Pergamon

Solid-State Electronics Vol. 42, No. 4, pp. 581–588, 1998
© 1998 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0038-1101/98 $19.00 + 0.00

PII: S0038-1101(97)00283-9

PERFORMANCE OF RANDOM-WALK CAPACITANCE EXTRACTORS FOR IC INTERCONNECTS: A NUMERICAL STUDY

Y.L. LE COZ[1], H.J. GREUB[1] and R.B. IVERSON[2]

[1]Center for Integrated Electronics, Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.
[2]Random Logic Corporation, 2934 Beau Lane, Fairfax, VA 22031, U.S.A.

(Received 5 June 1997; accepted 22 September 1997)

Abstract—With ever-shrinking feature geometries, multilevel IC interconnects will greatly influence overall circuit behavior. In particular, efficient numerical evaluation of 3D IC-interconnect capacitance is essential to achieving targeted design goals. Previously, we have reported a new random-walk (RW) algorithm for extracting capacitance of complex multilevel IC interconnects [see, Y.L. Le Coz and R.B. Iverson, *Solid-St. Electron.* **35**, 1005 (1991)]. Here, for the first time, we present a numerical study concerning the influence of interconnect complexity on RW-extractor performance. Of primary interest, are the empirical relationships among geometric complexity, run time, and memory usage. We also include, for reference, comparisons with conventional finite-element (FE) and boundary-element (BI) capacitance extractors. Despite the general computational limitations of these conventional extractors, we have attempted to normalize numerical errors to a single common value. The problem geometry selected for our study consists of a long "bus" wire situated beneath a series of shorter cross wires. Problem complexity is controlled by increasing the bus-wire length and adding cross wires. We have found that at 1% normalized error in bus-wire self-capacitance, the RW extractor has the shortest execution time, which is uniquely *independent* of problem complexity. In addition, because the RW extractor requires *no numerical meshing*, an RW:BI:FE memory-usage ratio of 1:10^3:10^7 was observed. We conclude that the RW method may possibly excel in the high-complexity regime characteristic of multilevel IC interconnects. © 1998 Elsevier Science Ltd. All rights reserved

1. INTRODUCTION

As ICs become more complex, and as their feature sizes are scaled down, multilevel interconnects will, to a large degree, control circuit performance. Parasitic IC-interconnect capacitance will become a key factor in predicting signal delay and electrical cross talk. Unfortunately, 1D, 2D and quasi-3D analyses of parasitic capacitance often fail to accurately describe the true 3D nature of the fringing electric fields. These fringing fields are an important component of interconnect capacitance when wire widths and spacings are small.

To date, *QuickCap*Ⓣ is one of the fastest and most reliable capacitance extractors for complex, 3D IC interconnects*[1]. It employs a novel random-walk algorithm to solve the underlying 3D Laplace equation, which we first reported here several years ago[2]. This extractor handles non-Manhattan shapes, multiple dielectrics, and full-chip geometries. It has been found to be in close agreement — in both 2D and 3D and for simple and complex problems — with exact analytical solutions, with finite-difference, finite-element and boundary-integral benchmark solutions, and with experimental data for a number of test structures[2–6].

We present here a numerical study investigating the influence of 3D problem complexity on a random-walk extractor's run time and memory usage. Our primary aim is to understand the numerical behavior of random-walk (RW) methods in high-complexity environments, such as multilevel IC interconnects. All RW results have been obtained with *QuickCap*. For reference, we have included data and extrapolated results from conventional finite-element (FE) and boundary-integral (BI) extractors.

We consider this work a suggestive case study concerning the influence of complexity on RW-extractor performance. We take a strict empirical approach in our analysis. We do not formally re-derive any mathematical theory concerning the origins of numerical error, instead, we simply observe the dependencies of both run time and memory usage on problem complexity. We normalize, as best as possible, all capacitance data to a common numerical error when comparing RW, FE, and BI extractors.

*Note, heuristic extraction methods may be required for full-chip net extraction; yet, fast physically based extractors must still be used for calibration or validation, and accurate extraction of specific critical nets in a full-chip environment.

FE and BI results were obtained using the best extractors available to us*. Limitations we encountered in using both these extractors were especially apparent in the high-complexity, large-problem-domain regime. The limitations, we believe, are not inherent to the extractors themselves, but are simply a reflection of the general numerical methodology (meshing and matrix inversion) on which they are based. For this reason, and since, we repeat, this work focuses on general trends and properties of, specifically, RW extractors, we report our FE and BI extractor results as "generic".

The geometry of our study is a long wire situated at right angles and below an arbitrary number of equally spaced, shorter cross wires. A large ground plane is also included in the problem. We calculate the self-capacitance of the long wire with each solver. The number of cross wires quantifies problem complexity.

Section 2 is a brief, qualitative overview of the relatively new RW method, with additional remarks pertaining to numerical limitations of all three (RW, FE, BI) methods. In Section 3, we present the results of our numerical study. General conclusions drawn from our findings are listed in Section 4.

2. NUMERICAL METHODS FOR CAPACITANCE EXTRACTION

We now present an overview of the RW (random-walk) method for capacitance extraction. A more detailed and extensive description of this numerical method is available in Ref.[2]. We will close this section with some remarks concerning two other competitor methods, FE (finite element) and BI (boundary integral). These latter methods are well established and need no lengthy description here (see for example[7–12]). We will, however, confine our remarks to numerical issues relating to the suitability of these capacitance extraction methodologies for complex geometric structures. Note lastly, for all numerical methods discussed here, capacitance extraction is tantamount to solving a 3D Laplace equation.

The RW method directly computes the electrode capacitance matrix[2]. It is based on the finite-domain Green's function for Laplace's equation within a scalable cube domain. This Green's function establishes nested-integral representations for each element of the capacitance matrix. An element of the capacitance matrix C_{ij} associated with two electrodes i and j can be formulated in terms of an infinite sum of nested integrals. Equation (1) shows the first few terms of such an infinite sum

$$\begin{aligned} C_{ij} = \int_{\mathscr{G}_i} \mathrm{d}\xi\, \mathrm{d}\eta\, s_i(\xi, \eta) \Bigg\{ \int_{S^j_{\alpha(\xi,\eta)}} \mathrm{d}\xi'\, \mathrm{d}\eta'\, w(\xi, \eta|\xi', \eta') \\ \times \bar{G}[\alpha(\xi, \eta)|\xi', \eta'] + \int_{\bar{S}_{\alpha(\xi,\eta)}} \mathrm{d}\xi'\, \mathrm{d}\eta'\, w(\xi, \eta|\xi', \eta') \\ \times \bar{G}[\alpha(\xi, \eta)|\xi', \eta'] \int_{S^j_{\alpha(\xi',\eta')}} \mathrm{d}\xi''\, \mathrm{d}\eta'' \\ \times \bar{G}[\alpha(\xi', \eta')|\xi'', \eta''] + \ldots \Bigg\}. \end{aligned} \quad (1)$$

Integrations are to be performed either over a Gaussian surface for the ith electrode $\mathscr{G}_i$, or over parts of "maximal-cube†" surfaces S^j_α and $\tilde{S}^j_\alpha$, of size α. The dummy parameters $(\xi, \eta, \xi', \eta', \ldots)$ correspond to maximal-cube surface points, and determine the respective cube size α in each integral above. S^j_α is the part of a maximal cube coincident with electrode j, $\tilde{S}^j_\alpha$ is the part not coincident. The sampling function s_i, weight function w, and surface Green's function $\bar{G}$ follow from theory[2].

Probability rules, in the form of random walks, can be established to estimate the infinite sum above. Elements of the capacitance matrix can thus be computed by summing statistically independent numbers arising from a series of random walks between electrodes in the computational domain. A random walk is made up of a sequence of hops to the surfaces of maximal cubes. A maximal cube is the largest cube, centered about any particular point in space, bounded by the nearest electrode (interconnect wire). Each maximal cube is centered on the surface point of a previous maximal cube, which determined the previous hop in the random walk. Walks emanate from cube center points on the Gaussian surface that surrounds the electrode from which the capacitances to all other electrodes is to be found. Random walks terminate when encountering any electrode, after which an appropriate numerical weight is recorded and a new walk starts from the original Gaussian surface. Coupling capacitance to a particular electrode is calculated by summing numerical weights for walks terminating on that electrode, and dividing the sum by the total number of walks to all electrodes.

Apart from capacitance, the RW method, uniquely, generates a statistical error corresponding to one statistical standard deviation. Performing a sufficiently large number of walks minimizes this error. The RW method's accuracy is limited by the computer's execution speed and by its numerical precision. Numerical errors can also arise when nonidealities exist in generating random numbers for the walks. These errors are the stochastic counterparts of the deterministic FE and BI rounding and discretization errors. Importantly, the RW

*Specific implementations of RW, BI, and FE methods for extraction can yield, possibly, significant differences in performance. We have chosen one of the best 3rd-party BI, and FE, extractors to benchmark our study. The BI extractor is noncommercial university code implementing an iterative Krylov-subspace technique. The FE extractor is a leading commercial product for IC CAD.

†We will precisely define maximal cubes later.

method requires no numerical mesh, so that only an extremely large number of IC-interconnect wires will tax computer memory.

We now make some remarks about FE and BI methods for solving Laplace's equation and extracting 3D capacitance. The FE method is a general procedure for solving differential equations (see for example[13]). The method applies local basis-function solution expansions over volumetric elements of the computational domain. Element volumes must be kept small to minimize numerical discretization error; hence, many elements are needed to fill the computational domain. Difference equations for the expansion coefficients of the basis functions within each element can be deduced from the original differential equation. For Laplace's equation, the FE expansion-coefficient equations constitute a linear system. Because of the computational effort and computer memory needed to mesh complex volumetric geometries, the FE method may have difficulties with the IC-interconnect problem. Moreover, the FE mesh will produce a large linear system of numerical equations. With increasing geometric complexity, rapid numerical solution of these equations may not be possible; not to mention, that when solution is possible, significant rounding errors can occur.

The other numerical procedure we include as a reference in our study is the BI method (see, for example, Ref.[13], pp. 461–81). The BI method divides up the surfaces of all capacitive electrodes into small areal panels. These panels contain surface charges that generate electric fields and potentials. Over each panel, the surface-charge distribution can be expanded in 2D basis functions. Using the infinite-domain Green's function, one can then integrate over panels to find their electric-potential contribution. A set of linear equations involving all panel surface-charge expansion coefficients can thus be derived, given the physical requirement of spatially constant potential on electrode surfaces. Like the FE method, a sufficiently small panel size minimizes discretization error. The BI method, nonetheless, is expected to be significantly faster and less memory intensive than the FE method, because of reduced mesh dimensionality. Nonetheless, the BI method can also tax computer memory when the number and the length of IC wires increase*. We note, as well, that though BI equations are generally fewer in number than corresponding FE equations, the BI coefficient matrix is dense compared with the sparse-banded FE matrix. The tightly coupled BI equations can become ill conditioned and, hence, can be much more sensitive to rounding error.

3. NUMERICAL STUDY†

We present a numerical study concerning the influence of complexity on the performance of a RW-based capacitance extractor. Also included, for comparison, are data for generic FE and BI extractors. All extraction software was installed on a common Sun SPARCstation™ 10/51 computer.

Figure 1(a) shows the simplest problem geometry in the study. It consists of a 2 μm long wire, or "bus", 1 μm above the ground plane and 1 μm below a 9 μm long cross wire, or "load". Both bus and load wires have a 1×1 μm square cross section.

There are a several reasons for selecting in our study the basic geometry of Fig. 1:

(i) *Fringing 3D fields*. The bus is perpendicular to the loads. Bus-to-load capacitance will consequently have a strong 3D fringing-field component.

(ii) *Small unit cell*. The unit-cell problem of Fig. 1(a) (one load) is small enough to ensure that FE and BI extractors can conveniently produce a fine numerical mesh, allowing characterization of numerical error as a function of run time and memory usage.

(iii) *Variable load number*. Problem complexity readily translates into number of loads, by periodic extension of the unit cell of Fig. 1(b).

(iv) *IC-interconnect geometry*. The problem geometry has important features that are suggestive of actual multilevel IC interconnects — a long wire with inter-level coupling to its neighbors.

In Section 3.1, we find the dependence of numerical error in extracted bus self-capacitance as a function of run time and of memory usage for the structure of Fig. 1(a). "Bus self-capacitance‡" is the total capacitance from the bus to all loads plus the ground plane. In Section 3.2, we find the dependencies of both run time and memory usage on problem complexity. Problem complexity is increased by lengthening the bus and adding loads spaced

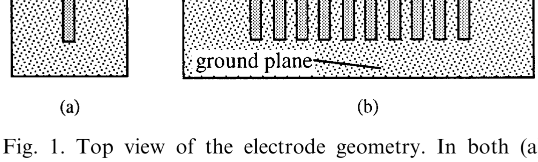

Fig. 1. Top view of the electrode geometry. In both (a) and (b), bus wires are horizontal and load wires are vertical. Problem complexity is measured by the number of loads.

*Dielectric interfaces will also consume additional run time and memory, as they require special interfacial panels.

†Portions of this section derive from *Users' Guide for QuickCap*, version 1.1, by Ralph B. Iverson and Yannick L. Le Coz, an unpublished work, ©Random Logic Corporation 1996. The separate, derivative work of this section in no way affects, limits, or infringes Random Logic Corporation's original copyright.

‡Hereafter, we abbreviate this term as "capacitance".

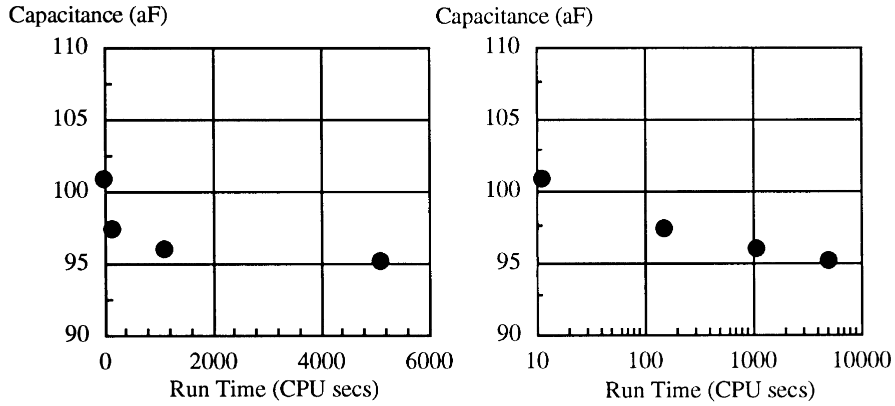

Fig. 2. Capacitance *vs* run time for the FE extractor. Problem geometry is shown in Fig. 1(a).

1 μm apart. Figure 1(b) shows an example multiple-load problem geometry. In this study, raw data yielded an RW extraction error of 0.1% and an FE extraction error of 8%, while the BI extraction error was 1%. The run times and memory usages are thus normalized (or calibrated) using the numerical-error data obtained from the analysis presented in Section 3.1. Hence, each extractor's effective numerical capacitance error can be adjusted to a specified common value.

3.1. *Error normalization*

Computational accuracy can be expressed as a numerical error, that is, the difference between the exact and calculated values of capacitance. We postulate the following empirical relations:

$$\text{Numerical error} = A \times (\text{Run time})^{-a},$$

$$\text{Numerical error} = B \times (\text{Memory usage})^{-b}. \quad (2)$$

Exponents a and b depend only on extractor methodology (RW, FE, or BI), independent of problem geometry. Constants of proportionality A and B, however, depend jointly on methodology and geometry.

For the FE and BI extractors, we estimate numerical errors by using converged capacitances for each respective extractor on the same problem. We take the converged value to be the zero-slope asymptote of capacitance vs logarithm of run time. FE and BI meshes become finer with increasing accuracy. A finer mesh implies a longer run time and requires greater memory usage. Because of the FE extractor's limitation in forming a large volumetric mesh, its converged capacitance value is estimated by extrapolation. Subtracting, for each extractor, its respective converged capacitance from its capacitance for any particular run time and memory usage defines "numerical error" in this study. That is, FE- and BI-extractor error is referred to their individual converged capacitances. This definition of error is an approximation of the "true error". True error depends on the single exact capacitance for the problem, which is not known.

The RW extractor, as previously mentioned, furnishes a statistical estimate of numerical error. We report, in this study, a numerical error corresponding to one standard deviation of summed RW data, the 1-σ error. This error corresponds to an average statistical error, and it is essentially a measure of true numerical error.

With one load, the FE extractor produced capacitance vs run-time data shown in Fig. 2. Because of the extractor's computational limitations, data were obtained only at four different volumetric mesh densities. In the left plot, which has a linear time axis, the last capacitance value appears converged. The same data in the right plot, however, which has a logarithmic time axis, shows that capacitance has not actually converged near the longest run time. Using an estimated extrapolated convergence value of 92aF allows us to plot numerical error against run time and memory usage, shown in Fig. 3.

We now turn to the BI method. The plot of Fig. 4 shows how capacitance varies with run time. As with the previous FE case, the one-load geometry of Fig. 1(a) has been used. Memory usage and run time increase together with areal mesh density. The converged capacitance value of about 143.5aF produced the error-normalization data of Fig. 5. Numerical error is almost inversely proportional to both run time and memory usage.

Converged capacitance values for FE and BI extractors do not agree (92aF vs 143.5aF, respectively). The disparity in converged capacitance is caused by differences in each extractor's boundary conditions*. We view the inability to customize boundary conditions as a general limitation of both FE and BI methods. For example, the BI method

*This disparity has been corroborated. With periodic boundary conditions, as are used by FE extractor, the RW extractor gives 92.8aF at 0.5% 1-σ error. With a finite ground plane, as is used by the BI-extractor, the RW extractor gives 143aF at 0.5% 1-σ error.

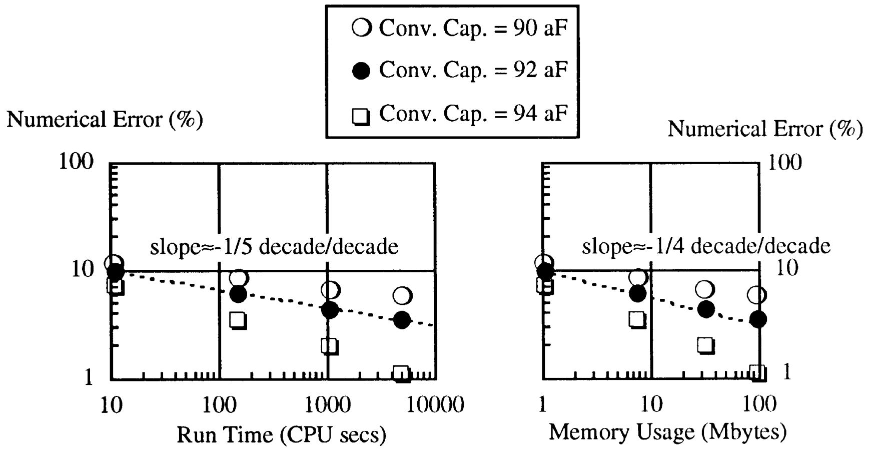

Fig. 3. Numerical error *vs* run time and *vs* memory usage for the FE extractor. Three possible converged capacitance values are plotted, 92aF was selected for error normalization.

uses infinite-domain Green's functions, and, therefore, can apply boundary conditions at an effective mathematical infinity. FE methods, in contrast, are not capable of modeling an infinitely large computational domain, because of practical constraints with meshing large volumes. Error-normalization data and our subsequent problem-complexity study is rather insensitive to any differences in boundary conditions. This, because of the relatively small number of numerical boundary equations compared with the number of numerical FE volume-element, or BI areal-panel, equations.

There is no need to generate similar error-normalization data for the RW extractor. This extractor estimates capacitance as a sum of statistically independent random variables. The sum possesses a well-known inverse-square root dependence

$$\text{Numerical error}|_{RW} \propto \frac{1}{\sqrt{\text{Run time}}}, \quad (3)$$

independent of memory usage. Our RW-extractor data supports the above error dependence.

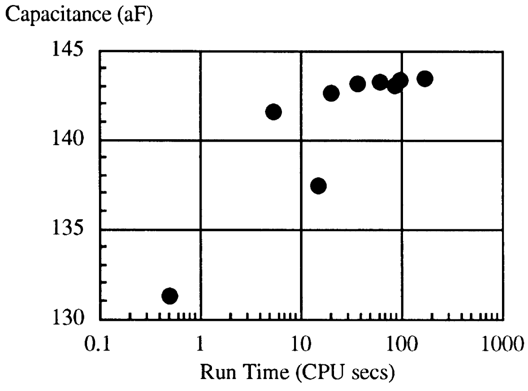

Fig. 4. Capacitance *vs* run time for the BI extractor. The problem geometry consists of one load of Fig. 1(a).

3.2. *Problem complexity*

We are ready to establish relationships among run time, memory usage, and problem complexity. Problem complexity, of course, is of utmost importance in analyzing IC interconnects. Like Equation (2), we postulate similar empirical relations:

$$\text{Run time} = C \times (\text{Problem complexity})^{-c},$$

$$\text{Memory usage} = D \times (\text{Problem complexity})^{-d}. \quad (4)$$

Exponents c and d, as before, depend only on the extractor methodology. Constants of proportionality C and D depend on both on methodology and problem geometry.

For the data that follow, effective numerical error in extracted capacitance was kept constant with increasing problem complexity. For the FE and BI extractors, mesh density was held constant; for the RW extractor, 1-σ error was held constant. Normalization of errors to a common value was achieved using Figs 3 and 5, and relation (3).

The plots of Fig. 6 are raw 1% BI-extractor data; RW and FE data have been normalized to 1%, from raw values with 0.1 and 8% numerical error, respectively. Observe, the RW-extractor run time is the shortest and is nearly independent of problem complexity. Run-time cross over between BI and RW extractors occurs at a complexity of about one load. The BI:FE run-time ratio is about 1:10^5. Memory usage follows an RW:BI:FE ratio of about 1:10^3:10^7.

Referring again to Fig. 6, the FE and BI run times generally increase with increasing problem complexity. This behavior is a consequence of having to add volumetric-mesh or areal-panel elements to accommodate a longer bus with a greater number of loads. Interestingly, RW run time is insensitive to problem complexity. This insensitivity is a direct result of the statistical averaging of random walks. As noted earlier, the RW method estimates

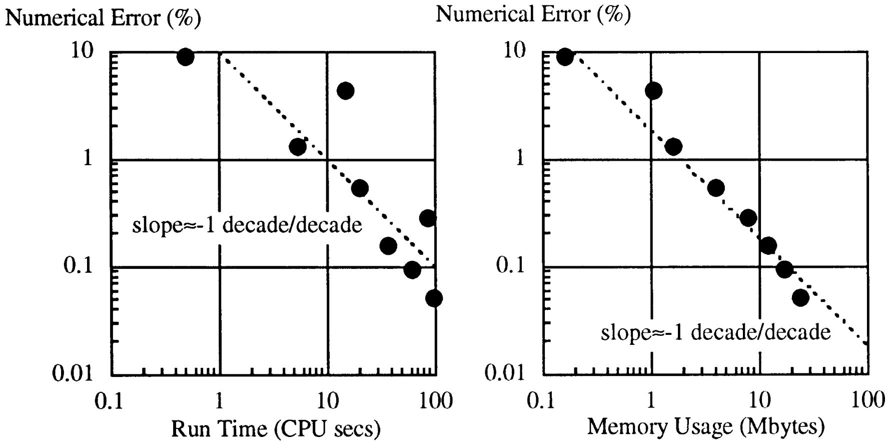

Fig. 5. Numerical error *vs* run time and memory usage for the BI extractor

capacitance by averaging a set of random numbers generated by each walk. The numbers correspond to estimates of the local bus self-capacitance per unit length near where the walks start, multiplied by the length of the bus. As long as the standard deviation of these capacitance-per-unit-length samples are independent of complexity, run time will, also, be essentially independent of complexity. The standard deviation of these capacitances is, by definition, the 1-σ error in total capacitance. We believe that insensitivity of self-capacitance to complexity is a general property of the RW method, even in actual IC interconnects where wire distribution is not spatially uniform.

Memory-usage data of Fig. 6 shows expected increases for all three methods with increasing complexity. Volumetric meshes consume more memory than equivalent areal meshes, thus the FE extractor

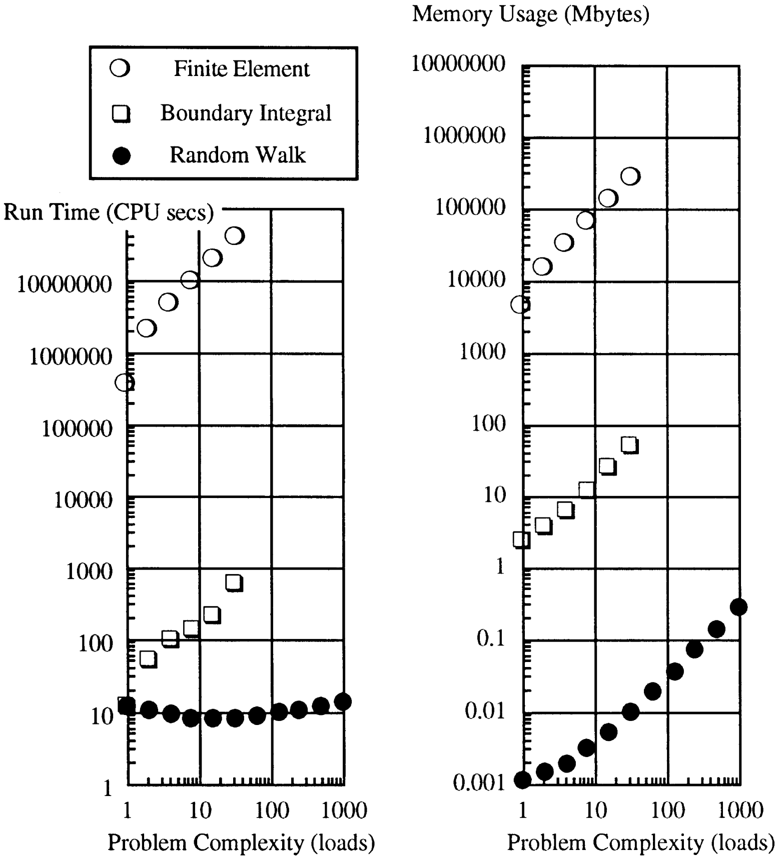

Fig. 6. Run time and memory usage *vs* problem complexity for RW, FE, and BI extractors. Numerical error is normalized to 1%.

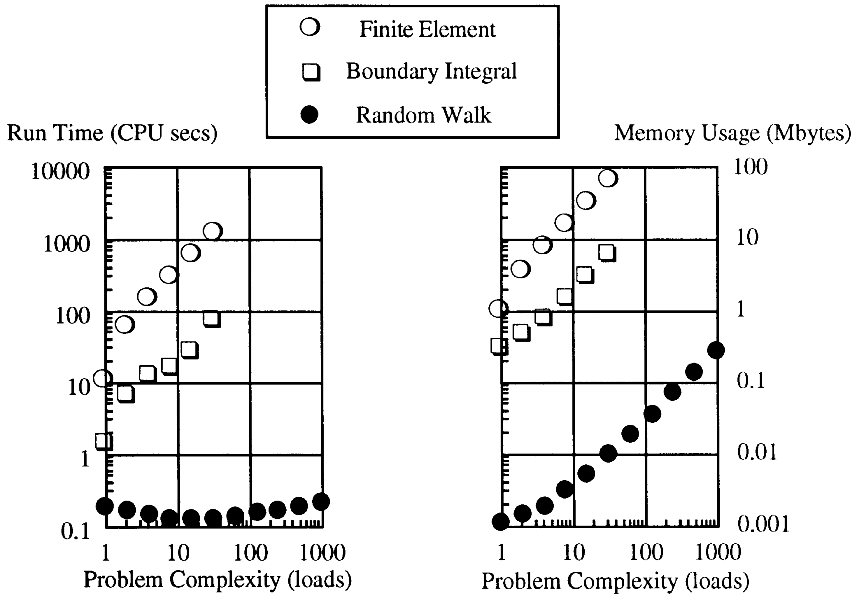

Fig. 7. Run time and memory usage *vs* problem complexity for RW, FE, and BI extractors. Numerical error is normalized to 8%.

requires more memory than the BI extractor. The RW extractor uses no mesh; instead, *entire* 3D parallelpipeds representing the bus and load wires are represented with their two body-diagonal endpoint coordinates.

Results for raw 8% FE-extractor data are plotted in Fig. 7. We also show RW and BI data normalized to 8%, from their raw values of 0.1 and 1% numerical error, respectively. At 8% normalized error, the BI:FE run-time ratio is only about 1:10. Again, the RW extractor has a nearly constant, superior run-time dependence. The RW:BI:FE memory-usage ratio is about 1:10^3:10^4. General trends and behaviors resemble the previous 1% normalized-error results. We conclude that differences in run time and memory usage among the three extractors tend to decrease somewhat with increasing numerical error, for this problem.

4. CONCLUSION

We have presented a numerical case study examining the influence of problem complexity on the performance of a RW extractor for IC interconnect capacitance. Comparisons have also been made with generic FE and BI extractors. These extractors use deterministic numerical methods to solve the underlying 3D Laplace equation. The RW extractor uses a stochastic method. We have studied the run-time and memory-usage requirements of each extractor, as influenced by problem complexity (number of interconnect wires), given a common (normalized) solution accuracy.

In summary, for our specific case study:

(i) IC-interconnect extractors must be compared over a wide range of problem complexities, not just the simple ones.

(ii) For complex problems, and for a given numerical error, the RW extractor is faster than FE and BI extractors. We consider this behavior a general property of the methods on which our extractors are based.

(iii) For a given numerical error, FE- and BI-extractor run times are order N, where N is complexity; run time for the RW extractor is order N^0 *.

(iv) For a given numerical error, memory usage for all the extractors is order N. However, the RW extractor consumes significantly less memory than the other extractors. We consider these behaviors a general property of the methods on which our extractors are based.

(v) The RW method has two unique distinguishing characteristics — run time for evaluating self-capacitance is virtually *independent* of problem complexity, and accuracy depends on run time but *not* on memory usage.

Acknowledgements—The work of YLL and RBI is supported by the SRC Center for Advanced Interconnect Science and Technology at Rensselaer, Contract 448.023.

REFERENCES

1. *QuickCap*, a trademark of Random Logic Corporation, Fairfax, VA.
2. Le Coz, Y. L. and Iverson, R. B., *Solid-St. Electron.*, 1991, **35,** 1005.

*We predict that run time for the RW extractor is actually order log(N) for significantly larger problems sizes than those presented in this study.

3. Le Coz, Y. L. and Iverson, R. B., *Proc. IEEE 8th Int. IEEE VLSI Multilevel Interconnection Conf.*, 1991, p. 365.
4. Le Coz, Y. L. and Iverson, R. B., *Proc. IEEE Multi-Chip Module Conf.*, 1992, p. 86.
5. Le Coz, Y. L., Iverson, R. B., Vu, Q. T. and Van Wijnen, P., *Proc. IEEE 10th Int. VLSI Multilevel Interconnection Conf.*, 1993, p. 526.
6. Le Coz, Y. L., Greub, H. J., Garg, A., McDonald, J. F., and Iverson, R. B., *Proc. 13th Int. IEEE VLSI Multilevel Interconnection Conf.*, 1996, p. 230.
7. Ruehli, A. E. and Brennan, P. A., *IEEE Trans. Microwave Theory Tech*, 1973, **MTT-21,** 76.
8. Cottrell, P. E. and Buturla, E. M., *IBM J. Res. Develop.*, 1985, **29,** 277.
9. Lai, F. S., *Solid-St. Electron.*, 1989, **32,** 141.
10. Zemanian, A. H., *IEEE Trans. Electron Dev.*, 1988, **35,** 985.
11. Nabors, K., Kim, S. and White, J., *IEEE Trans. Microwave Theory Tech.*, 1992, **7,** 1496.
12. Akcasu, O. E., Lu, J., Dalal, A., Mitra, S., Lev, L., Vasseghi, N., Pance, A., Hingarh, H. and Basit, H., *Proc. IEEE Int. Electron Dev. Meeting*, 1995, p. 495.
13. Lapidus, L. and Pinder, G. F., *Numerical Solution of Partial Differential Equations in Science and Engineering*. John Wiley and Sons, New York, 1982, pp. 34–107.