

# Simulation Approaches for Strongly Coupled Interconnect Systems

Joel R. Phillips  
Cadence Berkeley Laboratories  
Cadence Design Systems  
San Jose, CA, 95134  
jrp@cadence.com

L. Miguel Silveira  
INESC ID/ Cadence European Laboratories  
Dept. of ECE, IST/Technical Univ. Lisbon  
Lisboa, 1000 Portugal  
lms@inesc.pt

## Abstract

*Shrinking feature sizes and increasing speeds of operation make interconnect-related effects very relevant for current circuit verification methodologies. Reliable and accurate system verification requires the full analysis of circuits together with the environment that surrounds them, including the common substrate, the packaging structures, and perhaps even board information. In this paper we discuss circuit-level simulation algorithms that enable the analysis of the impact of strongly coupled interconnect structures on nonlinear circuit operation, so as to allow reliable and accurate system verification.*

## 1 Introduction

While circuit verification for many years has been concerned with issues of noise coupling between signals and the effects of parasitic devices, such considerations were usually assumed local and handled with tools that relied on simplified formulas and approximations. Now, such simplified approaches to circuit verification are not sufficient as rising frequencies of operation increase the performance impact of parasitic elements, while shrinking voltage levels and device feature sizes mean decreased margin for error. Particularly in analog applications, insufficient attention to parasitic modeling considerations can result in design failures. Therefore, there has been considerable attention devoted to the impact of package parasitics, non-ideal power/ground networks, substrate effects, relevance of on-chip inductance and device-level parasitics.

Traditional approaches to large interconnect problems are usually directed at one of the two ends of the accuracy/efficiency spectrum: very high speed, or very high accuracy. Rule-based heuristic approaches, aimed at determining the supposedly most relevant couplings have been the technique of choice for parasitic extraction. Unfortunately even mature pattern- and formula- based capacitance extraction tools, which have been under development for the better part of two decades, commonly produce errors in outlying cases of 20%-50% (or more). Formula-based tools for inductance and substrate extraction are far less mature and in the case of packaging nearly non-existent. Even if such formulas did exist, they would be targeted at extracting only the most relevant couplings. This can be very useful to a circuit designer, since the dominant sources of coupling can be quickly identified, and then remedied. However, such information is less useful to a builder of CAD tools, since one of the main points of performing detailed, circuit-level analysis is to obtain the accurate answers needed to verify design decisions. In

the context of parasitics analysis, this means building a tool that can predict the amount of residual coupling remaining *after* the dominant sources have been removed. Such sources are, by definition, second order, and without accurately predicting their magnitude, correct operation of the design cannot be verified. Unfortunately, in many cases, as we shall show, simple models are likely to underestimate coupling effects in circuits designed to be noise immune.

A more robust approach is based on building a model that fully describes the couplings using a very accurate method such as a field solver, and then devise a better numerical algorithm to accelerate the circuit simulation. Traditionally this approach has been considered infeasible for large circuits as the extraction stage quickly becomes computationally intractable since several thousand field solutions must be performed to extract the circuit matrix and even more modern "fast solvers" [13, 9, 12] are only capable of performing rapid extraction on small sections of layout geometry. Recently, however, super-fast field solvers [11] have emerged that are capable of performing single high-accuracy extractions on large portions of layout. Techniques have also been demonstrated [10] that show how to reduce the total number of extraction steps that must be performed. Since in circuit level simulation of parasitic effects the intent is to assess the performance impact of the parasitics on operation of nonlinear circuitry, the remaining challenge is to show how to efficiently include models extracted by such means in circuit-level simulation.

Inclusion of parasitic models introduces additional variables into the circuit equations which immediately represent a constant factor increase in cost, as simulation cost is roughly proportional to the number of unknown variables. More important, however, is that parasitic elements tend to make the circuit simulation problem more *strongly coupled*, increasing the average number of connections to each circuit node, behavior which adversely impacts the performance of the sparse-matrix operations needed in most circuit simulators by increasing the number of "fill-in" elements created during matrix factorization. Mutual inductive couplings in package models, and resistive couplings in substrate analysis, generate nearly full, or what is called *dense* large matrices which will lead to unacceptable simulation requirements. The complexity of the simulation which, in the absence of strong parasitic coupling, is empirically observed to scale at the rate of  $O(n) - O(n^{1.2})$ , where  $n$  is the total number of circuit variables, can increase to between  $O(n^{1.5})$  (for simple mesh topologies) and  $O(n^3)$  (for the most complicated package and substrate models). Furthermore, the storage of a dense  $N \times N$  matrix of couplings in a circuit simulator is impractical for large  $N$ , particularly in RF simulation. As a result simulation of

large circuits can quickly become intractable.

In this paper we examine several approaches to reduce the growth of complexity encountered when simulating large dense coupling networks representing interconnect and substrate structures. As a model problem, we will consider substrate coupling interactions, though we emphasize that most of our results can be applied to the other common interconnect problems as well. We present several algorithms that enable such a simulation within an acceptable timeframe while preserving high accuracy, and compare the virtues of each. Furthermore, the methods proposed are flexible in the sense that they allow easy trade-off of accuracy versus efficiency. Therefore they can be applied in accurate circuit-level simulation at acceptable albeit large cost, or they can be relaxed and applied in fast circuit verification using simplified models and algorithms such as used presently in fast timing simulators.

In Section 2, we present background on circuit simulation and our model problem, substrate coupling analysis, and establish notation. In Section 3, we overview the most simple approach to analyzing strongly-coupled interconnect, matrix truncation, and discuss some situations in which it may fail. The heart of the paper is Section 4, where we present algorithms capable of manipulating large substrate networks with minimal loss of accuracy. We present a novel class of semi-implicit integration schemes, and discuss the relation of these schemes with iterative solution methods for implicit discretizations. Section 5 presents the results of computational experiments and conclusions are drawn in Section 6.

## 2 Technical Background

### 2.1 Electrical Circuit Simulation

Contemporary circuit simulation programs are designed to solve the differential-algebraic equations governing the time-domain evolution of circuit state quantities in response to specified stimuli. Without loss of generality, we may write these equations as

$$\frac{d}{dt}q(v) + i(v) = u(t). \quad (1)$$

This notation is motivated from nodal analysis where  $q$  represents stored charge,  $i$  device currents,  $v$  node voltages that are the state variables,  $u$  external sources and the above equation enforces current conservation (KCL). For circuits containing inductors and other elements, modified nodal analysis can be used but in essence the formulation above is still valid albeit the meaning of the various variables needs to be changed appropriately. There is a vast literature (e.g., [6]) associated with the solution of such equations, to which the reader may refer for details of their solution. For our purposes it is enough to note that circuits generate equations that are *stiff*, meaning having a wide range of time constants in the solution, and that contain algebraic constraints. Such equations are solved using implicit discretizations, the simplest of which is the backward-Euler method:

$$\frac{q(v^{n+1}) - q(v^n)}{h} + i(v^{n+1}) = u(t^{n+1}). \quad (2)$$

where  $v^n$  is the approximate solution to (1) at the  $n$ th timepoint,  $t^n$ . At each timestep, this nonlinear set of equations must be solved for the solution at the next timepoint,  $v^{n+1}$ . This is usually done using Newton's method. Newton's method requires solution of consecutive solutions of linear equations,

$$\left[ \frac{C(v^{n+1})}{h} + G(v^{n+1}) \right] x = b, \quad (3)$$

for some  $x$  and  $b$ , where  $C$  and  $G$  are the Jacobian matrices associated with  $q$  and  $i$ , respectively:

$$C(v^{n+1}) \equiv \left. \frac{\partial q}{\partial v} \right|_{v^{n+1}} \quad G(v^{n+1}) \equiv \left. \frac{\partial i}{\partial v} \right|_{v^{n+1}}. \quad (4)$$

These equations are usually solved by direct LU factorization of the Jacobian matrix,  $J = C/h + G$ , using sparse linear algebra techniques [4].

### 2.2 Modeling Substrate Coupling

The substrate problem is a potentially major problem in analog, RF, and mixed-signal designs. It is generally defined as the unintended electrical coupling through the common substrate between circuit nodes that are designed to be isolated. Mixed-signal design, where the concern is usually with noise generated by fast-switching digital nodes impacting sensitive analog nodes, has received the most attention. However undesired effects of coupling through the substrate is also known to affect purely digital circuits by introducing noise in lines and affecting the delays of certain gates. In purely analog, and particularly RF designs, coupling between different parts of an analog circuit is the issue. Substrate parasitics may reduce performance, or in severe cases create unwanted feedback paths that can lead to instabilities.

Substrate coupling and the modeling of the effects of that coupling have been the subject of much research over the past few years. The problem is made more difficult due to the fact that substrate coupling is in the most general analysis a global phenomena and its effect cannot always be accurately and reliably predicted by analyzing small subsets of the layout. Fast switching logic components inject current into the substrate causing voltage fluctuation which can affect the operation of sensitive as well as far away analog circuitry for instance through the body-effect, since the transistor threshold is a strong function of substrate bias. The main challenge is therefore to accurately and efficiently include all the mutual interactions among the substrate nodes, sources and receptors, which can easily number in the thousands, each potentially coupled to all the other (or at least to a large number of other) nodes.

When a substrate model is added to the circuit equations, additional nodes, needed to model substrate contacts, transistor bulk nodes, and so forth, are typically added to the circuit equations. In any event we may always formulate and order the equations such that we can write the new augmented state vector  $v_{aug}$  as

$$v_{aug} = \begin{bmatrix} v \\ v_{sub} \end{bmatrix} \quad (5)$$

where  $v_{sub}$  are the new variables. As substrate coupling is a linear effect, to describe the computational impact on the above equations it is sufficient to describe the impact on the Jacobian matrices,  $C$  and  $G$ . These matrices are enlarged as

$$C \rightarrow C_{aug} = \begin{bmatrix} C & C_{cs} \\ C_{sc} & C_{sub} \end{bmatrix} \quad G \rightarrow G_{aug} = \begin{bmatrix} G & G_{cs} \\ G_{sc} & G_{sub} \end{bmatrix} \quad (6)$$

where  $G$  is the original conductive circuit Jacobian,  $G_{sub}$  represents coupling among the substrate nodes,  $G_{cs}$  and  $G_{sc}$  coupling between the primary circuit and the substrate parasitics. Similar notation holds for the  $C$ -matrices. In this notation the circuit equations become

$$\frac{d}{dt} \begin{bmatrix} q(v) \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} C & C_{cs} \\ C_{sc} & C_{sub} \end{bmatrix} \begin{bmatrix} v \\ v_{sub} \end{bmatrix} + \begin{bmatrix} i(v) \end{bmatrix} + \begin{bmatrix} G & G_{cs} \\ G_{sc} & G_{sub} \end{bmatrix} \begin{bmatrix} v \\ v_{sub} \end{bmatrix} = \begin{bmatrix} u(t) \end{bmatrix}. \quad (7)$$

The six substrate-related matrices are determined by the modeling strategy used. The simplest such strategy is to generate compact analytical representations of the substrate induced parasitics by experimenting with simple test configurations consisting of a small number of contacts [8, 15, 7, 16]. Such models can give good intuition about the underlying physics of the substrate, but cannot account for complex many-body interactions such as introduced by intervening contacts, guard rings and other isolation

An alternative approach is to perform a full numerical solution of the differential equations that describe substrate transport [3, 17, 14, 5]. Many different approaches to this problem are possible, including finite-difference, finite-element, boundary-element and related methods. Often the substrate itself is modeled as purely resistive, meaning that the matrices  $C_{cs}$ ,  $C_{sc}$ ,  $C_{sub}$  will contain entries due only to junctions, wells, and other localized circuitry. They are therefore sparse, and often do not contribute to the poor computational scaling of the extraction problem to any significant degree. The matrices  $G_{sc}$ ,  $G_{cs}$  are also often sparse, representing the relatively few connections the substrate nodes have with the external circuitry. In the most general case, however,  $G_{sub}$  is dense.

Both extraction and later circuit simulation is difficult if  $G_{sub}$  is dense. Extraction is difficult because, for each of the  $N$  columns of  $G_{sub}$ , a numerical field solution must be performed. Various algorithms are known for accelerating the step of extracting the coupling information pertaining to a single contact. However even the best approaches still require  $O(N)$  time, resulting in a total complexity of  $O(N^2)$ , with quite large constant factors. If a dense  $G_{sub}$  is used in a standard circuit simulation, the scaling of the computation time for simulation is as  $O(N^3)$ . Recently an approach [10] was published that claims to reduce the overall extraction time to  $O(N)$ , but no guidance was given as to how to approach the ensuing circuit simulation problem. We address that issue in the next section.

### 3 Truncated Substrate Models

The most widely used approach to making the dense interconnect simulation problem tractable is to simply truncate the model by dropping interactions that are thought to be small. If enough terms can be dropped out of the matrix, it will become sparse. Particularly in the case of inductive coupling, ad-hoc matrix approximations can lead to non-physical, unstable interconnect models, but we will assume for now that the truncation can be safely accomplished, and concentrate on the implications for simulation accuracy.

Truncation is used at two stages of the substrate analysis process. First, arguments about what substrate coupling conductances are likely to be small are often used to simplify the extraction process [16, 17]. For example, to obtain the mutual conductances associated with a given contact, only a section of layout surrounding the contact to a user-specified distance may be extracted. Or, perhaps only conductances to the bulk and to near-neighbors may be retained. These assumptions simplify the extraction process because they drastically reduce the complexity of the problem that must be analyzed for each contact extraction step. They also make the circuit matrix sparse. Secondly, once extraction has proceeded, in order to further reduce the complexity of circuit model that must be analyzed, conductances that are relatively small may be dropped from the matrix entirely.

Often, simple truncation procedures can be useful in analyzing the lowest-order effects of substrate parasitics. However, the performance of some analog circuits can be quite sensitive to parasitic effects, noise, and unwanted feedback paths. For example, an RF

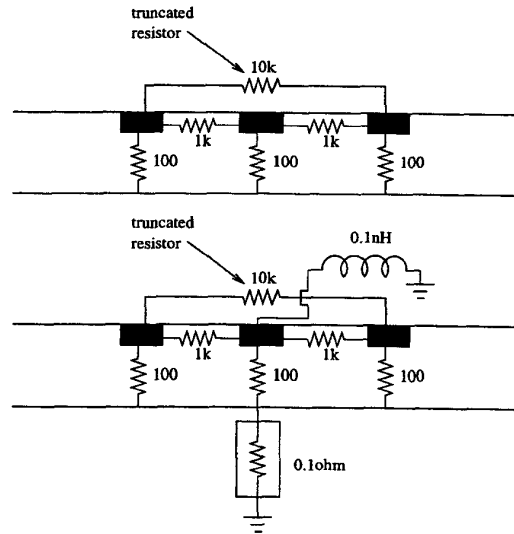


Figure 1: (a) simple substrate contact network. (b) with low-impedance backplane contact and grounded center tap.

front-end may be sensitive to microvolt noise levels and have 80-100dB of gain through the signal path. In such a high-gain system 100-120dB of isolation may be required to avoid forming possibly unstable feedback paths. Verifying correct operation of such sensitive circuits requires highly accurate estimates of how much isolation is actually available.

The problem with truncation-based strategies, and other simplified models, is that it is difficult to predict, a-priori, when they are likely to be successful and when they will fail. Partially this is because, in realistic layouts, with irregular geometries and shapes of widely varying sizes and aspect ratios, determining the critical couplings is not straightforward. But it is also true that which couplings turn out to be "important" depends on circuitry that is outside the purview of the extraction tool.

A simple example is shown in the top of Figure 1. If the nodal conductance matrix for this circuit,  $G_{sub}$ , is inspected, the entry corresponding to the 10kΩ coupling resistor will be observed to be very small compared to the other matrix entries. It is a common practice to truncate such resistors. For this simple circuit, virtually no change (less than 1%) will be observed in the amount of coupling between the first and the third node. Therefore, it might seem reasonable to discard this resistor during or after the extraction process. However, consider what occurs if some noise-prevention strategies are applied to this network. Suppose that first a low-impedance path to ground is provided for the substrate bulk or backplane, as shown in the lower portion of the figure. A substantial reduction in the amount of coupling between the first and third nodes will be observed. If that does not provide sufficient isolation, the second node might also be connected to a reference (the inductance shown above models the effect of the power/ground and/or package network). The second node might in fact represent a ground ring inserted specifically for this purpose. In this situation, truncating the 10k-Ω resistor can result in over-estimating the amount of isolation (under-estimating the coupling effects) by an order of magnitude. The reason is simple: in the absence of any other circuitry, the dominant one-three coupling path is from node one through a 100Ω

resistor to the backplane, then back through another 100Ω resistor to node three. The second most important path, with ten times the impedance, is through 1kΩ resistor to node two, then similarly to node three. But once these paths are removed from consideration by providing low-impedance paths to stable references, then only the formerly-small direct coupling 10kΩ resistor is left as the dominant path.

It should be clear then that the problem with a-priori estimation of the importance of parasitic elements is that the dominant noise coupling path potentially depends on all the connecting circuitry, in particular power/ground and package networks, behavior and connectivity of intervening structures such as guard rings, and loading on other substrate nodes (such as capacitive decoupling from transistor bulk nodes). The circuit context is also important. Small couplings to sensitive areas are more important to model accurately than large couplings in relatively noise-immune circuitry. In a large circuit, with each node having hundreds or thousands of possible couplings, there is the additional complication that many small noise contributors can sum to a non-negligible value. The net result is that truncation can have a poor tradeoff between speed and accuracy, when accuracy is properly measured.

On the other hand, if circuit context is available at the time of extraction, it is possible to exploit the information to reduce the complexity of extracted networks. For example, if information about the package and power/ground networks is available before extracting the substrate network, it is possible to estimate the strength of the indirect paths (such as between two nodes via the packplane or a package pin) in relation to the direct substrate conductance contribution. Direct substrate paths that are very weak in comparison (i.e., have conductance small compared to the divider circuit formed by the indirect parallel paths) can then be safely ignored.

## 4 Full Substrate Models

### 4.1 Multi-Level Representation

By this point it should be clear that any strategy that relies on manipulations of the full  $G_{sub}$  matrix is not likely to be efficient. Recently [10], it was proposed that representing the substrate conductance matrix in a multi-level, wavelet-like, basis can be more efficient. Adopting the notation of [10], the substrate conductance matrix is represented as

$$G_{sub} = Q \hat{G}_{sub} Q^T \quad (8)$$

where  $\hat{G}_{sub}$  is the conductance matrix in the new basis, and  $Q$  is the orthonormal change-of-basis matrix. It is argued that, for a given level of accuracy,  $\hat{G}_{sub}$  is much sparser than  $G_{sub}$ . By construction,  $Q$  turns out to be sparse (it can be shown to have  $O(n \log n)$  entries).

In our experience not all of the heuristics for “sparsifying”, or truncating,  $\hat{G}_{sub}$  proposed in [10] are robust, (in particular, well-separated wavelet basis functions do not always have vanishing interaction, particularly on problems with multiple geometric scales present, such as the example discussed in Section 5) but the multi-level analysis offers greater flexibility than the traditional basis in constructing approximate representations. This increases the likelihood that a matrix representation can be found that is simultaneously sufficiently sparse and sufficiently accurate. In principle the multi-level basis can compute “all to all” interactions in nearly optimal complexity by attacking the problem hierarchically. Interactions between separated regions are lumped into a few vectors of the basis  $Q$  that cover large areas, whereas in traditional truncation

approaches such couplings would be neglected altogether. Further, because the multi-level basis naturally separates substrate interactions by spatial scales, it may also provide a systematic way to provide coarse resolution of the substrate in some areas, without neglecting any important noise generators, while simultaneously providing fine resolution in sensitive areas.

No strategy is given in [10] for circuit simulation, since while  $\hat{G}_{sub}$  is sparse in the wavelet basis, it is dense in the basis of the original circuit equations.

### 4.2 Implicit Integration with Iterative Linear System Solution

The obvious way to include the multi-level substrate representation in circuit simulation is to use an iterative linear solver, such as GMRES, in the inner loop of Newton’s method. The algorithm would only require matrix-vector products with  $Q \hat{G}_{sub} Q^T$ , which can be accomplished on  $O(N \log N)$  time. If the GMRES method converges rapidly, this method is then potentially of near-optimal,  $O(N)$ , complexity. In practice, there are two difficulties. First, an adequate preconditioner must be devised. A preconditioner is a matrix  $P$  constructed such that  $P$  is an approximation to  $J^{-1}$ , the Jacobian inverse, but a matrix-vector product with  $P$  can be computed quickly, ideally in  $O(N)$  time. Instead of solving  $Jx = b$ , GMRES is applied to the equivalent system  $PJx = Pb$ . The hope is that  $P$  is a sufficiently good approximation that GMRES can be guaranteed to converge rapidly. It turns out to be worthwhile to defer discussing preconditioner construction until after the next section.

The second problem is that it may be necessary to converge GMRES to very tight tolerances, to avoid losing the stability of the integration scheme or corrupting other properties of the circuit equations (such as anomalies caused by loss of strict current conservation). Therefore, while this strategy can be attractive in terms of its asymptotic complexity, in practice it may have large constant factor costs that may make it slow. Therefore we seek other ways to leverage efficient substrate representations.

### 4.3 Semi-Implicit Integration

An alternative to accelerating the computations associated with the implementation of an implicit integration scheme is to change the integration scheme itself. Consider for the moment the problem of solving a linear, time-invariant system of ordinary differential equations,

$$\frac{dx}{dt} = Ax + u(t). \quad (9)$$

The backward-Euler discretization is

$$\frac{x_{n+1} - x_n}{h} = Ax_{n+1} + u(t_{n+1}). \quad (10)$$

In the backward-Euler (and other Gear discretizations), the key step is solution of a linear system with coefficient matrix  $I/h - A$ . If  $A$  is dense, this is difficult. The idea of a semi-implicit matrix is to devise a method that is not fully implicit, but implicit enough to retain sufficient stability [2, 18]. If we split the matrix  $A$  into two parts,  $A_i$  (for implicit) and  $A_e$  (for explicit),  $A = A_i + A_e$ , a general Euler-based semi-implicit scheme is

$$\frac{x_{n+1} - x_n}{h} = A_i x_{n+1} + A_e x_n + u(t_{n+1}). \quad (11)$$

One can think of this as a combination of forward- and backward-Euler, from which it should be obvious that this scheme is still first

order accurate. In the most common semi-implicit schemes, the Jacobi-Euler scheme [2, 18],  $A_i$  is chosen to be the diagonal of  $A$ , thus making the Jacobian,  $1/h + A_i$ , diagonal and trivial to invert. If  $A$  is connectedly diagonally dominant (this occurs in important cases, such as RC networks and finite difference discretization of diffusion equations), this scheme retains the A-stability (for a stable  $A$ , it is stable for any choice of timestep) of the original backward-Euler method. Such schemes are sometimes used for the solution of certain partial differential equations.

The simplified linear algebra does not come without a price, however. Although stability properties of the methods are often good, stability may hold for a weaker class of systems that for the fully implicit methods. Semi-implicit methods also suffer from a solution artifact that can restrict their usefulness. The popular semi-implicit methods (i.e., the diagonally implicit ones) are known to have a smaller *domain of dependence* than fully implicit methods. In the partial differential equation context, this means that information can propagate (for example, from the boundaries) across the solution domain only at a finite speed that is related to the size of the timestep. If the timestep is sufficiently large that this speed is far from the speed at which information propagates in the physical system, an error is introduced. In analog circuit simulation, the practical impact of this problem is that semi-implicit methods can exhibit a non-physical, numerically introduced, excess delay in the computed waveforms. For a given timestep, semi-implicit methods are generally less accurate than their parent implicit method.

The key to effectively applying semi-implicit methods to the substrate equations, Eq. 7, is to recognize that a more effective matrix partitioning can resolve some of the accuracy and domain of dependence problems of the semi-implicit methods. In this paper, we will consider splitting only the dense portion of the substrate matrix  $G_{sub}$ . This leads to the general Euler-based semi-implicit integration scheme

$$\begin{aligned} \frac{1}{h} \left[ q(v^{n+1}) - q(v^n) \right] + \frac{1}{h} \begin{bmatrix} C & C_{cs} \\ C_{cs} & C_{sub} \end{bmatrix} \begin{bmatrix} v^{n+1} - v^n \\ v_{sub}^{n+1} - v_{sub}^n \end{bmatrix} \\ + \begin{bmatrix} i(v^{n+1}) \\ 0 \end{bmatrix} + \begin{bmatrix} G & G_{cs} \\ G_{cs} & G_{sub,i} \end{bmatrix} \begin{bmatrix} v^{n+1} \\ v_{sub}^{n+1} \end{bmatrix} + \\ + \begin{bmatrix} 0 & 0 \\ 0 & G_{sub,e} \end{bmatrix} \begin{bmatrix} v^n \\ v_{sub}^n \end{bmatrix} = \begin{bmatrix} u(t^{n+1}) \end{bmatrix}. \quad (12) \end{aligned}$$

Different specific schemes can be obtained by varying the partitioning of the matrix  $G_{sub}$ , the general goal being to make the implicitly treated piece  $G_{sub,i}$  sparse enough so that the overall circuit Jacobian can be factored efficiently, but also to retain enough global information to resolve the domain-of-dependence problems of the semi-implicit discretization. The reader will note that the implementation of this technique has strong similarities to the implementation of preconditioned GMRES. It should be clear that any given matrix partitioning can be used both as a preconditioner to GMRES and also defines a semi-implicit integration scheme. Not surprisingly, we find that a partitioning that makes for an effective semi-implicit scheme also produces an excellent preconditioner for GMRES. We shall see, however, that the two methods can have different numerical properties and error/effort tradeoffs.

We have experimented with several partitioning strategies. Three are presented in this paper. All three keep the diagonal portion of  $G_{sub}$  in the implicit part of the matrix  $G_{sub,i}$ , in order to preserve the desirable stability properties of the diagonally-implicit methods, which are known to be sufficiently stable. Because we will retain part of the off-diagonal matrix as well, if anything our methods should be more stable.

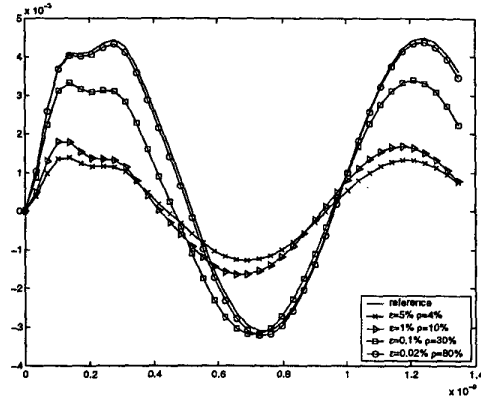


Figure 2: Truncated model with 1Ω backplane impedance to ground.

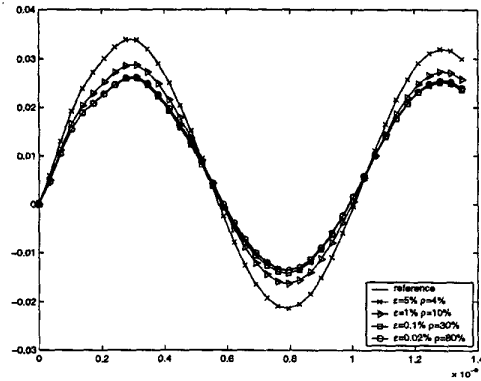


Figure 3: Truncated model with 1kΩ backplane impedance to ground.

The first method simply chooses a threshold, similar to naive truncation, and retains the largest entries of  $G_{sub}$  in the implicit part of the matrix  $G_{sub,i}$ . The difference from naive truncation, of course, is that the remaining matrix entries pass to the explicit part  $G_{sub,e}$  of the integration scheme, instead of being discarded entirely. The matrix  $G_{sub,e}$  is represented explicitly.

The second method uses the wavelet-sparsified matrix  $\hat{G}_{sub}$ , and is based on taking the  $q \times q$  principle submatrix of  $\hat{G}_{sub}$  that has largest norm, for some small  $q$ , to construct  $G_{sub,i}$ . Call this  $q \times q$  submatrix  $\hat{G}_q$ . Then we have

$$G_{sub,i} = Q_1 \hat{G}_q Q_1^T, \quad G_{sub,e} = Q [\hat{G}_{sub} - \hat{G}_q] Q^T. \quad (13)$$

Both  $G_{sub,i}$  and  $G_{sub,e}$  will be dense. In the case of  $G_{sub,e}$ , this presents no difficulty, as only a matrix-vector product with  $G_{sub,e}$  is required to compute the right-hand-side needed to implement the semi-implicit scheme, and all the constituent pieces ( $Q, \hat{G}_{sub}$ ) are sparse. However, we must re-formulate the semi-implicit scheme slightly to exploit to avoid manipulating a dense  $G_{sub,i}$ .

Suppose we must solve a linear system,

$$\begin{bmatrix} A & B_1 \\ B_2 & Q_1 \hat{G}_q Q_1^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \end{bmatrix} \quad (14)$$

with  $Q_1$  and  $\hat{G}^q$  low-rank. This may be reformulated as the equivalent, larger system by introducing the auxiliary variables  $x_3$  and solving

$$\begin{bmatrix} A & B_1 & 0 \\ B_2 & 0 & Q_1 \\ 0 & Q_1^T & -\hat{G}_q^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix}. \quad (15)$$

It should be recognized that the first system is the Schur complement of the second with respect to  $\hat{G}_q^{-1}$ . The semi-implicit discretization may be re-formulated in a similar way to preserve sparsity.

The third uses the finest-level basis functions to construct a preconditioner. This gives a much larger  $\hat{G}_q$  and  $Q_1$ , but it turns out that because the fine-level basis functions have only short range interactions, the implicit part of this partitioning is sparse in the original basis.

It is also possible to include the entire wavelet basis directly in the implicit part of the equations, without manipulation of  $\hat{G}_{sub}^{-1}$ . This will lead to fully implicit methods, not semi-implicit ones, but we discuss it in this section because of the similarity of implementation to some of the semi-implicit methods. Proceeding by analogy as in the above example, if the matrices  $Q^T B_2$  and  $B_1 Q$  are sparse, then we may solve

$$\begin{bmatrix} A & B_1 Q \\ Q^T B_2 & \hat{G}_q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \end{bmatrix} \quad (16)$$

directly. We may also form the (even larger) equivalent system

$$\begin{bmatrix} A & B_1 & 0 & 0 \\ B_2 & 0 & Q & 0 \\ 0 & Q^T & 0 & I \\ 0 & 0 & I & \hat{G}_q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (17)$$

which is guaranteed to be sparse, as long as  $A, B_1, B_2$ , and  $\hat{G}_q$  are.  $Q$  will be sparse by construction. A similar approach was used in [1].

## 5 Computational Results

For the computational studies in this paper, we have used a scalable mock-up of a situation where it is desired to assess the noise coupling between two well-separated objects consuming relatively large areas (these might be, for example, two spiral inductors in an RF design) in a sea of other, relatively unrelated, objects (e.g., transistors). Each of the intervening objects is capacitively coupled to a power/ground network. The power/ground network is modeled as a mesh of resistors, with the resistor values computed based on length of the mesh edge using parameters typical of upper layers in an aluminum interconnect process. The power grid is tied to ground through a lossy inductor intended to model package impedance. The number of objects (thus nodes in the substrate model) can be scaled by adjusting the size/density of the contacts in the intervening area. All substrate and other parasitic models were pre-computed and in this section when we quote computational costs, they pertain purely to the circuit simulation stage. For time-domain simulation, the “aggressor” node is driven with a unit-valued 1GHz sinusoid using a 100Ω impedance driver. Implementation was done using MATLAB’s sparse linear algebra, so simulation times should be considered in a relative sense only.

First we demonstrate the behavior of truncation-like algorithms. For purposes of illustration, we have followed the simplest strategy, discarding coupling conductances smaller than a threshold. We

have chosen several values for the threshold  $\epsilon$ , ranging from a relative 5% of the largest entry in  $G_{sub}$  (that is, we discard any resistor more than twenty times the smallest substrate resistor) down to relative  $\epsilon = 2e - 4$ . In Figures 2 and 3 we show the results. We have also shown  $\rho$ , defined to be the percentage of entities in  $G_{sub}$  that are non-zero.  $\rho = 100\%$  corresponds to a completely full matrix.

Figure 2 was performed with a low impedance from backplane to ground. The behavior is qualitatively similar to the small example discussed in Section 3, but on a much larger, more complex scale. The first point is that, on this couplings path, aggressive prunings are not accurate. Second, the waveforms approach the correct solution only slowly as the density of the matrix increases. This indicates a poor tradeoff of accuracy for computational cost. The final curve is close to the exact answer, but by that point, the matrix is nearly full.

Figure 3 shows the opposite case, high impedance from backplane to ground. As we might expect from the discussion in 3, the noise levels in this configuration are considerably higher. We also observe that truncation performs relatively well. This example illustrates, though, the difficulty of making generalizations about the effect on circuit behavior of ad-hoc matrix approximations. In contrast to the last example, where truncation underestimated the noise (which seems expected since fewer couplings paths are present), on this example, truncation causes the noise to be over-estimated.

Next we show the time-domain behavior of the semi-implicit methods. Figure 4 shows the solution obtained from the diagonally-implicit method. A phase lag error of about 20° is clearly noticeable. The first wavelet-implicit method discussed above, using the largest principle submatrix, is also shown. The solution is very accurate relative to the baseline backward-Euler method. It appears that well-chosen semi-implicit methods may have promise. The relevant question is, however, how do they compare to the other alternative, iterative solves with GMRES? That is, all other things being equal, using the same matrix partitioning, is it better to use GMRES iterative solves on a fully implicit scheme, or use semi-implicit methods? Figure 5 attempts to answer this question.

For this model, we solved the equations to high accuracy using the full substrate coupling matrix. In Figure 5, we show cost-error figures (the change in cost is to changing the number of timesteps; in addition to discretization error this affects how well the semi-implicit partitionings approximate the long-range couplings) for the various schemes. We have not shown direct LU decomposition or partitionings based on direct truncation – though on the small example that was used to generate these curves, the performance is competitive, and semi-implicit partitionings based on direct truncation can be competitive for fairly large examples – because our intent is to compare the properties of methods that are scalable to large circuits.

First, we note that the diagonally-implicit algorithms are not competitive. This is primarily due to the phase error effects discussed earlier. The wavelet-based semi-implicit schemes both have good cost/performance tradeoffs. GMRES based schemes generally are shifted right along the cost axis. In one case, some anomalies in the GMRES curves are evident. In other examples, we have observed that sometimes GMRES can be competitive with, or superior to, the semi-implicit schemes at high accuracies. They are rarely faster, however. Conversely one might say of the semi-implicit schemes that they are fast, but lack precision. The fully-wavelet-implicit scheme is also shown; for this example, it appears to be intermediate in cost/accuracy between GMRES and the semi-implicit methods.

With the preconditioners used in this paper, GMRES converges

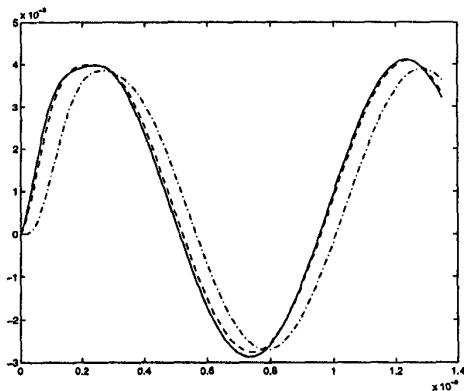


Figure 4: Backward Euler (solid line), diagonally-implicit Euler (dashed-dot line) and wavelet-implicit Euler (dash line).

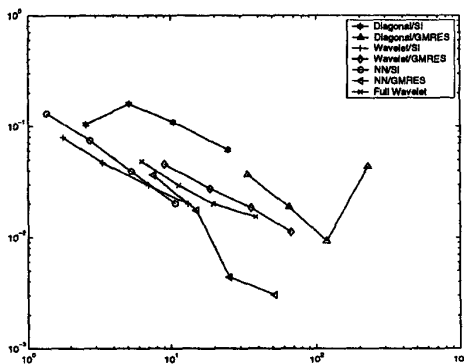


Figure 5: Error versus computation time for various scalable integration methods. Wavelet : principle-submatrix wavelet partitioning. NN: near-interaction wavelet partitioning.

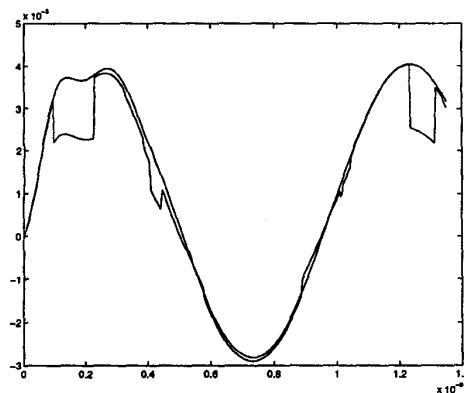


Figure 6: Waveform computation showing anomalies due to incomplete GMRES converge. Relative convergence tolerance for GMRES was  $10^{-4}$ .

within ten or twenty iterations. This indicates a preconditioner that is extraordinarily effective. However, even with this very rapid convergence, two circumstances conspire to raise the constant factors of the overall implicit-integration-with-iterative-solves method. First, the error properties of the method do not degrade gracefully as the GMRES convergence tolerance is relaxed. This can be seen in Figure 6, where the time-domain solution from the last point of the curve for cost/error (Figure 5) of the preconditioned GMRES iteration using the substrate matrix diagonals. The reason for the non-monotonic error curve is now clear : for this small timestep, the Jacobian matrix was sufficiently ill-conditioned that converging GMRES to a relative tolerance of  $10^{-4}$  was insufficient to produce a smooth curve. GMRES can be converged to tighter tolerances; in this case the cost curves move even further along the cost axis. Generally, the smaller the timestep, the tighter the GMRES tolerance must be. Practically speaking, this means that error control when using GMRES is often done by being very conservative. In contrast, the semi-implicit methods appear to have some ability to control error by timestep adjustment, and thus offer a smoother cost/performance tradeoff.

## 6 Conclusions

In this paper we presented some techniques for simulating large strongly-coupled interconnect problems. Outright matrix truncations in the original basis are not effective in reducing circuit simulation complexity, while preserving accuracy, if the substrate network undergoes non-trivial interactions with external circuitry. We presented a family of semi-implicit integration schemes, and contrasted their behavior with related iterative solution techniques. At very high accuracies, implicit integration schemes with iterative solves seem to be preferred. It appears that semi-implicit integration methods can be fast at moderate accuracy levels, if an effective partitioning is chosen. We believe these schemes are worthy of further investigation.

## 7 Acknowledgements

The authors would like to thank Joe Kanapka for useful discussions.

## References

- [1] M. Beattie, S. Gupta, and L. Pileggi. Hierarchical interconnect circuit models. In *Proceedings of the Int. Conf. on Computer-Aided Design*, pages 215–221, November 2000.
- [2] B. R. Chawla, H. K. Gummel, and P. Kozah. MOTIS - An MOS timing simulator. *IEEE Transactions on Circuits and Systems*, CAS-22:901–909, December 1975.
- [3] F. J. R. Clement, E. Z. M. Kayal, and M. Declercq. Layin: Toward a global solution for parasitic coupling modeling and visualization. In *Proc. IEEE Custom Integrated Circuit Conference*, pages 537–540, May 1994.
- [4] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Clarendon Press, Oxford, 1986.
- [5] R. Gharpurey and R. G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. *IEEE Journal of Solid-State Circuits*, 31(3):344–353, March 1996.
- [6] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic problems*. Springer-Verlag, 1991.

- [7] K. Joardar. A simple approach to modeling cross-talk in integrated circuits. *IEEE Journal of Solid-State Circuits*, 29(10):1212–1219, October 1994.
- [8] T. A. Johnson, R. Knepper, V. Marcellu, and W. Wang. Chip substrate resistance modeling technique for integrated circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, CAD-3(2):126–134, 1984.
- [9] M. Kamon, M. J. Tsuk, and J. White. Fasthenry: A multipole-accelerated 3-D inductance extraction program. *IEEE Transactions on Microwave Theory and Techniques*, 42(9):1750–1758, September 1994.
- [10] J. Kanapka, J. Phillips, and J. White. Fast methods for extraction and sparsification of substrate coupling. In *Proc. 37th Design Automation Conference*, June 2000.
- [11] S. Kapur and D. Long. Large scale capacitance extraction. In *Proceedings 37th Design Automation Conference*, pages 744–749, Los Angeles, California, June 2000.
- [12] S. Kapur and D. E. Long.  $IES^3$ : A fast integral equation solver for efficient 3-dimensional extraction. In *Proceedings of the Int. Conf. on Computer-Aided Design*, pages 448–455, November 1997.
- [13] K. Nabors and J. White. Fast capacitance extraction of general three-dimensional structures. *IEEE Transactions on Microwave Theory and Techniques*, June 1992.
- [14] T. Smedes, N. P. van der Meijs, and A. J. van Genderen. Extraction of circuit models for substrate cross-talk. In *International Conference on Computer Aided-Design*, San Jose, CA, November 1995.
- [15] D. K. Su, M. J. Loinaz, S. Masui, and B. A. Wooley. Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits. *IEEE Journal of Solid-State Circuits*, 28(4):420–430, April 1993.
- [16] A. J. van Genderen, N. P. van der Meijs, and T. Smedes. Fast computation of substrate resistances in large circuit. In *European Design and Test Conference*, Paris, February 1996.
- [17] N. K. Verghese, T. J. Schmerbeck, and D. J. Allstot. *Simulation Techniques and Solutions for Mixed-Signal Coupling in Integrated Circuits*. Kluwer Academic Publishers, 1995.
- [18] J. K. White and A. Sangiovanni-Vincentelli. *Relaxation Techniques for the Simulation of VLSI Circuits*. Engineering and Computer Science Series. Kluwer Academic Publishers, Norwell, Massachusetts, 1986.