

# Fast Methods for Extraction and Sparsification of Substrate Coupling

Joe Kanapka  
Massachusetts Institute  
of Technology  
Cambridge, MA 02139  
kanapka@rle-vlsi.mit.edu

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

Jacob White  
Massachusetts Institute  
of Technology  
Cambridge, MA 02139  
white@rle-vlsi.mit.edu

## ABSTRACT

The sudden increase in systems-on-a-chip designs has renewed interest in techniques for analyzing and eliminating substrate coupling problems. Previous work on the substrate coupling analysis has focused primarily on faster techniques for extracting coupling resistances, but has offered little help for reducing the resulting network whose number of resistors grows quadratically with the number of contacts. In this paper we show that an approach inspired by wavelets can be used in two ways. First, the wavelet method can be used to accurately sparsify the dense contact conductance matrix. In addition, we show that the method can be used to compute the sparse representation directly. Computational results are presented that show that for a problems with a few thousand contacts, the method can be almost ten times faster at constructing the matrix.

## 1. INTRODUCTION

As designers of mixed signal integrated circuits become more aggressive in their use of the technology, the conservative design practices that allowed them to ignore substrate coupling problems are being abandoned. Not only is this making substrate coupling problems more commonplace, but the ones that do occur are harder to analyze. For this reason, there is renewed interest in analyzing substrate coupling[1, 2, 3, 4, 5]. Because of the complexity of the substrate interactions, it is important to find robust and effective numerical techniques for substrate parasitic extraction and consequent simulation of substrate effects on circuit performance.

Compared to other parasitic extraction and analysis problems, such as capacitance extraction, the difficulty with extracting and simulating substrate coupling models is that the coupling is potentially *dense*. Evaluating the impact of parasitic capacitances is relatively easy in comparison, since due to strong shielding effects conductors are usually capacitively coupled only to a few other nearby bodies, and the extraction problem can be localized by analyzing only a section of the 3D geometry. Later simulation is efficient because the capacitive parasitics generate only a few more terms in the sparse circuit matrix. If the substrate is uniformly conduc-

tive, substrate coupling is more localized, so that the conductance matrix is already numerically sparse. In other cases, such as in substrates where there is a thin top layer of relatively low conductivity and lower layers of higher conductivity, a situation desirable for latchup suppression, the coupling tends to be global. In this case, every substrate contact may have important couplings to every other contact, implying that large sections of the substrate may need to be analyzed at once.

Recent work on numerical modeling of substrate coupling effects has focused on obtaining an  $n$  by  $n$  impedance or admittance coupling matrix, where  $n$  is the number of contacts. Any of the algorithms developed for rapid analysis of interconnect parasitics[6, 7, 8, 9] may be adapted to solve the problem of extracting the substrate coupling information associated with a single contact(e.g., [10, 11, 12]).

The difficulty with these approaches is that, in the substrate coupling context, knowing how to do each of these single-contact solves quickly is insufficient. First, the density of the extracted coupling matrix makes later circuit simulation prohibitively costly, because the now-dense circuit matrix must be factored hundreds or thousands of times in each simulation. Second, most methods of obtaining the  $n$  columns of the coupling matrix require  $n$  matrix solutions, which is computationally quite costly, making it impractical to solve problems with  $n$  larger than a few hundred. To address these two problems we are motivated by work on multi-scale, wavelet-like bases[13, 11, 9] for fast integral equation solutions.

By changing to a wavelet basis, which is constructed to efficiently represent coarse-grain information associated with the specific IC geometry under study, we hope that the resulting coupling matrix will become numerically sparse. That is, the use of the wavelet basis will allow us to “sparsify” the matrix, as many entries will become small in the new basis, and can simply be dropped with only small loss of accuracy. This will have advantages for later circuit simulation. It is important to note that because of the multi-resolution property of the wavelet basis, the matrix can be made sparse *while still preserving the global circuit couplings*. We show for an example that we can reduce the number of nonzeros by 90% while still achieving 1% accuracy.

In addition, because the construction of the wavelet basis gives us a good idea of the final sparsity structure of the matrix, we can exploit this structure to reduce the total number of solves needed to extract the full coupling information. To see this, consider that the naive approach to computing the conductance matrix is to per-

Permission to make digital/hardcopy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
DAC 2000, Los Angeles, California  
(c) 2000 ACM 1-58113-188-7/00/0006..\$5.00

form  $n$  solves with a unit vector or a basis function as a right-hand side. Then each solve produces one column of an  $n \times n$  conductance matrix. The number of solves can be reduced by constructing right-hand sides which are sums of basis functions whose associated current responses do not overlap. Thus we extract several columns of the conductance matrix while only doing one solve, in this way reducing the number of solves needed to obtain the complete (sparsified) coupling matrix. We present this acceleration of the extraction process as a black-box algorithm that can be used regardless of what underlying technology (finite-difference, finite-element, boundary-element) implements the detailed extraction algorithm. Our preliminary results show that number of solves can be potentially reduced by an order of magnitude for medium sized analog or RF circuits. It should be noted that on larger problems with sparser conductance matrices, many more basis functions will generate current responses that do not overlap. Therefore, each solve will yield more matrix columns, suggesting that the asymptotic complexity is closer to  $O(n)$  than  $O(n^2)$ .

The coupling matrix sparsification technique is developed in Section 3 and can be performed whether or not the technique of Section 4 for reducing the number of solves is used. In Section 2, we specify our formulation and briefly describe the simple Laplace solver we developed to do the solves. It is based on a standard finite-difference volume discretization, using a preconditioned iterative method for the solve. Numerical results are presented in Section 5.

## 2. FINITE-DIFFERENCE ANALYSIS

### 2.1 Formulation

To obtain a concrete solver technology in which to test the matrix sparsification ideas, we have implemented a simple finite difference solver. We were motivated in our choice by the fact that modern process technologies produce substrates with complex conductivity profiles as a result of features such as wells, diffusion gradients, and buried layers, so that in the future we can easily use the solver for these more complicated cases. However, we emphasize that the choice of finite-difference solver is independent of our algorithms for sparsifying the conductance matrix and reducing the number of solves, and other approaches such as boundary elements could be used.

The finite-difference formulation we use for our numerical experiments is standard and has been described elsewhere, so we discuss it only briefly here. We consider a substrate of rectangular cross section with  $L$  layers, ordered from top to bottom of thicknesses  $d_1 \dots d_L$  and conductivities  $\sigma_1 \dots \sigma_L$ . No current can enter or escape the substrate except through the contacts, which are on the top surface. This situation is described by Poisson's equation

$$-\nabla \cdot (\sigma_i \nabla \phi(\mathbf{r})) = \rho(\mathbf{r})$$

where  $\phi(\mathbf{r})$  is the potential (voltage) at  $\mathbf{r}$  and  $\rho(\mathbf{r})$  is the current flux density at  $\mathbf{r}$ . Note that this is zero inside the substrate. The boundary conditions are Dirichlet on the contacts and Neumann everywhere else:

$$\begin{aligned} \frac{\partial \phi}{\partial n} &= 0 \text{ on non-contact substrate surfaces} \\ \phi &= \phi(\mathbf{r}) \text{ on contacts} \end{aligned}$$

For simplicity of implementation, we work with piecewise-constant conductivities, so at the interfaces of regions of differing conductivity, current continuity must be enforced, leading to the interface

condition between layers  $i$  and  $i+1$ ,

$$\sigma_i \frac{\partial \phi^i}{\partial n}(\mathbf{r}) = \sigma_{i+1} \frac{\partial \phi^{i+1}}{\partial n}(\mathbf{r})$$

where  $\partial \phi^i / \partial n(\mathbf{r})$  denotes the normal derivative of  $\phi$  at  $\mathbf{r}$  on the interface, approached from layer  $i$ . The current flowing from the  $i$ th contact can be calculated as

$$\int_{\text{contact}} \sigma_1 \frac{\partial \phi}{\partial n}.$$

For computer solution of this mixed Dirichlet-Neumann problem, discretization is required. We use a uniform grid with a standard 7-point stencil. It is easiest to understand as a large 3-D grid of resistors. The only issue is what to do at the interfaces. If an interface is a fraction  $p$  between two grid layers (with conductivities  $\sigma_i$  and  $\sigma_{i+1}$ ) the conductance of the resistor crossing the layer is determined by using the formulas for resistors in series and resistance from resistivity and length:  $r = \frac{1}{\frac{p}{\sigma_i} + \frac{1-p}{\sigma_{i+1}}}$ .

### 2.2 Finite-difference solver

The finite-difference solver uses the preconditioned conjugate gradient method. The method is effective for our problem because the operator matrix  $A$  is quite sparse (at most 7 nonzeros per row) and thus cheap to apply, and because we were able to find a good preconditioner. Preconditioning is particularly important for large Laplace/Poisson type problems because the condition number grows quadratically with the discretization fineness.

Fast Poisson solvers are the method of choice for solving problems with uniform boundary conditions in each dimension. Unfortunately they cannot be applied directly to problems, such as ours, with irregular mixed boundary conditions. However, for a preconditioner to be effective it only needs to be an *approximate* inverse, suggesting the use of a fast Poisson solver *for the pure-Dirichlet problem* as a preconditioner [14]. For our experiments this usually resulted in solves to a  $10^{-5}$  relative residual tolerance in about 20 iterations.

Each iterative solve of the finite-difference equations, followed by a current calculation, can be considered as the (implicit) multiplication of the contact-contact substrate conductance coupling matrix  $G$  with a potential vector  $v$  that corresponds to voltages on the contacts. To extract the  $n \times n$  conductance matrix  $G$ , the standard approach is to set the voltages on each contact in turn to one volt and then obtain a column of  $G$  using the currents computed the finite-difference solver. That is, for column  $i$  of the conductance matrix, the voltage is set to unity on contact  $i$  and zero on all contacts  $j \neq i$ , and the vector of currents obtained from the iterative solve forms column  $i$ .

## 3. SPARSIFYING THE CONDUCTANCE MATRIX

Suppose for the moment that we have been able to extract the conductance matrix,  $G$ , that relates the voltages  $v$  on the contacts to the current flowing into the contacts, as  $Gv = i$ . In this section we explain how to sparsify the conductance matrix  $G$ —that is, find a change of basis which is inexpensive to apply and results in a matrix which is *numerically* sparse. Numerically sparse is a weaker condition than sparse in that we don't require that most entries are zero, only that most entries are smaller than a suitably chosen threshold. Then we can zero the entries below the threshold to obtain a truly

sparse approximation to the numerically sparse matrix. Of course, the threshold must not be chosen too large or the approximation will be inaccurate.

The algorithm is based on the wavelet methods developed in [9]. Here we adapt these methods to the quite different setting of substrate coupling. The methods of [9] sparsify the  $1/r$  potential-from-charge operator in three dimensions. (In making precise comparisons with the notation of [9], it is important to keep in mind that the analogous quantity to charge in [9] is *potential* for us, and the analogous quantity to potential in [9] is *current* for us, since we calculate the contact currents from the contact potentials using  $G$ ).

### 3.1 The algorithm

Assume for simplicity that the top surface of the substrate is a square with side length 1. This square can be subdivided into four squares of half the sidelength. In general, doing  $l$  levels of subdivision leads to a partition of the surface into  $2^{2l}$  squares of sidelength  $2^{-l}$ . The union of the  $2^{2l}$  squares on level  $l$  is denoted by  $S_l$ . For simplicity we assume that the level of refinement  $L$  is chosen so that contacts do not cross square boundaries. The union of the contacts in square  $s$  at level  $l$  is denoted by  $C_s$ . The standard basis consists of the characteristic functions of the  $n_s$  contacts in square  $s$ , denoted by  $\chi_{s,1} \cdots \chi_{s,n_s}$ .

Before going through the multilevel algorithm formally, we try to develop some intuition for what is going on. On the lowest level, the idea is that the new basis vector voltages will be linear combinations of the standard basis vector voltages in a given square  $s$  on level  $L$  which have *vanishing moments*, resulting in very local current response—that is, far away from  $s$  the current response will be very small. This leads to a numerically sparse matrix. Consider the zeroth-order moment

$$\mu_0(\sigma, s) = \int_s \sigma(x, y) dx dy,$$

where  $\sigma(\mathbf{r})$  is a linear combination of  $\chi_{s,1} \cdots \chi_{s,n_s}$ . On two equal-area contacts, if one voltage is set to 1 and the other to  $-1$ , we might expect that at distant points (relative to the distance between the two contacts), the current response from one contact would cancel most of the current response from the other. If we can set higher-order moments of our new basis functions to 0 as well, even faster decay of current response might be expected. In [9], there is a theorem to this effect for the  $1/r$  kernel.

Of course, some basis functions on the lowest level will *not* have vanishing moments. For example if we choose the  $(1, -1)$  voltage vector in the previous paragraph, the  $(1, 1)$  which is orthogonal to it has a nonzero zeroth-order moment. However, if we take linear combinations of basis functions in the four squares on level  $L - 1$  whose parent is the parent of  $s$ , it is possible to get moments which vanish on that higher level. This process is continued up through the levels.

We now describe the algorithm more formally. First, the moments  $\mu_{\alpha,\beta,s}$  of a function  $\sigma(x)$  are defined for  $s \in S_l$  by

$$\mu_{\alpha,\beta,s}(\sigma) = \int_s x'^\alpha y'^\beta \sigma(x, y) dx dy,$$

where  $(x', y') = (x, y) - \text{centroid}(s)$ . We want basis functions on the lowest level whose moments vanish up to order  $p$  (order  $(\mu_{\alpha,\beta,s}) := \alpha + \beta$ ). There are  $(p + 1)(p + 2)/2$  moments of order  $\leq p$ .

### 3.2 Lowest level

On level  $L$  in square  $s$ , we find the  $v_s$  basis functions  $\psi_{s,i}$  whose moments vanish and the  $w_s$  basis function  $\phi_{s,i}$  which are orthogonal to the  $\psi$  functions (and whose moments therefore do not vanish), for a total of  $n_s := v_s + w_s$  basis functions in square  $s$  on level  $L$ , by forming the  $(p + 1)(p + 2)/2 \times n_s$  matrix of moments

$$M_{(\alpha,\beta),j} = \mu_{\alpha,\beta,s}(\chi_{s,j}).$$

From this we will find the change-of-basis matrix  $Q_s = q_{i,j}$  such that

$$\begin{aligned} \phi_{s,i} &= \sum_j q_{j,i} \chi_{s,j}, & i = 1 \dots w_s \\ \psi_{s,i} &= \sum_j q_{j+w_s,i} \chi_{s,j}, & i = 1 \dots v_s \end{aligned}$$

by taking the singular value decomposition

$$M_s = [U_s] \left[ \begin{array}{c|c} S_{s,r} & 0 \end{array} \right] \begin{bmatrix} \Phi^T \\ \Psi^T \end{bmatrix}$$

which we can write in the more standard form by defining

$$S_s = \left[ \begin{array}{c|c} S_{s,r} & 0 \end{array} \right] \quad Q_s = \left[ \begin{array}{cc} \Phi & \Psi \end{array} \right]$$

so we have

$$M_s = U_s S_s Q_s^T$$

where  $U_s$  and  $Q_s$  both are matrices with orthonormal columns. The matrix  $M$  is a matrix that maps a vector representing a function  $f$  expressed as a sum of the characteristic functions  $\chi$  into moments of  $f$ . If  $Mf = 0$ , then  $f$  has vanishing moments. If  $f$  has vanishing moments, it must lie in the nullspace of  $M$ . We conclude that because by construction the rightmost  $v_s$  columns of the singular value matrix  $S_s$  are 0, then the  $v_s$  columns of the submatrix  $\Psi$  are the columns of the matrix  $Q$  corresponding to the basis functions  $\psi$  with vanishing moments, as only vectors in the space spanned by the columns of  $\Psi$  can have non-zero inner product with  $\Psi$ . Similarly the  $w_s$  columns of the submatrix  $\Phi$  give the basis functions with non-vanishing moments; these basis functions will be “pushed up” to the next level. Note that the number  $w_s$  of basis functions with nonvanishing moments is limited to  $(p + 1)(p + 2)/2$ , because the dimension of the nullspace of  $M_s$  is at least  $n_s - (p + 1)(p + 2)/2$ . We can re-organize the decomposition as  $M_s Q_s = U_s S_s$ , such that the left-hand side gives the moments of the new basis functions. This vector of moments  $M_s Q_s = U_s S_s$  will be used to calculate higher-level moments.

### 3.3 Higher levels

Call the union of lowest level basis functions over all the lowest-level squares  $\phi^{(L)}, \psi^{(L)}$ , where the  $\phi^{(L)}$  have nonvanishing moments and the  $\psi^{(L)}$  have vanishing moments. Now we describe inductively the construction of  $\phi^{(l)}$  and  $\psi^{(l)}$ , the basis functions on level  $l$  for  $l < L$  with nonvanishing and vanishing moments respectively. Each level- $l$  basis function will be nonzero in only one level- $l$  square, and the moments are computed with respect to the center of that square, just as on the lowest level. Assume  $\phi^{(k)}$  and  $\psi^{(k)}$  have been constructed for all  $k > l$ . Then the basis functions  $\phi$  and  $\psi$  on level  $l$  are combinations of the  $\phi$  functions (nonvanishing moments) of the four child cubes on level  $l + 1$ :

$$\begin{aligned}\phi_{s,i} &= \sum_{j,\alpha} q_{(\alpha,j),i} \phi_{\alpha,j}, \quad i = 1 \dots w_s \\ \psi_{s,i} &= \sum_{j,\alpha} q_{(\alpha,j),i+w_s} \phi_{\alpha,j}, \quad i = 1 \dots v_s\end{aligned}$$

Again  $Q_v$  is found by a singular value decomposition. It would be inefficient to calculate the moments directly on the larger levels since the squares get larger and larger. Fortunately this is not necessary, because one can use the already-calculated moments of the children. To do this the center with respect to which the moments are calculated must be shifted. A  $6 \times 6$  matrix results, whose entries can easily be calculated by expanding out  $(x-x_0)^\alpha(y-y_0)^\beta$  for  $\alpha + \beta \leq p$ .

The process is continued all the way to the top level. It is cheap in computational cost, as the analysis in [9] shows. In this way a change-of-basis matrix  $Q$  is calculated which is sparse, and in the new basis the coupling matrix  $G$  becomes  $Q^T G Q$ .

#### 4. REDUCING THE NUMBER OF SOLVES

In the preceding section, we assumed that we were given a known conductance matrix,  $G$ . In the situation of [9], the  $1/r$  kernel is known explicitly: given charges, potentials are calculated. In our situation, given potentials at the contacts, we calculate the currents using the finite-difference solution procedure. The  $G$  matrix is known only implicitly. Instead of a simple  $1/r$  calculation (or even a relatively simple panel integration) to calculate the potential at panel  $y$  due to a charge at panel  $x$ , for us, calculating the current on contact  $y$  due to a voltage at contact  $x$  will, as detailed in Section 2, involve a computationally expensive solve. To obtain an explicit form for the entire conductance matrix,  $n$  solves are required if the computation is performed in the usual manner.

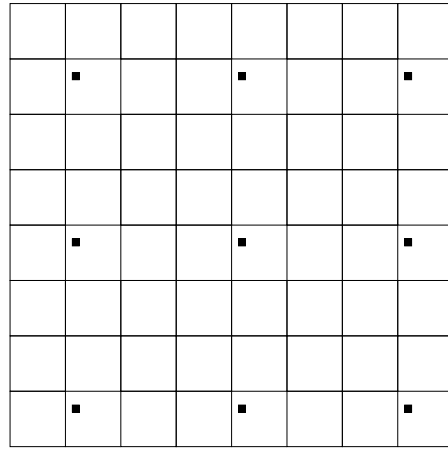
In order to reduce this cost, basis vectors can be added and the current response of the sum found in one solve. If we know where the current response will be significantly different from zero, and these areas of large entries do not overlap for the basis vectors which were added together, then we can extract the current responses for each basis vector in the sum from the solution vector. To see how this might work, consider a sparse matrix :

$$A = \begin{bmatrix} a & & & \\ c & d & & \\ & e & f & \\ & & g & h \end{bmatrix}.$$

In general, to extract all the entries of  $A$  by performing only matrix-vector product operations (which, recall, in the substrate context, are actually themselves iterative solves), we would need four products (i.e. solves), since  $A$  has four columns, one product with each of the identity vectors. But if we know the sparsity structure of  $A$ , we only need two vectors,

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Note that  $v_1$  is the sum of the first and third unit vectors, and  $v_2$  the sum of the second and fourth. Since the columns extracted by entries in the first and second rows of  $v_1$  and  $v_2$ , respectively, cannot overlap with columns extracted by placing ones in the third



Schematic representation of basis vector constituents of rhs vector for solve-reduction technique (each basis vector represented by a black square). Note that neighbor squares of distinct basis vector squares do not overlap.

and fourth rows the non-zero entries in  $A$  can be extracted just by examining the vectors  $Av_1$  and  $Av_2$ .

In order to create the right-hand side vectors, we need to make some assumption on where the large entries are. The assumption we use, of rapidly decaying response to basis functions with vanishing moments, is made by analogy with properties which are probably true in the case of the multipole algorithm for the  $1/r$  kernel. Intuitively, it is plausible that other physically based integral operators, and their inverses, share similar properties. To understand our assumptions, consider two basis vectors,  $\phi_1$  in square  $s$  on level  $l$  and  $\phi_2$  in square  $t$  on level  $m \leq l$ . (By construction, every basis vector in the new basis has support in only one square at some level  $l$  and its moments vanish at that level, except for some at level 0 whose moments don't vanish.) If  $l = m$ , we assume that  $\phi_1$  and  $\phi_2$  have a large interaction *only* if  $s$  and  $t$  are the same or neighbors. If  $m < l$ , we denote by  $p$  the parent square on level  $m$  of  $s$ , and assume that  $\phi_1$  and  $\phi_2$  have a large interaction *only* if  $p$  and  $t$  are the same or neighbors.

In order to provide the needed separation, we only combine basis vectors which are on the same level and which are at least 3 squares apart. See Figure 4. Then we have right hand side vectors for each level  $l$ ,  $i = 0 \dots 2$ ,  $j = 0 \dots 2$ , given by

$$\theta_{l,(i,j),m} = \sum_{\substack{1 \leq k,n \leq 2^l \\ (k,n) = (i \bmod 3, j \bmod 3)}} \Psi_{(k,n),m} = \Psi_{(k,n),m}$$

where  $\Psi_{(k,n),m}$  is the  $m^{\text{th}}$  vanishing-moment basis vector in the square in row  $k$  and column  $n$  on level  $l$ . Under our assumptions, for each  $\psi$  vector from the  $\theta$  solution, we can then extract the current response component in the direction of each transformed basis vector on levels  $\leq l$ . The rest of the conductance matrix (current response components on levels  $> l$  to basis functions on level  $l$ ) is obtained by symmetry.

By choosing the highest-level squares sufficiently small so that

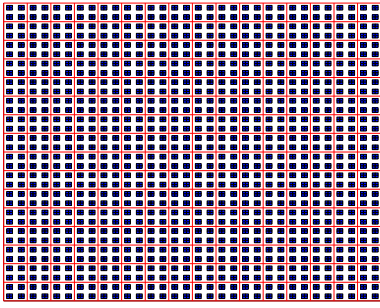


Figure 1: Regular contact layout.

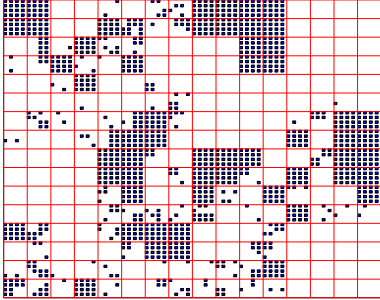


Figure 2: Irregular contact layout.

each square has at most a constant number of contacts in it, we can assure that each square at every level will have at most a constant  $c$  vanishing-moments basis functions (the constant  $c$  depends on the order of moments used). Notice that there are at most  $9c$  vectors  $\theta_{i,(i,j),m}$  in a given level, so the number of solves needed is  $9c \cdot$  number of levels, which for large  $N$  will be much smaller than  $N$ . (The number of levels depends on the problem size but for practical purposes is not large, usually of order ten).

The next section of computational results shows the accuracy of our method for some examples.

## 5. COMPUTATIONAL RESULTS

For all the examples we use two layers with the bottom-layer conductivity 100 times the top-layer conductivity. The grid used was 8 points in the  $z$ -direction by  $128 \times 128$  in the  $xy$  plane, with an interface just below the top surface ( $z = 0.5$ ). Contacts of size  $2 \times 2$  grid points were placed in a square pattern with a spacing of 4 grid points (between corresponding points on adjacent contacts), for a total of 1024 contacts. To show that our method applies to irregular layouts as well, we present a second example with 1199 contacts chosen using a randomized multilevel method designed to create clustering of contacts. Figures 1 and 2 show the contact layouts for the two examples. Even for these small examples, doing all the solves required naively (one per contact) requires around 4 hours on a SUN Ultrasparc.

First we examine the matrix  $Q^T G Q$  obtained by using the change-of-basis matrix derived in Section 3, using order 2 moment matching. We measure its numerical sparsity by choosing a “drop tolerance”  $t$  and counting the number of entries in  $Q^T G Q$  that are greater than  $t$ . We calculate a “sparsification ratio”, the factor by which we can reduce the number of entries in the matrix, by dividing the total number of entries ( $n^2$ ) in the original matrix by the number of

Example	Contacts	sparsification ratio	$L_2$ error
Regular	1024	1.001	$1 \times 10^{-3}$
Irregular	1199	1.004	$2 \times 10^{-3}$

Table 1: Sparsification in the standard basis.

Example	Contacts	sparsification ratio	$L_2$ error
Regular	1024	6	$1 \times 10^{-3}$
	1024	24	$1 \times 10^{-2}$
Irregular	1199	9	$2 \times 10^{-3}$
	1199	29	$3 \times 10^{-2}$
Regular	4096	13	$1 \times 10^{-4}$
	4096	18	$2 \times 10^{-3}$
	4096	34	$2 \times 10^{-2}$

Table 2: Sparsification using the wavelet basis.

entries greater than  $t$ .

Clearly we can achieve any desired sparsity by setting the threshold  $t$  appropriately. The real test is getting good sparsity with small error. There are many ways to measure error; we choose the  $L_2$  norm error, which is computationally easy to estimate and has a simple intuitive interpretation. The  $L_2$  norm error measures the maximum possible ratio of the length of the error vector (that is, the difference between the computed currents in the sparsified and unsparsified representations) to the input (voltage) vector length. To obtain a relative measure of error, we scale the  $L_2$  error by the  $L_2$  norm of the original (unsparsified) matrix. It is important to note that iterative methods are available for  $L_2$  norm estimation, and we can apply the unsparsified  $G$ , without actually having  $G$ , by using the solver. This is important since computational savings in the extraction depends on *not* having to extract the full  $G$ .

A common approach to reducing the density of coupling in the substrate conductance matrix is simply to drop entries that, in the normal basis, are small. Table 1 shows a sparsity ratio and error obtained by thresholding *without* a change of basis, demonstrating that this more obvious approach can be quite ineffective. However, when the multiscale basis is employed, much better results can be obtained. Table 2 shows the sparsity ratio obtained for three examples in the wavelet basis. On the larger example, over an order of magnitude compression can be achieved with very small error. Moreover, note that the compression ratio seems to increase with problem size. Figures 3 and 4 show the numerical sparsity pattern for the 1024 contact and 1199 contact coupling matrices respectively, in the wavelet basis. The multilevel structure is clearly visible.

Now we look at the performance of the technique for reducing the number of solves. We actually simulated the technique, in order to save time in the exploration of the sparsification tolerance space, by using the already-calculated standard basis coupling matrix  $G$  and taking linear combinations of its columns to get the solutions for the  $\theta$  vectors in the solve-reduction technique. Since aside from a modest setup cost, the expense of the actual algorithm is precisely proportional to the number of solves, the results will be the same doing the actual solves up to iterative method tolerance. See Table 3. The “speedup ratio” column gives the ratio of the number of solves required naively ( $n$ ) to the number of solves needed with the multiscale basis and basis vector combining, i.e., the factor by which the extraction can be accelerated. The  $L_2$  error column gives

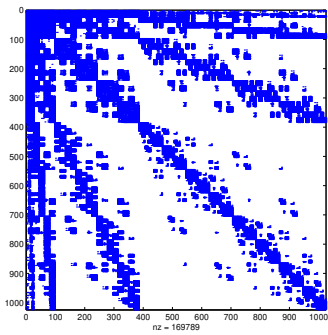


Figure 3: Numerical sparsity pattern for size 1024 regular contact coupling matrix

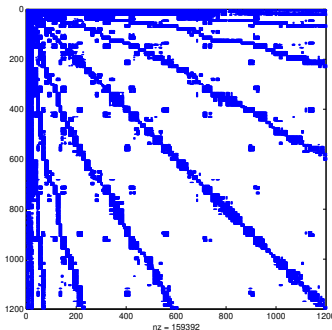


Figure 4: Numerical sparsity pattern for size 1199 irregular contact coupling matrix

the approximate norm of the difference between the dense matrix obtained directly and the sparse matrix obtained through solve reduction. This gives the maximum current error length as a fraction of the voltage vector input length. The number of solves required is reduced substantially with very little loss in accuracy.

## 6. CONCLUSIONS AND FUTURE WORK

We have demonstrated a promising new method for sparsifying dense coupling matrices, which could be used to speed up circuit simulation of substrate coupling. We also showed how to obtain the sparse approximate coupling matrix quickly by combining many solves into one. The results show considerable improvements in sparsity with only small loss in accuracy. This algorithm can be applied to accelerate a variety of substrate extraction algorithms. Ideally we would like to have an analysis, analogous to that for the multipole or wavelet algorithms for the  $1/r$  kernel, giving error bounds for the new algorithm.

## 7. ACKNOWLEDGEMENTS

J. Kanapka and J. White would like to acknowledge the support of an IBM fellowship and grants from SRC and MARCO.

Example	Contacts	solve speedup ratio	$L_2$ error
Regular	1024	2.94	$2.3 \times 10^{-4}$
Irregular	1199	2.86	$7.3 \times 10^{-4}$
Regular	4096	8.03	$1.4 \times 10^{-4}$

Table 3: Solve-reduction technique performance

## 8. REFERENCES

- [1] D. K. Su, M. Loinaz, S. Masui, and B. Wooley, "Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits," *IEEE Journal Solid-State Circuits*, vol. 28, no. 4, pp. 420–430, April 1993.
- [2] F. J. R. Clement, E. Zysman, M. Kayal, and M. Declercq, "LAYIN: Toward a global solution for parasitic coupling modeling and visualization," in *Proc. IEEE Custom Integrated Circuit Conference*, May 1994, pp. 537–540.
- [3] N. K. Verghese, T. Schmerbeck, and D. J. Allstot, *Simulation Techniques and Solutions for Mixed-Signal Coupling in ICs*, Kluwer Academic Publ., Boston, MA, 1995.
- [4] T. Smedes, N. P. van der Meijs, and A. J. van Genderen, "Extraction of circuit models for substrate cross-talk," in *Proc. IEEE International Conference on Computer Aided Design*, November 1995, pp. 199–206.
- [5] R. Gharpurey and R. Meyer, "Modeling and analysis of substrate coupling in integrated circuits," *IEEE Journal Solid-State Circuits*, vol. 31, no. 3, pp. 344–353, March 1996.
- [6] K. Nabors, S. Kim, and J. White, "Fast capacitance extraction of general three-dimensional structure," *IEEE Trans. on Microwave Theory and Techniques*, vol. 40, no. 7, pp. 1496–1507, July 1992.
- [7] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3D structures," *IEEE Trans. CAD*, pp. 1059–1072, 1997.
- [8] S. Kapur and D. Long, "IES<sup>3</sup>: a fast integral equation solver for efficient 3-D extraction," in *Proc. IEEE International Conference on Computer Aided Design*, November 1997, pp. 448–455.
- [9] J. Tausch and J. White, "A multiscale method for fast capacitance extraction," in *Proceedings of the 36th Design Automation Conference*, New Orleans, LA, 1999, pp. 537–542.
- [10] R. Gharpurey and S. Hosur, "Transform domain techniques for efficient extraction of substrate parasitics," in *Proc. IEEE International Conference on Computer Aided Design*, November 1997, pp. 461–467.
- [11] M. Chou, *Fast Algorithms for Ill-Conditioned Dense-Matrix Problems in VLSI Interconnect and Substrate Modeling*, Ph.D. thesis, Massachusetts Institute of Technology, 1998.
- [12] J. P. Costa, M. Chou, and L. M. Silveira, "Precorrected-DCT techniques for modeling and simulation of substrate coupling in mixed-signal IC's," in *Proc. IEEE International Symposium on Circuits and Systems*, May 1998, vol. 6, pp. 358–362.
- [13] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, "Wavelet-like bases for the fast solution of second-kind integral equations," *SIAM J. Sci. Comput.*, vol. 14, no. 1, pp. 159–184, 1993.
- [14] P. Concus, G. H. Golub, and D. P. O'Leary, "A generalized conjugate gradient method for the numerical solution of elliptic PDE," in *Sparse Matrix Computations*, J. R. Bunch and D. J. Rose, Eds., pp. 309–332. Academic Press, New York, 1976.