An Adaptive Window-Based Susceptance Extraction and its Efficient Implementation

Guoan Zhong, Cheng-Kok Koh, Venkataramanan Balakrishnan, and Kaushik Roy
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285
{zhongg,chengkok,ragu,kaushik}@ecn.purdue.edu

ABSTRACT

The determination of the set (or window) of segments that are inductively coupled to a significant degree with a given segment plays a fundamental role in window-based techniques for the extraction of the susceptance of interconnect structures. We present a measure that quantifies the degree of coupling between segments in a window, thereby paving the way for an adaptive scheme for determining the coupling window associated with each segment. This measure has the properties that: (i) it is well-correlated with the simulation error that is inherent in window-based susceptance extraction techniques, and (ii) it can be computed efficiently using incremental and computation reuse techniques.

Categories and Subject Descriptors

B7.2 [**Integrated Circuits**]: Design Aids—*Simulation, Verification*

General Terms

Algorithms

Keywords

inductance, susceptance, interconnect

1. INTRODUCTION

With the continual increase in clock frequencies and global interconnect lengths and decrease in signal transition times, the accurate modeling of inductance effects is becoming more and more important. The partial inductance matrix L obtained from the PEEC model [1] is large and dense. Direct simulation of the full L matrix is usually impractical, owing to the enormous demands it places on computation time and memory. Moreover, it is now being recognized that the *susceptance* matrix $K = L^{-1}$ is often (approximately) sparse [2, 3]; sparsification of K based on this observation can yield considerable savings in inductance (or more precisely susceptance) extraction, and simulation.

DAC 2003, June 2–6, 2003, Anaheim, California, USA.

If K is exactly sparse, each wire is coupled with only a few other wires in terms of susceptance. We will therefore refer to the set of those wires as the *coupling window* associated with the wire of interest. It has recently been shown that all the entries of the matrix K can be reconstructed just from the entries of the sub-matrices in L that correspond to the coupling windows, without having to invert the entire L matrix. To generate the non-zero entries of K, we only need to invert these sub-matrices, which are usually very small [4].

In summary, if K is exactly sparse, and if the sparsity pattern in K is known, its nonzero entries can be computed with much reduced computation as compared with inverting L [4]. However, in practice, K is never exactly sparse. Moreover, the sparsity pattern in K is not known *a priori*. Of course, deducing the sparsity pattern in K by inverting L is out of the question as this is precisely the operation that we are trying to avoid. In practice, the (approximate) sparsity pattern in K can be deduced from proximity arguments: It is realistic to assume that a wire influences and is influenced by only "nearby" wires; this argument has been used in a number of earlier approaches [5, 3, 6, 7]. A more sophisticated technique, developed in the context of capacitance extraction, adaptively determines the coupling window for each wire [8]. All these techniques can be viewed as applying sound heuristics for the coupling window determination, and hence the sparsity pattern in K. While these heuristics often work well, it would be desirable to have a systematic, quantitative, approach for determining the coupling window. Moreover, the measure used to determine the coupling window must correlate well with the error in simulation that results from the use of this coupling window.

The main contribution of this paper is a quantitative measure that enables the systematic determination of the coupling window. This measure has the properties that:

- It correlates well with the simulation error that arises from windowing.
- It can be computed very efficiently using incremental techniques.

We describe this measure and discuss its properties in §2. In §3, we show how this measure can be computed efficiently. Numerical results are presented in §4.

2. DETERMINING SPARSITY IN THE SUSCEPTANCE MATRIX

Our objective is a systematic, quantitative, technique for determining of the coupling window associated with a wire.

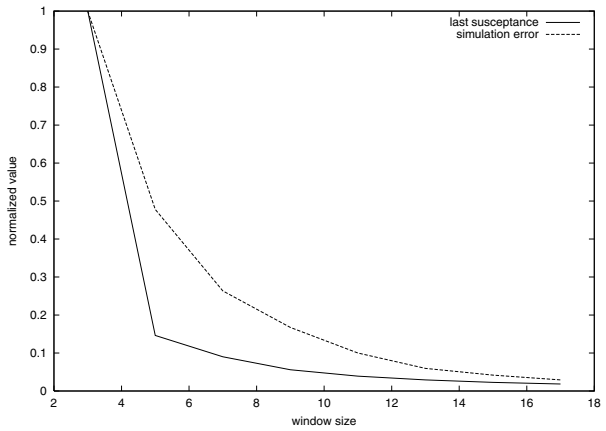

Figure 1: A comparison between the normalized simulation error and the trend predicted by the relevant entry of the windowed mutual susceptance matrix. A severe mismatch is evident.

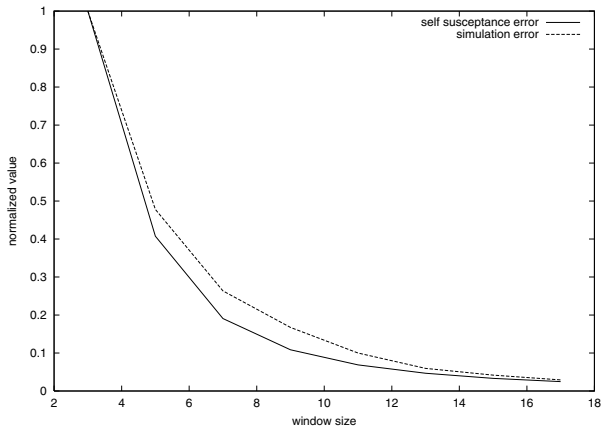

Figure 2: A comparison between the normalized simulation error and the trend predicted by the normalized difference between the exact and windowed mutual susceptance values.

Clearly, there is a trade-off between coupling window size and accuracy. A smaller coupling window offers increased computational savings in the (approximate) computation of the susceptance: By inverting a matrix composed of only those entries in L that correspond to the coupling window, we may compute approximately the corresponding entries of the K matrix. To be precise, these are the "windowed" susceptance entries; we will therefore denote any entries of the windowed susceptance matrix with a "hat", i.e., by $\hat{K}_{ij}$. Of course, the use of the windowed susceptance entries leads to errors in simulation. A reasonable technique for determining the coupling window would be to begin with a small window size, and increment it until the simulation error is acceptable. Of course, the simulation error cannot be directly used because it requires, by definition, simulation with the full L matrix. Thus, the key to a systematic determination of the coupling window size is a measure which has the following properties:

P1. It correlates well with the simulation error.

P2. It can be computed with reasonable computational effort.

The main contribution of the paper is the presentation of such a measure.

A first attempt at postulating such a measure is as follows: "Begin with a small-sized coupling window, and compute the corresponding windowed susceptance entries. Increment the coupling window size, and inspect the new mutual susceptance entry that corresponds to the interaction of the wire under consideration with the new wire that has been included. Stop when the new entry is small."

The measure underlying this approach, i.e., the size of the new entries in the windowed susceptance matrix, satisfies properties [P2]. However, it does not possess property [P1], that is it fails to be well-correlated with the simulation error. This is illustrated by an example consisting of 55 wires. Suppose that we wish to determine the coupling window of wire 28. Figure 1 shows the normalized values of the windowed mutual windowed susceptance between wire 28 and the wire that is being added, as a function of the window size. Also shown in this plot is the normalized simulation error as a function of window size. The simulation error is defined as the value, averaged over a number of switching patterns, of the quantity

$$\sum_j \frac{\int |v_j(t) - \hat{v}_j(t)|\, dt}{\int |v_j(t)|\, dt},$$

where $v_j(t)$ is the exact voltage profile at wire j, and $\hat{v}_j(t)$ is the voltage profile at wire j obtained from the simulation that uses the windowed susceptance matrix $\hat{K}$. Clearly, there is poor agreement between the trend predicted by the mutual susceptance values and the exact simulation error.

Remarkably, it turns out that the *difference* between the diagonal entries of the windowed susceptance matrix $\hat{K}$ and the exact susceptance matrix K, plotted as a function of window size, tracks very well the simulation error. For instance, consider Figure 2 which contains a plot of the normalized difference between the exact and windowed self-susceptance values of wire 28, shown as a function of the window size. It is clear that this quantity, as a function of the window size, accurately tracks the normalized simulation error, also shown in the figure. Of course, the difference between the exact and windowed self-susceptance values cannot be directly used (as it requires knowledge of the exact susceptance matrix). However, the actual values of this difference are not important; what is crucial is that they predict the simulation error trend. Thus, when the windowed self-susceptance values do not change significantly with window size, it is evident that the simulation error values will also not change significantly. With most reasonable heuristics, proximity arguments are used to decide on which wires should be included in a coupling window. With such heuristics, the *change* in the simulation error with window size decreases monotonically, i.e., a small *change* in the simulation error implies a small value for the simulation error. Thus, the *change* in the difference between the exact and windowed self-susceptance values can be used as an excellent predictor for determining an acceptable window size. *The important point is that the change in the difference between the exact and windowed self-susceptance*

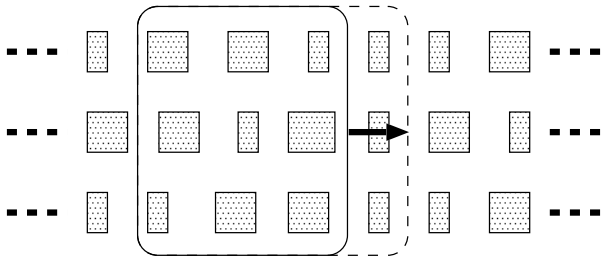

Figure 3: Growing of the coupling window using proximity arguments.

values equals the change in the windowed self-susceptance values themselves, and does not require the knowledge of the exact self-susceptance values themselves. Thus the *change* in the windowed self-susceptance values can be used as an excellent predictor of window size.

3. EFFICIENT COMPUTATION OF COUPLING WINDOW

We now briefly discuss how the procedure for determining the window size can be incorporated into a susceptance extraction procedure. In a typical extraction procedure, wires are first divided into segments. Fat wires may have to be meshed at the crosssection to capture skin effect and proximity effect. For each segment, proximity and geometrical information are used to grow the coupling window associated with the segment. Figure 3 illustrates the idea. While the growth of the coupling window is indicated in only one direction, there are typically six directions in which the window can be extended. If the change in the self-susceptance value of the segment of interest falls below a pre-determined threshold, a suitable coupling window is deemed to have been determined.

A straightforward implementation of the procedure for determining the coupling window for each segment requires a matrix inversion each time the coupling window is updated. Although the window sizes are relatively small, the computational cost of these inversions can accumulate to be formidable. This issue can be addressed using two related techniques:

1. Suppose that we have computed the windowed susceptance matrix $\hat{K}_n = L_n^{-1}$ associated with a segment for a certain coupling window size, and suppose that the coupling window grows in size by one (for simplicity). Thus we need to invert the matrix L_{n+1} to obtain the new windowed susceptance matrix $\hat{K}_{n+1}$. The matrix L_{n+1} is simply the matrix L_n padded with extra elements that correspond to the self and coupling inductances with the added segment. This observation can be used to reduce the computation of L_{n+1}^{-1} by an order of magnitude. This will be described further in §3.1.

2. Suppose that we have completed the determination of the coupling window (and the associated windowed susceptance matrix) for segment i, and we move on to an adjacent segment $i+1$ (see Figure 4). From proximity arguments, it is clear that the coupling window for segment $i+1$ will have a significant number of segments in common with that of segment i. This fact can be used to "reuse" numerical calculations. We will discuss this further in §3.2.

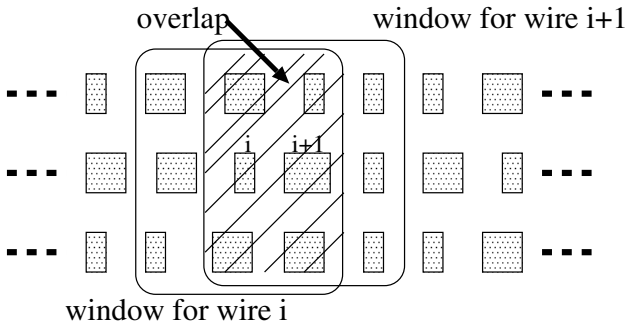

Figure 4: Computation reuse between neighboring segments.

3.1 Incremental computation of susceptance matrix

Suppose a window size of n is not acceptable, and we need to add one more segment. (While we consider the addition of one segment for simplicity, the following development can be extended to the addition of multiple segments.)

Then the inductance matrix of the new window is:

$$L_{n+1} = \begin{bmatrix} L_n & m \\ m^T & l \end{bmatrix}, \tag{1}$$

where L_n is the inductance matrix of the original window, l is the self inductance of the added segment, and m is the mutual inductance between the newly-added segment and segments in the original window. It can be shown that

$$L_{n+1}^{-1} = \begin{bmatrix} L_n^{-1} + \frac{1}{k}vv^T & -\frac{1}{k}v \\ -\frac{1}{k}v^T & \frac{1}{k} \end{bmatrix}, \tag{2}$$

where

$$v = L_n^{-1}m \text{ and } k = l - m^T L_n^{-1} m \tag{3}$$

Eqn. (2) can be used to update the windowed susceptance matrix efficiently. To get L_{n+1}^{-1} from L_n^{-1}, we need to calculate v, k,v/k, and $\frac{1}{k}vv^T$. These calculations need only $2(n^2+n)$ multiplication operations that is an order-of-magnitude smaller than the computation required for evaluating L_{n+1}^{-1} from scratch.

We observe that $L_{n+1}^{-1} > 0$, and therefore $k > 0$. Thus every diagonal entry of $L_n^{-1} + \frac{1}{k}vv^T$ is larger than the corresponding entry of L_n^{-1}. Thus the windowed self-susceptance increases monotonically with window size.

3.2 Computation reuse between neighboring segments

Suppose that we have completed the determination of the coupling window $\mathcal{C}_i$ for segment i, and we wish to determine the coupling window $\mathcal{C}_{i+1}$ for a nearby segment $i+1$. From proximity arguments, it is clear that $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ will have many segments in common. Thus, $\mathcal{C}_{i+1}$ can be obtained by:

1. First removing all segments from $\mathcal{C}_i$ that are not significantly coupled with segment $i+1$, i.e., those segments that are not needed in $\mathcal{C}_{i+1}$.

Table 1: Extraction time (in seconds).

Number of segments	40K	100K
adaptive window with direct inversions	2079	5144
adaptive windowing with incremental inversion and computation reuse	270	680

2. Next adding other segments that are significantly coupled with segment $i+1$.

This procedure is illustrated in Figure 4.

The first step of this procedure can be performed as follows. Beginning with the coupling window $\mathcal{C}_i$ for segment i, we use proximity arguments to remove segments, and each time efficiently update the windowed susceptance matrix using the techniques from §3.1. To illustrate, suppose that we wish to remove segment 1, so that we wish to compute $L_{n+1}(2:n+1, 2:n+1)^{-1}$ from L_{n+1}^{-1}, where we have used $A(i:j, m:n)$ to denote the sub-matrix at the intersection of rows i to j and columns m to n of A.

Using Eqn. (2), we have

$$L_{n+1}(2:n+1,2:n+1)^{-1} = L_{n+1}^{-1}(2:n+1,2:n+1) - L_{n+1}^{-1}(2:n+1,1)L_{n+1}^{-1}(1,2:n+1)/L_{n+1}^{-1}(1,1). \quad (4)$$

Thus, $L_{n+1}(2:n+1,2:n+1)^{-1}$ can be computed from L_{n+1}^{-1} efficiently, with only (n^2+n) multiplication operations. (The same idea can be used to remove any segment from $\mathcal{C}_i$, not just segment 1.)

The procedure of removing segments is continued until the change of the self-susceptance of segment $i+1$ is larger than a pre-determined threshold, indicating that the segment that was removed is significantly coupled to segment $i+1$.

Step 2 in the above procedure can be performed efficiently, using techniques that are essentially the same as the ones described in §3.1 for growing the coupling window.

4. NUMERICAL RESULTS

We present numerical results obtained from Matlab implementations of the adaptive susceptance extraction procedure. Two sets of results are presented. The first corresponds to an implementation where the coupling windows are determined adaptively, but with the windowed susceptance matrices computed using direct matrix inversions. The second set corresponds to the more efficient implementation based on the incremental and computation reuse techniques from §3.

The implementations were tested on two randomly generated interconnect structures consisting of 40,000 segments and 100,000 segments respectively. A relative change in self-susceptance of 0.01% was used as the threshold to determine whether a segment should be included in a coupling window. The extraction times for an Intel Pentium IV 1.4GHz system are shown in Table 1. It is clear that the techniques of §3 cut down the computation by a factor of seven, when compared with an implementation that employs direct matrix inversions.

We also investigate how the relative change threshold for the self-susceptance influences the window size and the simulation error. Figure 5 consists of two plots; the one shown in solid lines demonstrates the dependence of the simulation error on the threshold, and validates the observation made in §2 that the change in the windowed self-susceptance values serves as an excellent predictor of simulation error. The plot in dotted lines shows window size as a function of the threshold. A natural trade-off between the simulation error and the window size (and therefore extraction time) is evident. While the threshold can be any positive number, the window sizes are discrete, resulting in piecewise constant curves.

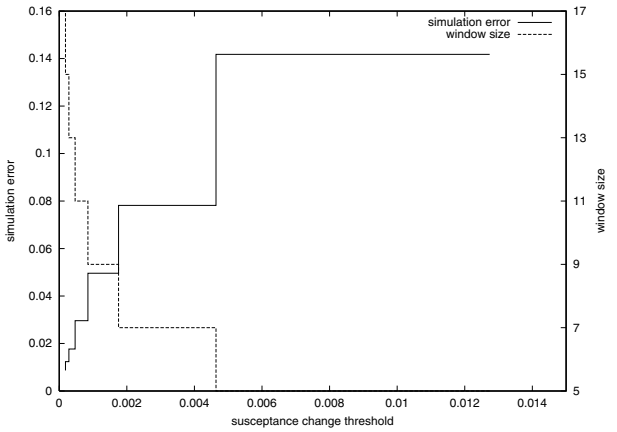

Figure 5: Simulation error and window size as a function of the relative change threshold.

5. REFERENCES

[1] A. E. Ruehli, "Inductance calculation in a complex integrated circuit environment," *IBM Journal of Research and Development*, pp. 470–481, Sept. 1972.

[2] A. Devgan, H. Ji, and W. Dai, "How to efficiently capture on-chip inductance effects: introducing a new circuit element K," in *Proc. Int. Conf. on Computer Aided Design*, 2000, pp. 150–155.

[3] M. Beattie and L. Pileggi, "Efficient inductance extraction via windowing," in *Proc. Design Automation and Test in Europe Conf.*, 2001, pp. 430–436.

[4] G. Zhong, C.-K. Koh, and K. Roy, "On-chip interconnect modeling by wire duplication," in *Proc. Int. Conf. on Computer Aided Design*, 2002, pp. 341–346.

[5] H. Ji, A. Devgan, and W. Dai, "KSim: a stable and efficient RKC simulator for capturing on-chip inductance effect," in *Proc. Asia South Pacific Design Automation Conf.*, 2001, pp. 379–384.

[6] T. H. Chen, C. Luk, H. Kim, and C. C.-P. Chen, "INDUCTWISE: Inductance-wise interconnect simulator and extractor," in *Proc. Int. Conf. on Computer Aided Design*, 2002, pp. 215–220.

[7] K. L. Shepard and Z. Tian, "Return-limited inductances:a practical approach to on-chip inductance extraction," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 425–436, Apr. 2000.

[8] M. Beattie and L. Pileggi, "Error bounds for capacitance extraction via window techniques," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 311–321, Mar. 1999.