

Power Grid Simulation Via Efficient Sampling-Based Sensitivity Analysis and Hierarchical Symbolic Relaxation

Peng Li

Department of Electrical Engineering, Texas A&M University
College Station, TX 77843, USA

pli@neo.tamu.edu

ABSTRACT

On-chip supply networks are playing an increasingly important role for modern nanometer-scale designs. However, the ever growing sizes of power grids make the analysis problem extremely difficult thereby introducing severe challenges in design and optimization. The inherent analysis complexity calls for innovations in simulation techniques that must provide appropriate accuracy, efficiency as well as the tradeoff thereof to aid design verification and optimization. In this paper, we first present a sampling-based sensitivity analysis by employing the notation of *importance sampling* in a Monte Carlo based circuit simulation framework. This technique allows the extraction of multi-parameter sensitivities for the node voltages of interest in the same Monte Carlo runs that are used for computing the nominal voltage values. For more efficient nonstructured whole-grid solution approaches, we further introduce a new *direct* solution method by embedding symbolic relaxation steps in a hierarchical fashion. As a direct method, the proposed hierarchical symbolic relaxation is suitable to both dc and transient analyses. Circuit examples are included to demonstrate the efficacy of the proposed techniques.

Categories and Subject Descriptors: B.7.2

General Terms: algorithms, performance, design

Keywords: Power grids, sensitivity and hierarchical analysis.

1. INTRODUCTION

In recent years, the technology trend has put substantial pressure on on-chip power delivery network design. The proper design of the power grid is critical for minimizing the supply induced noise and meeting reliability constraints. With the continuous shrinkage of the supply voltage level and the increasing power consumption, it is expected that ensuring the integrity of the power distribution network is becoming even more challenging for future designs [15].

At present, the designers are routinely dealing with multi-million node power distribution networks. The sheer size of these distribution networks brings significant challenges in the verification and optimization tasks. In the past several years, power distribution networks have been an active topic of research. Both the power grid analysis [1-5, 7-8] and optimization issues [10, 12-15] are being addressed by the EDA community. In terms of analysis, given the fact that the conventional direct solution methods do not scale with the problem size, it does not only make sense, but also becomes imperative to develop specific simulation techniques that can offer a good tradeoff between accuracy and efficiency. The latest developments pursue this goal through a

variety of approaches, for instance, multi-grid like approaches [1-2, 4], hierarchical method [3], Monte Carlo approach [5, 7] and design specific methodology [8].

Primarily motivated by the techniques pioneered in [1, 2, 3, 5], we present two new techniques along the same line of addressing the analysis scalability. In the first technique, we extend the principle of random walks in [5] to an efficient sensitivity analysis methodology. By deploying one particular type of *importance sampling* technique, namely *ratio estimate* [16], we develop a sampling based sensitivity analysis technique suitable for computing sensitivities with respect to multiple design or process parameters for a few circuit nodes of interest. With a small additional cost, the sensitivity information can be extracted in the same Monte Carlo runs used for estimating nominal voltage values, thereby offering useful aides for design fine tuning or process variation analysis.

To derive a more effective solution method for the entire grid, which can be efficient for both dc and transient analysis, we present a new *direct* hierarchical method. Our focus is on unstructured grids for which hierarchical information of the design is not available to simplify the analysis. To this end, we start by describing random walks technique more formally via finite Markov chains [17]. More importantly, we show a revealing correspondence between the solution of finite Markov chains with absorbing states and the standard deterministic relaxation methods. Under the context of hierarchical methods for nonstructured grids, we show that the seemingly most efficient *hierarchical Monte Carlo* method is in fact in the form of a *deterministic* relaxation method, which is what we propose as *hierarchical symbolic relaxation*. To understand the observed effectiveness of the proposed hierarchical method despite its simple form, we interpret its interesting properties under the context of multigrid methods and contrast it against the multigrid-like reduction techniques introduced in [1, 2, 4]. The usage of proposed techniques is demonstrated on several power grid analysis problems.

2. SAMPLING BASED SENSITIVITY ANALYSIS

In [4], a power grid, which is modelled as a RC network is translated to a random walk. Due to the correspondence between the electric network (grid) and the random walk, the electronic network solution can be equivalently solved in random walk in terms of the *expectation* of a *gain* function. Formally, a random walk can be described as a finite Markov chain. There exists a one-to-one correspondence between the nodes in the electrical network and the states in the Markov chain. As shown in Fig. 1(b), the corresponding Markov chain for the circuit in Fig. 1(a) is shown. Throughout this paper, we will use the terminologies for the electrical network and the Markov chain interchangeably. The gain function has its boundary conditions defined on the states corresponding to V_{dd} nodes in the grid, which have a *known* gain function value of V_{dd} . These states are absorbing states in the chain.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.
Copyright 2005 ACM 1-59593-058-2/05/0006...\$5.00.

For a dc analysis problem, consider node 1 in Fig. 1(a). The transition probabilities from node 1 to its neighbours in the chain are determined by circuit element values [5]

$$p_{1,2} = G_1 / \sum_{i=1}^3 G_i, p_{1,3} = G_2 / \sum_{i=1}^3 G_i, p_{1,4} = G_3 / \sum_{i=1}^3 G_i \quad (1)$$

The current source I_1 is translated to a amount of cost incurred at node 1, whose value is given by $C_1 = -I_1 / (G_1 + G_2 + G_3)$. The walker wanders about in the network and picks the next state to visit randomly based on the transition probabilities. If he reaches an absorbing state, the walk terminates and he is paid by an amount of V_{dd} . Otherwise, if he passes by some other states he may pay for a cost. It can be shown that the voltage at any node i in the electrical network is equal to the expected gain the walker will receive if he starts from the corresponding node in the random walk. Thus, the node voltage can be estimated by running N walks from node i as

$$V_i = 1/N \cdot \sum_{k=1}^N D_k, \quad (2)$$

where D_k is the gain of the k th walk.

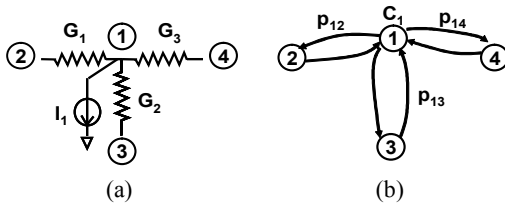


Fig. 1. (a) An equivalent Markov chain model of a RC power grid, and (b) one circuit node in the grid.

In addition to the nominal voltage value at a node, we may also want to compute the voltage's sensitivities w.r.t circuit parameters, e.g., in order to aid the gradient based design optimization [14] or process variation analysis. One question arises naturally is that whether or not we can still the random walk in an efficient way to collect the sensitivity information as we compute the nominal voltages. It shall be noted that the classical adjoint sensitivity analysis [11] depends the complete linear system solution of the system matrix that is what we try to avoid computing.

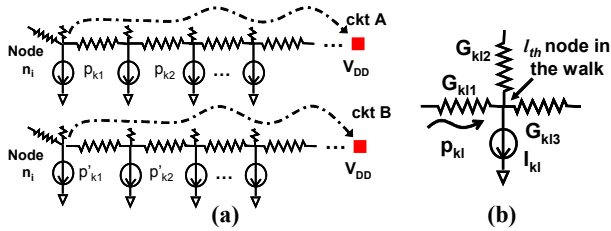


Fig. 2 (a) Two RC networks with the same topology, and (b) one node along the walk

Consider two RC networks with an identical topology but differ in circuit element parameters, as shown in Fig. 2 (a). Let us consider one arbitrary walk in circuit A. The edge transition probabilities along the walk are $p_{k1}, p_{k2}, \dots, p_{km_k}$. Thus, the probability of walking down this complete path is given by

$$p_k = \prod_{i=1}^{m_k} p_{ki}. \text{ It is possible to pursue the same walk in circuit B, but}$$

with a different probability $p'_k = \prod_{i=1}^{m_k} p'_{ki}$. The voltage at node i is the weighted sum of gains over all possible walks

$$V_i = E(D_k) = \sum_{k=0}^{\infty} p_k D_k \quad (3)$$

Similarly for circuit B, we have

$$V'_i = E(D'_k) = \sum_{k=0}^{\infty} p'_k D'_k = \sum_{k=0}^{\infty} p_k (p'_k D'_k / p_k) \quad (4)$$

The above expression suggests that we may reuse the random walks in circuit A to compute the same voltage in circuit B, but with a proper scaling of the gain obtained for each walk. Fortunately enough, the above observation has a theoretical soundness that is rooted in the concept of *importance sampling* from the statistical analysis [16]. Importance sampling has been applied to speed up the convergence of Monte Carlo simulation and collect statistics efficiently for multiple distributions. In this paper, we use one particular form of importance sampling, namely *ratio estimate*. Applying the ratio estimate to (4) based upon a set of N samples (random walks) taken in circuit A, we estimate the node voltage of node i in the circuit B as

$$\hat{V}'_i = \frac{\sum_{k=1}^N W_k D'_k}{\sum_{k=1}^N W_k}, W_k = \frac{p'_k}{p_k}, \quad (5)$$

where D'_k is the gain (defined in circuit B) for the k th walk, which is actually sampled in circuit A, and W_k is a correction factor for the k th walk to compensate for the bias introduced by taking samples from circuit A. In (5), dividing the weighted sum by the summation of W_k 's is to ensure that the final weights of these N samples sum up to unity.

Now it becomes clear that we can in fact use Monte Carlo runs performed in one particular network to conduct estimations on several other networks with the same topology. We now show how this nice property of importance sampling can be adopted in a proper fashion to facilitate an efficient sensitivity analysis for practical large circuit problems. Going back to our previous circuit example in Fig 2 (a), but this time we treat circuit A as a parametric circuit under the influence of s parameters $\bar{\rho} = [\rho_1, \rho_2, \dots, \rho_s]^T$, which might be a set of process or design parameters. Now our goal is to reason about the first order parametric circuit variations based upon samples we take from the nominal circuit, where $\bar{\rho} = \bar{\rho}_0$. After taking N random walks on the nominal circuit, we apply the ratio estimate of (4) to estimate the node voltage of the parametric network as

$$\hat{V}'_i(\bar{\rho}) = \frac{\sum_{k=1}^N W_k(\bar{\rho}) D'_k(\bar{\rho})}{\sum_{k=1}^N W_k(\bar{\rho})}, W_k = \frac{p_k(\bar{\rho})}{p_k(\bar{\rho}_0)}, \quad (6)$$

where, $D'_k(\bar{\rho})$ is the gain of the k th walk defined in the parametric circuit, $p_k(\bar{\rho})$ and $p_k(\bar{\rho}_0)$ are the probabilities of taking k th walk in the parametric and nominal circuits, respectively.

Next we want to show that sensitivity computation simply amounts to propagate and process the first order expansion coefficients in $\bar{\rho}$ for all the terms in (6) as the walker wanders about in the nominal network. To do so, we first need to consider the sensitivities of the circuit elements w.r.t. $\bar{\rho}$ and propagate

them to all the terms we encounter in (6). As an example, consider the l -th node reached in the k th random walk as depicted in Fig. 2 (b). Now we consider the next move the walker will take to reach the $(l+1)$ -th node. Assume that the walker takes the edge corresponding to conductance G_{kl1} in the next move. The probability for such an instance to happen is

$$p_{k,l+1}(\bar{\rho}) = \frac{G_{kl1}(\bar{\rho})}{\sum_t G_{kl,t}(\bar{\rho})} \approx \frac{G_{kl1}(\bar{\rho}_0) + G_{kl1}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}}{\sum_t G_{kl,t}(\bar{\rho}_0) + G_{kl,t}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}} \quad (7)$$

where $G_{kl,t}'$ is the sensitivity of conductance $G_{kl,t}$ connecting at the node. Finally, we have

$$\begin{aligned} p_{k,l+1}(\bar{\rho}) &= \frac{G_{kl1}(\bar{\rho}_0)[1 + G_{kl1}'(\bar{\rho}_0)^T / G_{kl1}(\bar{\rho}_0) \cdot \Delta\bar{\rho}]}{\sum_t G_{kl,t}(\bar{\rho}_0) \left[1 + \left(\sum_t G_{kl,t}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho} \right) / \sum_t G_{kl,t}(\bar{\rho}_0) \right]} \\ &\approx p_{kl}(\bar{\rho}_0) \left(\frac{1 + G_{kl1}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho} / G_{kl1}(\bar{\rho}_0) - \left(\sum_t G_{kl,t}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho} \right) / \sum_t G_{kl,t}(\bar{\rho}_0)}{\left(\sum_t G_{kl,t}'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho} \right) / \sum_t G_{kl,t}(\bar{\rho}_0)} \right) \\ &= p_{kl}(\bar{\rho}_0) \left(1 + \left(\frac{G_{kl1}'(\bar{\rho}_0)^T / G_{kl1}(\bar{\rho}_0) - \left(\sum_t G_{kl,t}'(\bar{\rho}_0)^T \right) / \sum_t G_{kl,t}(\bar{\rho}_0)}{\left(\sum_t G_{kl,t}'(\bar{\rho}_0)^T \right) / \sum_t G_{kl,t}(\bar{\rho}_0)} \right) \cdot \Delta\bar{\rho} \right) \end{aligned} \quad (8)$$

In (8), we have expressed the first order expansion of $p_{kl}(\bar{\rho})$ in $\bar{\rho}$ using the nominal probability, (scaled) additions and subtractions of circuit element (conductance) sensitivities. The first order expansion for the cost incurred at this point of random walk due to current I_{kl} can be similarly derived.

Now assume that all the transition probabilities and costs can be expressed in a unifying first order expansion form $v(\bar{\rho}) = v(\bar{\rho}_0) + v'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}$. To continue processing the first order expansions in (7), in addition to trivial additions or subtractions cast in the unifying standard first order form, we also need to consider multiplications and divisions, which can be processed easily. For instance, the division of the two first order terms simply leads to

$$\begin{aligned} \frac{v_1(\bar{\rho})}{v_2(\bar{\rho})} &= \frac{v_1(\bar{\rho}_0) + v_1'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}}{v_2(\bar{\rho}_0) + v_2'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}} = \frac{v_1(\bar{\rho}_0) + v_1'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho}}{v_2(\bar{\rho}_0) \left(1 + v_2'(\bar{\rho}_0)^T \cdot \Delta\bar{\rho} / v_2(\bar{\rho}_0) \right)} \\ &\approx \frac{v_1(\bar{\rho}_0)}{v_2(\bar{\rho}_0)} + \left(\frac{v_1'(\bar{\rho}_0)^T}{v_2(\bar{\rho}_0)} - \frac{v_1(\bar{\rho}_0)v_2'(\bar{\rho}_0)^T}{(v_2(\bar{\rho}_0))^2} \right) \Delta\bar{\rho} \end{aligned} \quad (9)$$

All that means is that we can efficiently collect the sensitivities of node voltages of interest with respect to multiple parameters along the same set of random walks we take to compute the nominal voltages. The additional costs at each step incurred for sensitivity computation are only due to a few simple scalar additions, subtractions, multiplications and divisions. That is, we are able to compute the nominal voltage values as well as multiple parametric sensitivities at a computational cost just somewhat higher than that of the nominal voltage computation.

3. HIERARCHICAL METHODOLOGY

For the solution of a complete unstructured grid where design hierarchy is difficult to obtain, applying Monte Carlo methods can become inefficient. For these cases, there are two *orthogonal* reasons that limit the efficiency of Monte Carlo methods. Firstly, for a large grid where there exist a few V_{dd} connections (bond-wired chips), the probability for a walk to terminate at an absorbing state within a few steps is very small. This situation will lead to time consuming long walks. In [7], the authors have proposed a very interesting idea to alleviate this problem by

adopting hierarchical random walks. However, the second problem still exists that is more inherent to the nature of Monte Carlo methods, i.e., the number of samples needed for an estimate with a high confidence level may remain high even for a small problem. For instance, solving a *three-node* circuit problem using random walks may still require hundreds or thousands of samples to ensure a guaranteed accuracy while solving it using any deterministic method is completely trivial.

We seek a different and more efficient deterministic hierarchical method for solving the entire grid. In addition, the approach falls into the class of direction solution methods such that it can be employed efficiently for transient analysis, where significant runtime saving is made possible by reusing the initial matrix factorization for the subsequent time steps.

We illustrate the construction of our hierarchical methodology in Fig. 3. Starting from the original network, which is at the bottom of our solution hierarchy, we select a certain percentage of nodes as *high-level* nodes which we keep at the next hierarchical level. We refer to other nodes as *internal* nodes. Since we do not assume the availability of any design hierarchical information, the *high-level* nodes are selected without knowing any circuit boundaries. Once high-level nodes are selected, we build a *reduced* network relating only these nodes by symbolically solving internal nodes in terms of high-level nodes and the right hand side, and finally eliminate all the internal nodes in the reduced network. The last operation is achieved by writing the KCL equation at each individual high-level node and substituting in the symbolic expressions of its neighbouring internal nodes. In this process, we also generate new right hand side to the reduced network which is a linear mapping of the original right hand. We keep these mappings for later use. This process repeats until we reach to a point where the matrix problem size is small enough to be factorized either iteratively or directly. To solve each of these problems symbolically, we adopt symbolic Gauss-Jacobi or Seidel iterations hierarchically as a solution engine.

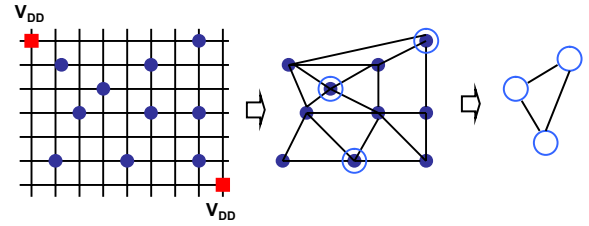


Fig 3. Hierarchical methodology

After the factorization phase, to solve the actual matrix solution for a given right hand side, we first map the right hand at the bottom level bottom-up to the top level by using the mapping functions already built in the factorization phase for each of these levels. We substitute the mapped right hand side into the factorized reduced matrix and obtain the numerical solution at the topmost level. Finally, we repeatedly propagate the solution of high-level nodes downwards and obtain the solution at the hierarchy one-level down. This process repeats till we reach back to the bottom of hierarchy where the complete system solution is obtained.

4. SYMBOLIC FACTORIZATION

To obtain an understanding of the proposed technique, we first revisit the Markov chain interpretation of power grid analysis problem, and then offer an interpretation of our proposed hierarchical symbolic relaxation approach using Markov chain. Finally, we interpret the approach under the context of multigrid.

4.1 Markov chain interpretation

Let us consider to use the random walk based hierarchical method such as what is outlined in [7] for unstructured grids, where design hierarchical information is not available. Recall that in the standard random walk algorithm, an unknown node is selected at a time, followed by a sequence of random walks. We illustrate this sequence of operations in Fig. 4(a). One drawback

of this approach is that it precludes any work sharing between separate walks even though they might visit the same nodes along the way. It now becomes very natural to think of the possibility of simultaneously starting multiple walks, possibly originated from multiple starting nodes such that the maximum amount of work sharing can be allowed. This motivates us to shift from the depth-first like “serial” walks of 4(a) to the breadth-first like “parallel” walks of Fig. 4(b).

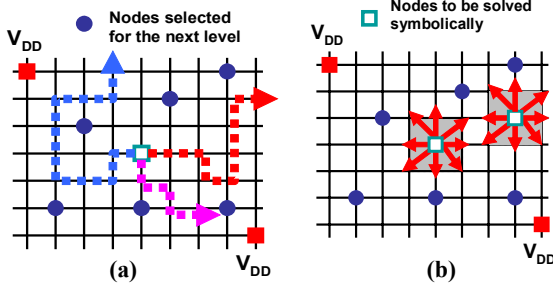


Fig. 4. “Serial” walks vs. “parallel” walks.

To explore this possibility, we employ the formal description of finite Markov chains with absorbing states [17] for the hierarchical analysis context under which high-level nodes are also absorbing states

$$f = \begin{bmatrix} f_B \\ f_I \end{bmatrix} = Pf + c = \begin{bmatrix} I & 0 \\ R & Q \end{bmatrix} \cdot \begin{bmatrix} f_{B0} \\ f_I \end{bmatrix} + \begin{bmatrix} 0 \\ c_I \end{bmatrix}, \quad (10)$$

where f is a function defined on the states of the chain which we want to solve (corresponding to node voltages for our power grid), P is the transition or fundamental matrix whose entries define the transition probabilities of a pair of states and c is the cost incurred at each state. Notice that the entries in the above equation can be determined in a way as illustrated in Fig. 1. In (10), f is partitioned as follows. f_B corresponds to the absorbing states whose function values are known (corresponding to V_{dd} nodes or high-level nodes for our case), and f_I corresponds to all internal nodes. We refer the readers to [17] for a more detailed treatment of finite Markov chains. Notice that (10) is slightly different from the standard form since we have also included expense function c to take into account the current sources attached at internal nodes [5]. The objective here is to solve for f_I given f_B . From (10), we have

$$f_I = (1 - Q)^{-1} (Rf_{B0} + c_I) = \sum_{i=0}^{\infty} Q^i (Rf_{B0} + c_I) \quad (11)$$

Keeping only first k terms in the summation of (11) leads to

$$f_I = (1 - Q)^{-1} (Rf_{B0} + c_I) \approx \sum_{i=0}^{k-1} Q^i Rf_{B0} + \sum_{k=0}^{k-1} Q^i c_I \quad (12)$$

Now (12) can be very intuitively understood by thinking of starting parallel “walks” from every internal node simultaneously. Here, these walks are different from the random walks. They are sequences of state transitions with known probabilities. Thus, they are dispatched in a *deterministic* way without relying on taking statistical samples to discover the known probabilities. Now assume that we only allow these parallel walks to go on for a depth less than k . Then some of them will terminate at an absorbing state some will not. The first term at the right hand side of (12) represents those walks terminated within $k-1$ steps and the second term represents the amount of cost accumulated by all the walks in the first $k-1$ steps. Those walks which cannot terminate are

discarded, which represent the truncation error of (12). Restricting the depth of walks to $k-1$ (k is small) is in fact very reasonable for our hierarchical power grid analysis when there are *many* high-level nodes at each hierarchical level. This is because that the nearby internal nodes are not likely to be influenced significantly by far way high-level nodes. Obviously, for any reasonable power grid we know that the voltage of any node should not be dramatically different from its neighboring nodes. To reduce the truncation error of (12), it is better to treat those non-terminating walks as if they were ended at their nearest absorbing states at the last stop. This leads to a more accurate approximation

$$f_I = (1 - Q)^{-1} (Rf_{B0} + c_I) \approx \sum_{i=0}^{k-1} Q^i Rf_{B0} + \sum_{i=0}^{k-1} Q^i c_I + Q^{k-1} f_{app} \quad (13)$$

In (13), f_{app} is the vector of absorbing states used to force the lingering walks to stop at the nearest absorbing states.

4.2 Symbolic relaxation

Interestingly enough, (13) can be used for solving the system but in a way where no matrix is explicitly formed. We can show that (13) is equivalent to the following symbolic Gauss-Jacobi iteration relaxation that is what we are proposing to use as a multi-level engine:

- 1 Set $i \leftarrow 0$ and initial condition $f_I^i \leftarrow f_{app}$
- 2 while $i < k$,
 - 2.1 for each internal node j ,
 - 2.1.1 $f_j^{k+1} \leftarrow \sum_{l, p_{jl} > 0} p_{jl} f_l^k + c_j$;
 - 2.1.2 $k++$;
 - 2.2 end j ;
- 3 end while;

The above algorithm can be easily derived if we rewrite (11) in the standard GJ iteration form and consider f_{app} as the initial guess. For N_I internal nodes, the above symbolic GJ takes $O(kN_I)$ operations to complete. According to the interpretation of (13) in the prior section, the above deterministic procedure is expected to be much more efficient than standard random walks since: a) as we propagate parallel walks from all these internal nodes to their neighbors we have weighted them according to the exact probabilities as opposed to using a large number of randomly generated samples to discover these weights in a statistical manner; b) any amount of work common to any two walks is fully reused as we sequentially relax all the internal nodes in the GJ iterations. This implies that for this particular case, the *most* efficient way of solving the problem *statistically* is actually in the form of *deterministic* relaxation methods.

This result might not be entirely surprising since the deterministic linear system solutions are often applied to analyze probabilistic Markov chains. We should also notice that the important differences between the above approach and the typical way of applying relaxation methods. Under the hierarchical solution framework, the above procedure is symbolic, i.e., symbolic solutions in terms of high-level nodes and right hand side are solved for internal nodes. Additionally, we have set the symbolic initial conditions for all internal nodes according to their nearest high-level nodes neighbors f_{app} .

In our implementation, we have used both symbolic GS and GJ iterations. As a short note, with proper handling, we can guarantee the positive definiteness property of the matrix problem across all hierarchical levels. Thus, there is a theoretical soundness in applying Gauss-Jacobi or Seidel iterations. Same as in a traditional

relaxation method, a good initial condition will speedup the convergence for the symbolic relaxation method. This is achieved by setting the (symbolic) initial value of each internal node by its nearest high-level node neighbor. To better handle nonuniform grids, at any hierarchical level, we adopt a shortest path based heuristic to assign each internal node to its “nearest” high-level neighbor, where the distance is measured in terms of “electrical connectivity” instead of geometric distance. This strategy tries to setup the initial values for internal nodes according to their most strongly connected high-level node. Our experiments have proven the effectiveness of the approach.

4.3 Connection to multigrid

The efficiency of classical multigrid methods is built upon a proper understanding of the properties of relaxation methods such as Gauss Jacobi and Gauss Seidel. That is, they are quite efficient in removing spatial high-frequency solution errors while being ineffective for damping long-range low-frequency errors. To overcome this difficulty, in multigrid high-frequency errors are removed at fine grids while low-frequency errors are damped more effectively at coarser grids. A proper interaction between fine and coarse grids offers the ultimate efficiency of method, which is provided by performing relaxation steps in between different grids.

When applying multigrid like techniques to solve the power grid problems, a modification from the classical multigrid approach was introduced to better serve the specific need of circuit analysis [1, 2, 4]. That is, relaxation steps are removed to make the overall approach a direct solution method, which would be very beneficial for transient analysis. Neglecting the relaxation steps is based on the assumption that for well designed power grids the solution errors are smooth. In a sense, these approaches are reduction methods in which strongly connected nodes are continuously removed or collapsed to reduce the problem size. However, as more and more nodes are removed or collapsed, long-range low frequency supply variation starts to emerge as high-frequency variation such that any further reduction of the grid will artificially filter out the true supply level variation leading to inaccurate results. This situation is illustrated in Fig. 5 (a). In our experiments, we have observed that the grid reduction method of [4] performs fairly well in terms of both accuracy and runtime for most of cases, but for grids with more varying current loads, it tends to produce larger errors. Removing smoothing steps can also limit the extent to which a grid can be reduced. The reduction must be stopped before it starts to smooth out the true high-frequency supply voltage variations. For a large grid, this implies that a direct solver is forced to be applied to a relatively large reduced network at the bottom of hierarchy.

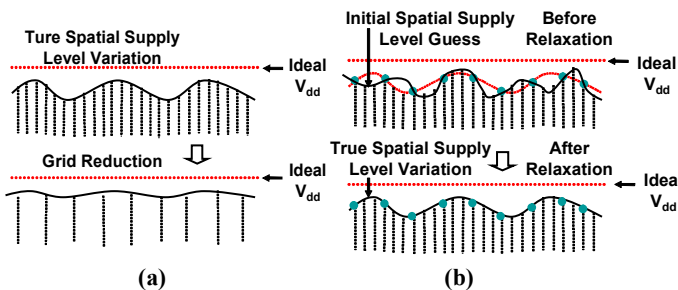


Fig. 5. (a) Grid reduction, and (b) symbolic relaxation.

The hierarchical symbolic relaxation approach is more like a problem-specific, controllable, approximate multi-level matrix factorization method. Symbolic relaxations are only employed to achieve the goal of building such a direct method. Although Gauss Jacobi or Seidel is known to be inefficient in removing low frequency solution errors, however, it is not an issue under our hierarchical approach. Notice that different from the standard multigrid method where relaxations steps are applied numerically to damp high frequency error for a given input excitation, in our approach relaxations are applied symbolically in order to solve all internal nodes in terms of *exact* symbolic values of high-level nodes and right hand side. Since during the iteration these symbols

are not committed to any numerical values, they are exact and do *not* contain any low-frequency errors. As shown in Fig 5 (b), any error in the symbolic initial guess of internal nodes would in fact appear to be of high frequency, thus can be removed quickly by a few relaxation steps. This explains why our symbolic Gauss Jacobi and Seidel iterations can work very well. But there is a price to pay. Symbolic relaxations operations need to be implemented based on row operations of sparse matrix data structures and are more expensive to apply than their numerical counterpart.

As a side note, avoiding the iterative nature of classical multigrid method to suit our circuit specific analysis need (e.g. transient analysis) also implies that the resulting direct method will not operate on the basic mechanism on which the classical multigrid methods reply. Therefore, it makes sense to directly consider more generally about constructing controllable, approximate, direct matrix factorization method which might also be hierarchical. In a sense, this objective is pursued in our approach by leveraging on problem-specific knowledge, e.g., exploring positive definiteness of the problem property (using GS/GJ to derive controllable approximate matrix factorization).

5. RESULTS

The proposed techniques were implemented in a prototyped circuit simulator SCE (*Scalable Circuit Emulator*). All our experiments were executed on a Pentium 4 PC with 2GB memory running Linux operating system.

5.1 Sensitivity analysis

For the grids shown in Table 1, we apply the importance sampling based random walks to estimate the nominal node voltage of one output node and its sensitivities to several parameters. For the first three grids, we compute sensitivities with respect to 20 parameters while for the last two grids 10 sensitivities are considered. The 2nd column of the table indicates the runtime of the direct sensitivity analysis (direct LU factorization), the 3^d column shows the runtime for computing the nominal node voltage without estimating sensitivities using random walks, the fourth column is the importance sampling based sensitivities analysis and the last two columns are relative errors for the estimations of the nominal node voltages and sensitivities. The last two grids are too large to apply the direct method. Compared to the direct sensitivity analysis, by avoiding the full matrix factorization, the proposed sensitivity analysis offers very accurate sensitivity estimates while incurring very low runtimes. It can be also seen that processing additional multi-parameter sensitivity information while performing random walks does not increase the runtime significantly. This means that by spending somewhat more time, both the nominal node voltage and its sensitivities can be obtained.

Table 1. Comparison on sensitivity analyses

# Nodes	Direct Sens.	Nom. R.W.	Imp. Samp.	Nom. * Error	Sens. * Error
40K	87s	<1s	1	0.4%	3.53%
90K	381s	5s	16	0.1%	9.72%
250K	43m	6s	18	0.0%	4.23%
1.1M	N/A	8s	16	---	---
1.44M	N/A	14s	19	---	---

* Normalized with the largest IR drop in the grid.

5.2 DC analysis results

In Table 2, we compare the proposed hierarchical symbolic relaxation approach with our implementation of the grid reduction approach RAMG in [4]. For the first two smaller designs, we use direct solve as the comparison reference while the other two grids are too large to apply the direct solve. Notice that in this example, the relative errors in the last two columns are obtained by normalizing the absolute voltage errors with respect to the largest IR drop in the grid. In Fig. 6, we compare the error distributions among all the nodes for these two methods. For both cases, the proposed technique appears to be more accurate. In term of runtime, the two methods are comparable for smaller grids.

However, for larger examples, our method actually outperforms. This is because that the method can generate very small reduced system at the topmost level without incurring too much error.

5.3 Transient analysis results

We applied our transient analysis on four grids as shown in Table 3. For each of these cases, we perform two matrix factorizations – one for DC solution and the other for transient analysis, followed by 40 steps of transient simulation using a fixed time step. For the first two grids in the table, the direct solve is applicable. We randomly choose the time-domain waveforms of 60 circuit nodes and list the corresponding average and max relative errors in the table. In Fig. 7, we also compare the time domain waveforms computed by the proposed technique against those of the direct solve. As can be seen, our results are fairly accurate. For the two largest grids to which a direct solve becomes too difficult to apply, the proposed technique is still able to complete the simulation in a reasonable amount of time.

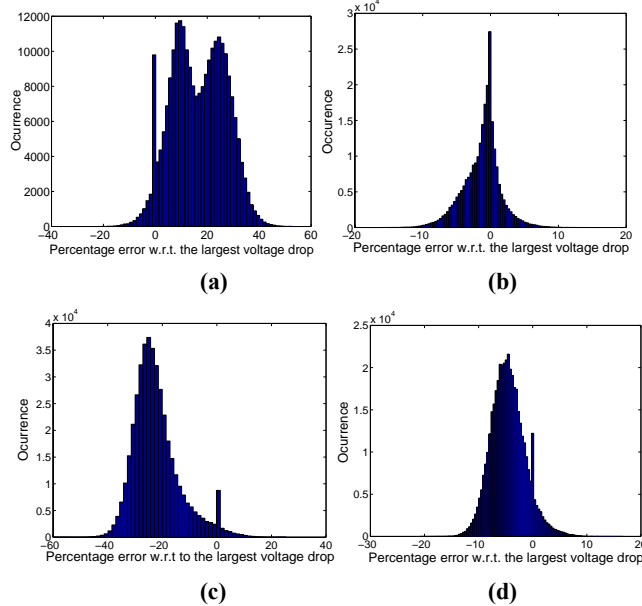


Fig. 6. Node voltage error distributions (a) RAMG (0.25M), (b) SCE(0.25M), (c) RAMG(0.42M) and (d) SCE(0.42M).

Table 2. Comparison on DC analysis

# of Nodes	Max IR Drop	Direct CPU Time	RAMG [4]		SCE	
			Ave. Error	CPU Time	Ave. Error	CPU Time
0.25M	73mv	71m	16.8%	19s	2.16%	13s
0.42M	51mv	14h *	22.3%	82s	4.83%	32s
1.1M	---	---	---	390s	---	56s
1.44M	---	---	---	450s	---	79s

* Executed on a somewhat overloaded Sun Ultra Spare workstation.

6. CONCLUSION

We have presented two techniques for fast power grid simulation. Firstly, we have exploited importance sampling to derive an efficient sampling based sensitivity analysis. The presented analysis retains the nice property of the Monte Carlo method such that the required sensitivities are obtained without solving the entire system. Furthermore, a symbolic relaxation

based hierarchical method is presented for efficiently computing both the DC and transient responses for the entire grid. Experiments have confirmed the efficacy of both techniques.

Table 3. Comparison on transient analysis

# of Nodes	Direct Solve CPU Time	SCE			
		Total CPU Time	CPU Time /Step	Ave Error	Max Error
0.25M	210m	1m30s	0.7s	0.10%	0.55%
0.42M	599m	2m4s	1.8s	0.16%	0.57%
1.1M	---	3m19s	2.3s	---	---
1.44M	---	4m37s	2.4s	---	---

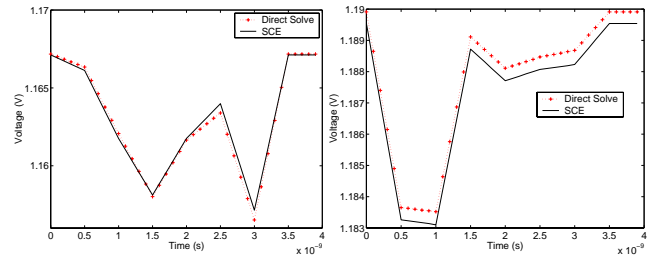


Fig. 7. Transient simulation of the 0.25M node grid.

7. ACKNOWLEDGMENTS

This work is supported in part by the SRC contract 2005-TJ-1298.

REFERENCES

- [1] S. Nassif and J. Kozhaya, "Fast power grid simulation," *IEEE/ACM DAC*, 2000.
- [2] J. Kozhaya, S. Nassif and F. Najm, "A multi-grid like technique for power grid analysis," *IEEE Trans. CAD*, vol. 21, no. 10, Oct. 2002.
- [3] M. Zhao, R. Panda, S. Sapatnekar and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE Trans. on CAD*, vol. 21, no. 2, Feb. 2002.
- [4] H. Su, E. Acar and S. Nassif, "Power grid reduction based on algebraic multigrid principles," *Proc. IEEE/ACM DAC*, 2003.
- [5] H. Qian, S. Nassif and S. Sapatnekar, "Random walks in supply network," *Proc. of IEEE/ACM DAC*, 2003.
- [6] R. Rubinstein, "Simulation and Monte Carlo Method," *John Wiley & Sons*, 1981.
- [7] H. Qian and S. Sapatnekar, "Hierarchical random-walk algorithms for power grid analysis," *Proc. of IEEE/ACM ASP-DAC*, 2004.
- [8] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," *IEEE/ACM ICCAD*, 2004.
- [9] Y. Coz and R. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid State Electronics*, vol. 35, no. 7, pp. 1005-1012, 1992.
- [10] X. Tan, C. Shi, D. Lungeanu, J. Lee and L. Yuan, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," *Proc. of IEEE/ACM DAC*, 1999.
- [11] S. Director and R. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circ. Theory*, vol. 16, no. 8, pp. 318-323, Aug 1969.
- [12] S. Zhao, K. Roy and C.-K. Koh, "Decoupling capacitance allocation for power supply noise suppression," *Proc. of ISPD*, 2001.
- [13] H. Zheng, L. Pileggi and B. Krauter "Electrical modeling of integrated-package power and ground distributions," *IEEE Design & Test of Computer*, vol. 20, issue 3, May-June 2003.
- [14] H. Su, S. Sapatnekar and S. Nassif, "Optimal decoupling capacitor sizing and placement for standard-cell layout designs," *IEEE Trans. on CAD*, vol. 22, no. 4, April 2003.
- [15] H. Chen and D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip designs," *IEEE/ACM DAC*, 1997.
- [16] T. Hesterberg, "Advances in importance sampling," *Ph.D. Dissertation*, Dept. of Statistics, Stanford University, 1988.
- [17] M. Iosifescu, "Finite Markov processes and their applications," *John Wiley & Sons*, 1980.