{-10}$ and the minimum truncation error $\varepsilon_3 = 0.01\varepsilon_1 = 10^{-11}$. In experiments, we choose a problem node q and impact a single-stimulus vector I_q on the circuit, in which q is under 1A stimulus and the other under no stimulus. Four comparison algorithms are used to solve the corresponding voltage vector V_q for comparing the locality and accuracy. Then 900 problem nodes are used to compare the runtime of four algorithms.

Experimet 1. A circuit of 202440 nodes is used to compare the convergence, relaxation locality, and maximum relaxation error of four algorithms. The iteration number N_{it} is used to verify the algorithm convergence and less N_{it} means the better convergence. The number of effective relaxation nodes $NN_{\mathrm{SOR}}^{(k)}$ is used to verify the algorithm locality and less $NN_{\mathrm{SOR}}^{(k)}$ means the better locality. $NN_{\mathrm{SOR}}^{(k)}$ is the number of all nodes of $\delta > \varepsilon$ for GSOR and $NN_{\mathrm{SOR}}^{(k)}$ is the length of $A^{(k)}$ for the other algorithm. And $E_{\max}^{(k)}$ is the maximum relaxation error among all nodal error $e_i^{(k)}$ in the kth iteration. The following equation is used to compute $e_i^{(k)}$:

$$e_i^{(k)} = \left| x_i^{(k)} - \frac{b_i + \sum_{j \in N_i} (g_{i,j} x_j^{(k)})}{\sum_{j \in N_i} g_{i,j}} \right|, \tag{8}$$

where all terms are defined in Eq. (5).

Figure 5 shows $E_{\max}$ changes with relaxation iterations. We can draw the following conclusions from the figure.

(1) Four algorithms produce nearly the same $E_{\max}$, which is guaranteed by the assertion proposed in Theorem 3 and Theorem 4. The localized relaxation algorithm is as accurate as the global one.

(2) SNSOR and LSOR2 have the fastest convergence speed from the viewpoint of $E_{\max}$.

Figure 6 shows that $NN_{\mathrm{SOR}}^{(k)}$ changes with relaxation iterations. We can draw the following conclusions from Figure 6. 1) GSOR (in Figure 1) must relax all nodes in the circuit, but effective relaxation nodes of $\delta > \varepsilon$ are only corresponding to 5% of total nodes in the circuit, suggesting that with single-node

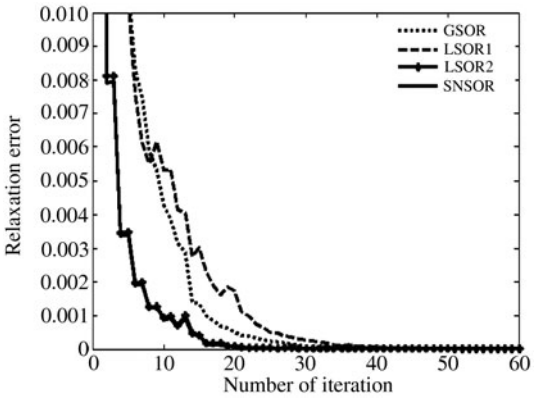

Figure 5 $E_{\max}$ convergence speed.

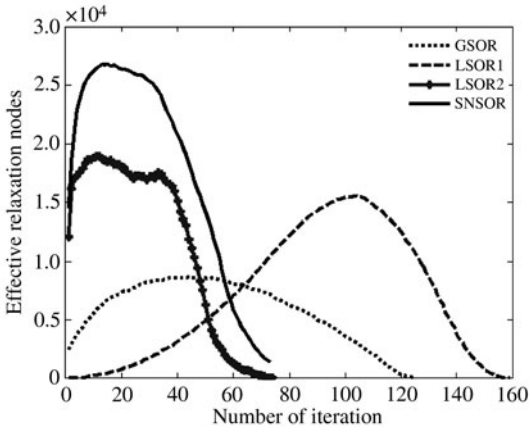

Figure 6 The number of effective relaxation nodes.

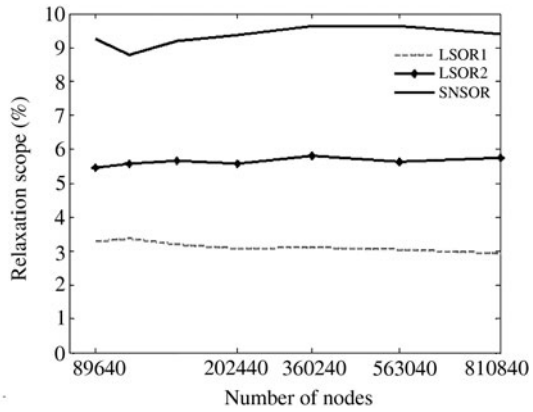

Figure 7 Relaxation scope comparison.

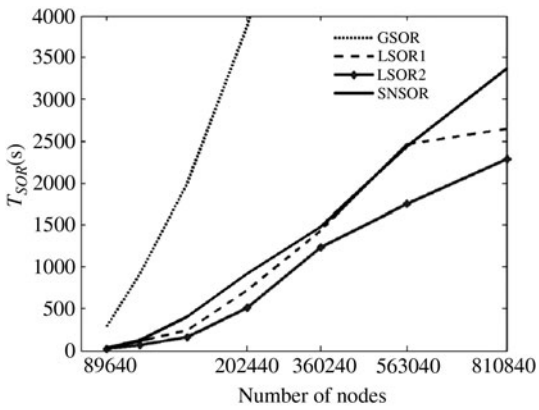

Figure 8 T_{SOR} comparison.

stimulus or local stimuli, P/G analysis is of locality property guaranteed by Theorem 2. 2) LSOR1 (in Figure 3) needs the most iterations and relaxes the minimum nodes in each iteration. $N_{\rm it}$ of LSOR1 is 159 while $N_{\rm it}$ of LSOR2 and GSOR are 74 and 125 respectively. Thus LSOR1 has the weakness of slow convergence. 3) SNSOR (in Figure 2) has the minimum $N_{\rm it}$ but relaxes the maximum nodes in each iteration among three localized SOR algorithms. 4) LSOR2 (in Figure 4) overcomes not only LSOR1 weakness on slow convergence but also SNSOR shortcome on maximum effective relaxation nodes, which means LSOR2 is a wonderful localized relaxation algorithm with fast convergence and small relaxation scope.

Experiment 2. Accumulating all $NN_{\rm SOR}^{(k)}$ in Figure 6 to obtain $NNA_{\rm SOR}$, which is the number of all effective relaxation nodes in total SOR process, and then using the equation $NN_{\rm SOR} = NNA_{\rm SOR}/N_{\rm it}$ to obtain the number of the average effective relaxation nodes per iteration, and last using $NN_{\rm SOR}/N$ to obtain the relaxation scope for comparison algorithms, where N is the node number of the circuit. This work uses seven circuits to investigate the relaxation scope for three localized SOR algorithms and all data are listed in Figure 7. Horizontal ordinate in Figure 7 is the node number of experimental circuits. We can draw the following conclusions:

(1) All three localized relaxation algorithms show perfect locality property and relaxation scopes are less than 10% of experimental circuits. LSOR1 is of the best locality property with about 3% relaxation scopes though it has the slowest convergence speed.

(2) As for LSOR2 and SNSOR of similar convergence speed, the relaxation scope of LSOR2 is 5%–6% and that of SNSOR is 8%–10%.

Thus LSOR2 has better locality property than SNSOR, which means that LSOR2 has lower complexity than SNSOR.

Experiment 3. We choose 900 problematic nodes in each circuit and solve 900 voltage vectors under single-node stimulus, which takes T_{SOR} runtime. All T_{SOR} data listed in Figure 8 demonstrate the following facts:

(1) GSOR is slowest and certainly of highest complexity.

(2) LSOR2 is fastest and certainly of least complexity.

(3) Compared with SNSOR presented by authors, LSOR2 may visibly reduce the algorithm complexity.

Conclusion. Combining conclusions drawn from Figures 5–8, we reach the following general conclusions.

(1) ET analyses, with diagonal dominant and positive definite sparse coefficient matrix, are of the locality property.

(2) Localized SOR algorithms are much faster than the traditional GSOR but with the same accuracy.

(3) LSOR2, the improved practical localized SOR algorithm, is of fastest convergence speed and shortest runtime.

(4) As for ET analyses of the locality property, LSOR2 can magnificently increase the solving speed with the same accuracy.

4 Apply the localized SOR analysis theory in ET analyses

This section will describe several ET analysis applications of the localized SOR analysis theory, including single-node statistical P/G analysis on voltage variations of over-IR-drop nodes [11], single-node statistical thermal analysis on temperature variations of hot spots [12], and efficient localized P/G analysis on single open-defect [13]. This section focuses on single-node statistical P/G analysis about voltage variations of over-IR-drop nodes, which shows superiorities of the localized SOR theory.

4.1 Single-node statistical P/G analysis method

With IC technology scaling into nanometer regime, process variation (PV) is so evident that it has to be considered in P/G design. Thus P/G analysis methods change from deterministic ones to statistical ones and some statistical P/G analysis methods are reported [2]. Global P/G analysis methods are of too high complexity to statistically analyze larger and larger scale practical P/G grids. Therefore ref. [11] only chooses problematic over-IR-drop nodes for efficient analyzing single-node voltage variation, which improves the efficiency of statistical P/G analysis. The single-node statistical P/G analysis method goes in the following four steps.

(1) Utilize G_0 and I_0, nominal value of G and I, to compute V_0, nominal value of V.

(2) Use V_0 to mark all problematic over-IR-drop nodes which compose the problematic node set.

(3) Use localized SOR algorithm to solve resistor vector R_q for the problematic node q.

(4) With the help of R_q, use conductor variation matrix ΔG and current variation vector ΔI to directly compute the standard voltage deviation for Δv_q, the voltage variation of q.

Among the four steps, it is most time-consuming for the localized SOR algorithm to solve resistor vector R_q, which means that efficient localized SOR algorithm is the key to improve the efficiency of the statistical method. In fact, R_q is the response vector V_q originating from the stimulus vector $I_q = [0, \ldots, 0, 1, 0, \ldots, 0]^{\mathrm{T}}$. The following formulation shows the relationship of I_q, V_q and R_q

$$V_q = G^{-1} I_q = R I_q = R_q, \tag{9}$$

where $R = G^{-1}$ is the resistor matrix. It is inefficient to use global ICCG [4] and SOR [5] algorithms to solve Eq. (9). Only one stimulus in I_q adds node q into relaxation set A in Step 1.3 of LSOR2, the length of A is one and LSOR2 is transformed into a single-node SOR algorithm.

GSOR, ICCG, and LSOR2 are used to statistically analyze the single-node voltage variation. Experimental P/G circuits are supplied by 31×31 PAD array, which cause 900 over-IR-drop nodes. Both current variations and conductance variations are spatially correlated. But current variations are independent of conductance variations. Variations of nodal currents follow the same normal distribution variations

$N(i\mu, i\sigma), i\sigma = 0.2i\mu$. And variations of conductance also follow the same normal distribution variations $N(g\mu, g\sigma), g\sigma = 0.2g\mu$.

The Mont-Carlo (MC) simulation method uses GSOR to simulate 5000 samples for obtaining standard voltage deviations (SVD) of 900 over-IR-drop nodes, which takes $T_{\rm sim}$ runtime. And then three analytic methods use GSOR, ICCG, and LSOR2 to solve R_q and directly solve SVDs of 900 over-IR-drop nodes, which takes $T_{\rm solv}$ runtime.

All accuracy data are collected in Table 1. With respect to small-scale circuits, MC simulation results are used as the golden to compute the relative errors of 900 nodal SVDs of three analytic methods so as to obtain the average error and maximum error. All three analytic methods are accurate enough because of little average errors (<1.5%) and maximum errors (<5%). Thus GSOR, ICCG, and LSOR2 are accurate enough to solve R_q and the analytic statistical method, using R_q, G, I to directly compute SVDs, is of theory correctness. Interested readers can find the detail of the analytic statistical method in [11]. For large-scale circuits, GSOR solving results are used as the golden to verify the accuracy of the other analytic methods. Since LSOR2 makes ignorable maximum errors (<0.03%), LSOR2 is as accurate as GSOR, proved by above mentioned theorems.

According to all efficiency data collected in Table 2, we can draw following conclusions.

(1) For small-scale circuits, all three analytic methods are faster than MC simulation method. LSOR2 can speed up 299 times and is much faster than GSOR and ICCG, which means that the localized algorithm is much more efficient in solving R_q than its global counterparts.

(2) For large-scale circuits, localized LSOR2 is 50 times faster than the global GSOR.

(3) LU-decomposition based ICCG is more efficient than iteration-based GSOR because the coefficient matrix is a five-diagonal sparse matrix. As for the sparser coefficient matrix, LU-decomposition based algorithm is more efficient than iteration-based one.

Combining conclusions drawn from Table 1 and Table 2, we can assert that LSOR2 can magnificently speed up single-node P/G analysis with the high accuracy.

In order to investigate the reason for high-efficiency of LSOR2, we increase the number of circuit blocks from 900 to 2500 and observe LSOR2 changes on speedup times, SOR iteration number ($N_{\rm it}$), the number of average effective relaxation nodes per iteration ($NN_{\rm SOR}$), and the relaxation scope. All data are listed in Table 3. As circuit blocks increase from 900 to 2500 in number, although $N_{\rm it}$ and $NN_{\rm SOR}$ change slightly, LSOR2 relaxation scope shrinks quickly and thus, the speedup times increase from 50.99 to 111.03.

Therefore, we can draw following conclusions.

(1) LSOR2 algorithm is of the locality property.

(2) The locality property of LSOR2 decides its algorithm efficiency and thus smaller the relaxation scope, faster the solving speed.

Random walk algorithm (RW) [7] is a single-node solving algorithm and has been used in single-node P/G analyses [7,8]. As two single-node algorithms, they are used to solve the expected values of nodal voltage variations for 900 problematic nodes, in which truncation error ε is assigned as 10^{-7} for LSOR2 while RW takes 10000 walks and the longest walk is pre-defined as the maximum distance in the circuit. We use ICCG to compute 900 expected values as the golden to obtain average errors and maximum errors of LSOR2 and RW. All data are listed in Table 4, in which $NNA_{\rm RW}$ is the number of RW's average random walking nodes per problematic node (PPN), $NNA_{\rm SOR}$ is the number of LSOR2's average effective relaxation nodes PPN, and $NNA_{\rm SOR}/NNA_{\rm RW}$ is the ratio. With data listed in Table 4, we can draw the following conclusions:

(1) LSOR2 is more efficient since it relaxes 18.47% nodes as RW walks.

(2) LSOR2 is more accurate because its maximum errors and average errors are only 0.188% and 0.165% while RW's maximum errors and average errors are 3.85% and 0.89%.

Thus LSOR2 is a better localized (single-node) analysis algorithm than RW algorithm.

Table 1 Accuracy comparison amid different algorithms

Nodes	Conductors	GSOR		ICCG		LSOR2	
(k)	(k)	Mean error	$E_{\max}$	Mean error	$E_{\max}$	Mean error	$E_{\max}$
89.64	180	0.96%	4.70%	0.96%	4.70%	0.96%	4.70%
202.44	405	0.47%	3.39%	0.47%	3.39%	0.47%	3.39%
360.24	720	1.44%	3.52%	1.44%	3.52%	1.45%	3.53%
563.04	1125	Base	Base	0.0015%	0.0021%	0.0068%	0.0085%
810.84	1620	Base	NA	0.0010%	0.0018%	0.0103%	0.0125%
1103.64	2205	Base	NA	0.0003%	0.0008%	0.0151%	0.0184%
1441.44	2880	Base	NA	0.0013%	0.0027%	0.0201%	0.0243%

Table 2 Efficiency comparison amid different algorithms

Nodes	Conductors	Simulation	GSOR		ICCG		LSOR2	
(k)	(k)	T_{sim}(s)	T_{solv}(s)	Accelerate	T_{solv}(s)	Accelerate	T_{solv}(s)	Accelerate
89.64	180	2396.6	336.6	7.12	307.9	7.78	21.20	113.00
202.44	405	17350.4	2641.3	6.57	1411.3	12.29	69.70	248.90
360.24	720	47422.5	7629.8	6.22	4344.7	10.91	158.36	299.50
563.04	1125	NA	15604.1	Base	9168.0	1.70	308.14	50.64
810.84	1620	NA	28523.7	Base	15288.3	1.87	559.38	50.99
1103.64	2205	NA	48496.9	Base	24157.7	2.01	1024.05	47.36
1441.44	2880	NA	69608.7	Base	35625.0	1.95	1758.47	39.58

Table 3 Reasons for LSOR2's high efficiency

Circuit	Nodes	Conductors	GSOR	LSOR2			
Blocks	(k)	(k)	N_{it}	Accelerate	N_{it}	$NN_{\text{SOR}}(k)$	Relaxation percentage
900	810.840	1620	147.71	50.99	151.60	41.03	5.060%
1225	1103.305	2205	147.87	67.97	150.83	42.05	3.811%
1600	1440.720	2880	147.55	77.78	151.48	41.12	2.854%
2025	1823.085	3645	147.87	89.09	151.47	43.28	2.374%
2500	2250.400	4500	147.02	111.03	152.74	35.65	1.584%

Table 4 Algorithm comparison between LSOR2 and RW

Nodes	Conductors	RW			LSOR2			
(k)	(k)	NNA_{RW}	Mean error	$E_{\max}$	NNA_{SOR}	Percentage	Mean error	$E_{\max}$
89.64	180	926.70	0.81%	2.90%	49.59	5.35%	0.105%	0.121%
202.44	405	2182.12	0.78%	3.14%	557.46	25.55%	0.023%	0.033%
360.24	720	3992.18	0.79%	2.97%	901.35	22.58%	0.038%	0.049%
563.04	1125	6328.24	0.87%	3.49%	1078.08	17.04%	0.069%	0.080%
810.84	1620	9238.06	0.86%	4.42%	1423.63	15.41%	0.437%	0.486%
1103.64	2205	12626.04	0.99%	4.62%	2559.78	20.27%	0.193%	0.221%
1441.44	2880	16581.23	1.10%	5.43%	3830.40	23.10%	0.289%	0.327%
			0.89%	3.85%		18.47%	0.165%	0.188%

4.2 Localized statistical thermal analysis method

Like in Subsection 4.1, GSOR, ICCG, LSOR2 are used to analytically solve the single-node temperature variations of hot spots in statistical 3D thermal analysis. Test cases are divided into 900 blocks in $X-Y$ planar similarly to [12]. Half blocks are impacted with weak power density stimuli and the other with strong stimuli. Weak and strong blocks are arranged into a chessboard array in which each strong block has four weak blocks as neighbors. Thus centers of 450 strong blocks are hot spots. Both power density

Table 5 Algorithm comparison on accuracy and efficiency

Nodes	Thermal conductor	GSOR		ICCG		LSOR2	
(k)	(k)	Accelerate	$E_{\max}$	Accelerate	$E_{\max}$	Accelerate	$E_{\max}$
220.5	613.21	23.75	2.32%	7.55	2.32%	371.5	2.42%
544.5	1518.01	24.44	1.67%	5.53	1.69%	745.3	1.66%
1012.5	2826.01	24.31	0.57%	5.49	0.57%	1109.6	0.55%
1624.5	4537.21	24.57	1.84%	5.38	1.83%	1403.2	1.83%
2380.5	6651.61	Base	Base	NA	NA	69.3	0.05%
3280.5	9169.21	Base	Base	NA	NA	77.2	0.05%
4324.5	12090.01	Base	Base	NA	NA	82.5	0.05%
4900.5	13701.61	Base	Base	NA	NA	90.1	0.05%

Table 6 Algorithm comparison on 1000 single open-defect P/G analyses

Nodes	Conductors	GSOR	LSOR2		
(k)	(k)	T_{solv}(s)	T_{solv}(s)	Accelerate	$E_{\max}$
89.64	180.60	117.970	9.22	21.84	0.002%
202.44	405.90	381.840	21.84	21.84	0.002%
360.24	721.20	1155.780	45.23	21.84	0.002%
563.04	1126.50	1767.141	77.36	22.84	0.002%
810.84	1621.80	3300.578	115.30	28.63	0.002%
1103.64	2207.10	7742.594	163.50	47.36	0.002%
1441.44	2882.40	9197.094	239.14	38.46	0.002%

variations and thermal conductance variations are spatially correlated. But power density variations are independent of thermal conductance variations. Variations of nodal power density follow the same normal distribution variations $N(i\mu, i\sigma), i\sigma = 0.2i\mu$. Variations of thermal conductance also follow the same normal distribution variations $N(g\mu, g\sigma), g\sigma = 0.2g\mu$.

Although we can obtain similar data shown in Table 1 and Table 2, we simplify the data and list them in Table 5. The Monte-Carlo (MC) simulation method uses GSOR to simulate 5000 samples for obtaining standard temperature deviations (STD) of 450 hot spots, which takes T_{sim} runtime. And then three analytic methods use GSOR, ICCG, and LSOR2 to solve Rq and directly solve STDs of 450 hot spots, which takes T_{solv} runtime. For small-scale cases, we use the equation $T_{\text{sim}}/T_{\text{solv}}$ to compute speedup times and use MC simulation results as the golden to obtain the maximum relative errors for three analytic methods. Due to smaller maximum errors ($\leqslant$2.42%), all analytic methods are accurate enough. Localized LSOR2 method has similar errors to those global methods such as GSOR and ICCG. For large-scale cases, we use GSOR solving results as the golden to obtain the maximum relative errors for LSOR2 results. The ignorable maximum errors ($\leqslant$0.05%) clearly show that LSOR2 is as accurate as GSOR, as proved by above mentioned theorems.

Efficiency data in Table 5 demonstrate the following facts.

(1) For small-scale cases, all three analytic methods are faster than MC simulation method. Localized LSOR2 can speed up 1403 times and is much faster than its global counterparts, GSOR and ICCG.

(2) For large-scale circuits, localized LSOR2 is 90 times faster than the global GSOR.

(3) LU-decomposition based ICCG becomes less efficient than iteration-based GSOR because the coefficient matrix is a seven-diagonal sparse matrix. As for the denser coefficient matrix, LU-decomposition based algorithm is less efficient than iteration-based one.

Combining conclusions drawn from Table 5, we can assert that LSOR2 can magnificently speed up single-node 3D thermal analysis with high accuracy.

4.3 Localized P/G analysis method on single open-defect

Similar to the single open-defect P/G analysis in [13], this work uses LSOR2 and GSOR in 1000 single open-defect P/G analyses. In order to guarantee the perfect locality of LSOR2, LSOR2's truncation

error ε_1 is set to 3ε , where ε is the GSOR's truncation error. All data are listed in Table 6. Compared with the global GSOR algorithm, localized LSOR2 algorithm can speed up 21–47 times with ignorable maximum errors ($\leqslant$0.02%). Thus LSOR2 can magnificently speed up single open-defect P/G analysis with high accuracy.

5 Conclusions

As for the ET analysis of sparse coefficient matrix, this work deduces the localized relaxation theory on ET analysis from the viewpoint of computing mathematics. It further asserts that the localized relaxation method relaxing a local part of a circuit can obtain the same accuracy as the global one. And a practical localized SOR algorithm named LSOR2 is proposed. Then LSOR2 is used to work out four realistic ET analysis problems. This shows that compared with the traditional global SOR algorithm (GSOR), LSOR2 can speed up 1–2 orders of magnitudes with nearly the same accuracy. From the viewpoint of theory, this work is an important supplement to the existing analysis theory on large-scale electric grids. From the viewpoint of practice, this work paves a way to effectively cut down the algorithm complexity for more and more complicated IC character analyses.

Acknowledgements

This work was supported by National High-Tech Research & Development Program of China (Grant No. 2009AA-01Z126), National Natural Science Foundation of China (Grant Nos. 60876025, 61076034, 61171014), and Fundamental Research Funds for the Central Universities.

References

1 The International Technology Roadmap for Semiconductors (ITRS), http://public.itrs.net/, 2008update
2 Luo Z Y. Power consumption and process variations: two challenges to design of next-generation ICs (in Chinese). J Comput, 2007, 30: 1054–1063
3 Luo Z Y. Survey and preview on studies of electro-thermal (ET) analysis (in Chinese). J Comput-aid design Comput Graph, 2009, 21: 1203–1211
4 Wu X H, Xian L H, Cai C, et al. Area minimization of power distribution network using efficient nonlinear programming techniques. IEEE Trans CAD, 2004, 23: 1086–1094
5 Zhong Y, Wong D F. Fast algorithms for IR drop analysis in large power grid. In: Proceedings of IEEE/ACM Int Conf Computer Aided Design. New York: ACM Press, San Jose, CA, 2005. 351–357
6 Cai Y C, Pan Z, Luo Z Y. Geometric multigrid based algorithm for transient RLC power/ground grid analysis (in Chinese). J Comput-aid Design Comput Graph, 2005, 17: 33–38
7 Qian H F, Nassif R, Sapatnekar S S. Random walks in a supply Network. In: Proceedings of IEEE/ACM Design Automation Conf. New York: ACM Press, Anaheim, CA, 2003. 93–98
8 Li P. Variational analysis of large power grids by exploring statistical sampling sharing and spatial locality. In: Proceedings of IEEE/ACM Int Conf Computer Aided Design. New York: ACM Press, San Jose, CA, 2005. 644–650
9 Luo Z Y, Cai Y C, Tan X D, et al. Time-domain analysis methodology for large-scale RLC circuits and its applications. Sci China Ser F-Inf Sci, 2006, 49: 665–680
10 Luo Z Y, Tan X D. Efficient statistical analysis method of power/ground(P/G) network. Prog Nat Sci, 2008, 18: 189–196
11 Luo Z Y, Tan X D. Statistic analysis of power/ground networks using single-node SOR method. In: Proceedings of ISQED'08, San Jose, CA, USA, 2008. 867–872
12 Luo Z Y, Fan J, Tan X D. Localized statistical 3D thermal analysis considering electro-thermal coupling. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'09), Taipei, 2009. 1289–1292
13 Luo Z Y, Zhang Y B, Yu X C. Single open-defect analysis method for power/ground networks (in Chinese). J Comput R&D, 2009, 46: 1234–1240
14 Wozniakowski H. Round-off error analysis of iterations for large linear systems. Numer Math, 1978, 30: 301–314
15 Lynn M S. On the round-off error in the method of successive over-relaxation. Math Comput, 1964, 18: 36–49