Characterizing Substrate Coupling in Deep-Submicron Designs

Luis Miguel Silveira and Nuno Vargas
Technical University of Lisbon

The accurate modeling of noise-coupling effects caused by crosstalk through the substrate is an increasingly important concern for design and verification of analog, digital, and mixed systems. With the technique described here, designers can efficiently extract accurate substrate-coupling parameters from deep-submicron designs.

THE INDUSTRY TREND of integrating higher levels of circuit functionality in chips designed for compact consumer electronic products and the widespread growth of wireless communications have triggered the proliferation of mixed analog-digital systems. Single-chip designs combining digital and analog blocks built over a common substrate feature reduced levels of power dissipation, smaller package counts, and smaller package interconnect parasitics. Designing such systems, however, is becoming increasingly difficult owing to coupling problems resulting from the combined requirements for high-speed digital and high-precision analog components.

Noise coupling caused by the common chip substrate's nonideal isolation contributes significantly to the coupling problem in mixed-signal designs.[1,2] Fast-switching logic components inject current into the substrate, causing voltage fluctuation. Because substrate bias strongly affects the transistor threshold voltage, voltage fluctuations can affect the operation of sensitive analog circuitry through the body effect. Figure 1a illustrates this coupling mechanism, in which a switching digital node injects current into the substrate (currents J_1 and J_2 are drawn to ground, but J_2 affects the analog transistor bulk potential), causing the local substrate potential V_b to vary at an analog node. Figure 1b illustrates this interaction from the circuit viewpoint. Other known mechanisms for current injection into the substrate include hot-carrier injection and parasitic bipolar transistors.[2] The effects of substrate coupling largely depend on the layout specifics. Therefore, accurate analysis of these effects is possible only after extraction of the circuit features and the parasitics.

As technology and circuit design advance, substrate noise is beginning to plague even fully digital circuits. In these circuits, the cumulative effect of thousands or millions of logic gates changing state across the chip causes current pulses that are injected and absorbed into the substrate. Those currents are then transmitted to power and ground buses through direct feedthrough and load charge and discharge. Such couplings are highly destructive because pulsing currents, partially injected into the substrate through impact ionization and capacitive coupling, can be broadcast over great distances and picked up by sensitive circuits through capacitive coupling and the body effect. The resulting threshold voltage modulation dynam-

0740-7475/02/$17.00 © 2002 IEEE

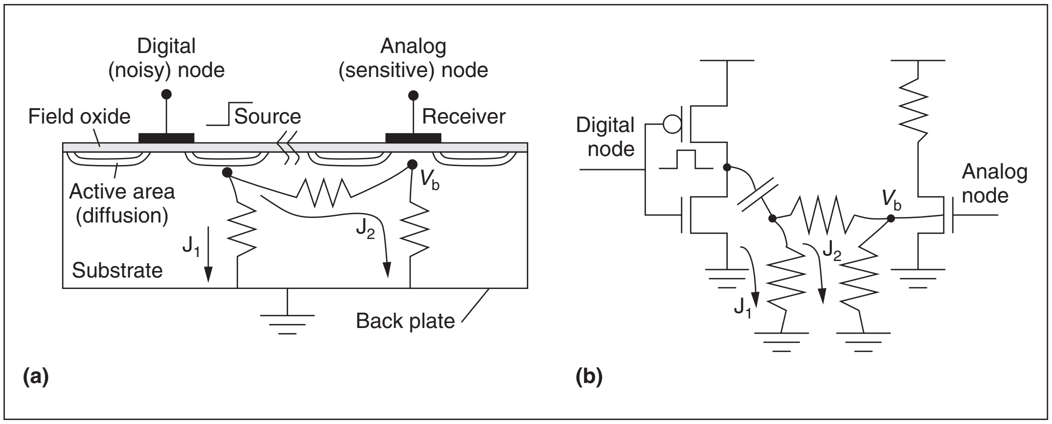

Figure 1. Substrate coupling: mechanism (a) and circuit (b). (The ≈ symbol indicates that we are showing only a portion of the substrate.)

ically changes gate delays locally, affecting performance unpredictably. Switching noise is especially detrimental to dynamic logic, memories, and embedded analog circuits such as phase-locked loops, wreaking havoc in otherwise functional circuits.

The reliable verification of analog, digital, and mixed-signal circuits requires an accurate characterization and a model of the design's substrate-coupling effects. Here, we present a method for accurately modeling substrate-coupling effects in deep-submicron designs. We use a numerical formulation obtained through finite-difference discretization of the substrate medium. We then solve the set of equations describing the system with a multilevel method that speeds up the computation of the model. Although we assume certain approximations, the finite-difference formulation is general and can be extended for future technologies.

Background

The naive but still common way to handle substrate coupling is costly trial and error. Clearly, such a methodology, which requires fabricating multiple versions of a design and relies heavily on the designer's expertise and experience, is not adequate in the face of rising fabrication costs and increasing demands for shorter design cycles. Researchers have attempted several methods of quantifying substrate noise-coupling effects to avoid expensive redesigns and multiple fabrication runs. The simplest of these methods use heuristic rules to determine the most relevant substrate couplings and to generate a simplified model of those couplings.[1,3] Such techniques reduce verification time, but extracting the correct couplings is difficult. Even mature pattern- and formula-based capacitance extraction tools, under development for the better part of two decades, commonly produce errors that in certain cases can be as high as 50% different from the true value.

Formula-based substrate extraction tools are far less mature, and their reliability is at least questionable. A subtler problem is that such tools are usually designed to extract only the most relevant couplings. This information can be useful to circuit designers because it lets them quickly identify and then remedy the dominant sources of coupling. However, such information is less useful from the verification standpoint. One of the main goals of performing detailed analysis is to obtain the accurate answers needed to verify design decisions. In the substrate-coupling context, this means that a tool must predict the amount of residual coupling remaining after the dominant sources are removed. Such sources are, by definition, second order, and without accurately predicting their magnitude, a designer cannot verify the design's correct operation. Unfortunately, most heuristic formulations completely neglect such couplings and may incorrectly judge a circuit as having no relevant coupling.

At the other end of the spectrum, methods that use an appropriate formulation of the substrate's electromagnetic interactions and rely on

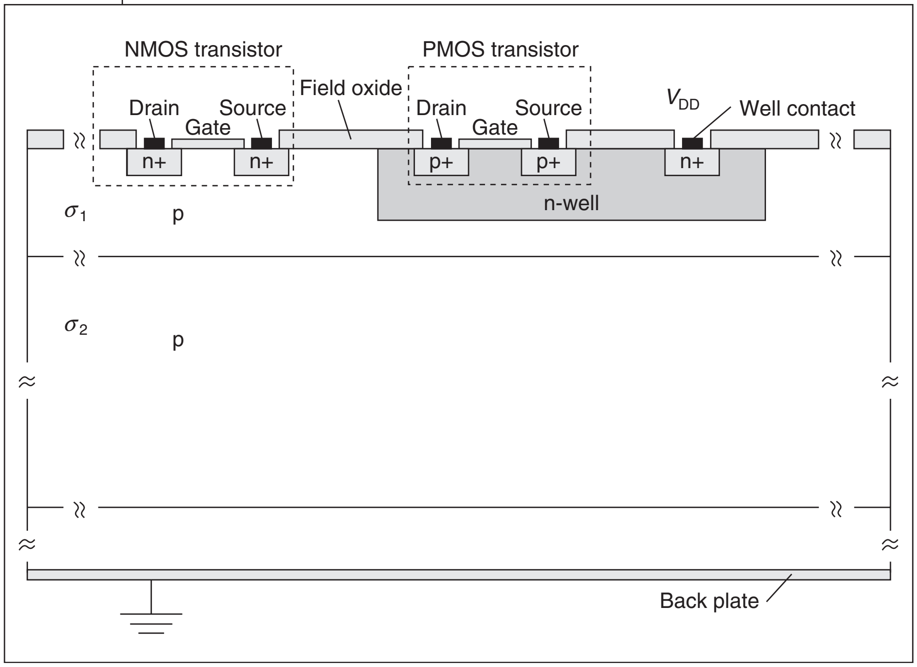

Figure 2. Typical substrate profile. This substrate contains two layers with different doping concentrations, rusulting in conductivities σ_1 and σ_2; n+ and p+ indicate doping regions.

detailed numerical analysis are usually accurate and can estimate all the possible couplings through the substrate. Device simulators such as Medici and Pisces are appropriate tools for this task. However, these methods are generally too slow because they simulate the drift-diffusion phenomenon in semiconductors, whereas we are interested only in circuit-level effects. Thus, such methods aren't efficient or versatile enough for implementation in standard CAD systems. In fact, they can be used only to analyze small portions of a design, including only a few devices.

Boundary-element methods have been applied with some success to the problem of modeling substrate coupling.[3-6] By requiring only the discretization of the relevant boundary features, these methods lead to smaller matrix problems. However, the matrices they produce are dense, limiting their use to small to medium problems. Therefore, speeding up the computations in boundary-element formulations is crucial to obtaining accurate models for large substrate-coupling problems. In recent years, researchers have devoted much work to this goal with encouraging results.[4,6]

Methods based on differential equations, such as finite-element and finite-difference numerical methods (like the one described here), can compute all the currents and voltages in the substrate, given a pattern of injected currents.[1,7,8] These techniques perform a full domain discretization on the large but bounded substrate volume and can easily handle irregular substrates (such as wells or doping profiles). Because these methods rely on volume meshing of the entire substrate, the number of unknowns resulting from the discretization can easily become very large. However, the resulting matrices are extremely sparse; thus, these methods, with appropriate solution algorithms, are a competitive option for substrate-model extraction in large, dense designs.

The cost of extracting a substrate model, though large, is a penalty paid only once. After extracting the model, designers can evaluate it many times to analyze substrate-coupling effects. For the most part, designers will use such models in standard circuit simulators such as Spice or Spectre. Desirable model characteristics, therefore, include easy incorporation in standard circuit simulators, high accuracy, and low evaluation cost.

Problem formulation

Figure 2 shows the profile of a typical substrate. We assume that the substrate is a stratified medium composed of several homogeneous conductive layers. A deposition process using appropriate materials builds devices on top of these layers. Ports or contacts at the top of the stack of layers correspond to the highly doped, strongly conductive areas where the circuit interacts with the substrate. Back plate contacts can improve isolation but increase the design cost.

In ICs, for frequencies up to a few GHz, the wavelengths of the magnetic fields far exceed a typical die's dimensions. Thus, we can assume a quasi-static approximation. This approximation is acceptable for current mixed-signal designs and technologies but may soon need to be revised. Under this approximation, we derive the following equation for the electric field's behavior:

$$\varepsilon(\partial/\partial t)(\nabla \bullet E) + \sigma\nabla \bullet E = 0 \qquad (1)$$

where E is the electric field, ε is the dielectric constant, and σ is the medium's conductivity (assumed constant per layer).[7] We derived this equation from basic principles—namely, Maxwell's equations, which describe the behavior of the electric and magnetic fields. Here we consider only the electric field.

A simple, but not unique, way to solve equation 1 is to perform a spatial discretization of the substrate volume and approximate the electric field vector between adjacent nodes in this 3D grid, using a finite-difference operator. That is, $E_{ij} = (V_i - V_j)/h_{ij}$, where h_{ij} is the distance between adjacent nodes i and j in the volume grid. Using, for instance, a standard seven-point stencil leads to

$$\sum_j \left[G_{ij}\left(V_i - V_j\right) + C_{ij}\left(\frac{\partial V_i}{\partial t} - \frac{\partial V_j}{\partial t}\right)\right] = 0 \qquad (2)$$

where G_{ij} and C_{ij} are functions of ε, σ, and the box dimensions.[7]

Equation 2 can be modeled as a simple linear network of lumped circuit elements, organized as a three-dimensional grid, as shown in Figure 3. To compute the model, we set boundary conditions to indicate substrate connection nodes and the substrate's physical limits. Hence, we treat active areas (contacts, devices, and possibly the back plate) as Dirichlet boundaries for voltages, with constant fixed voltages, whereas we impose Neumann boundary conditions or reflective conditions for both voltage and current on all other boundaries. For typical values of ε and σ, the substrate's dielectric relaxation time is on the order of tens of picoseconds, much smaller than the circuit's typical time scales. Thus, it is reasonable to neglect intrinsic substrate capacitances for operation frequencies up to a few gigahertz.

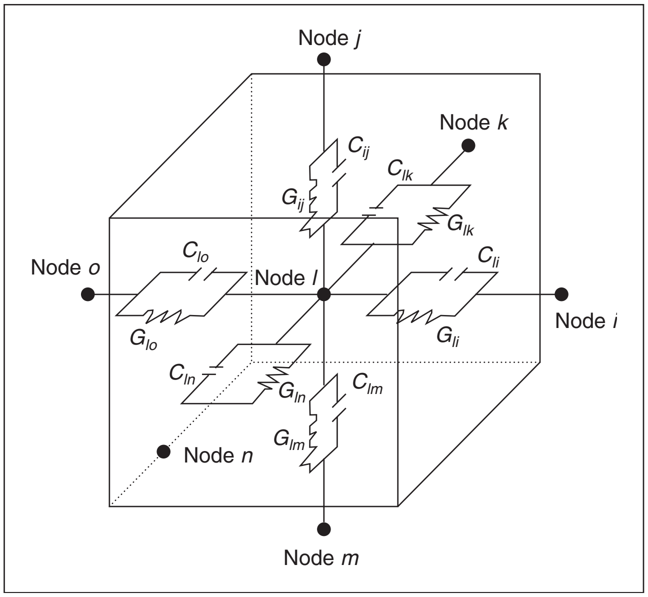

Figure 3. Model of the substrate as a 3D mesh of resistances and capacitances connecting nodes in the mesh.

Experimental comparisons conducted with detailed device simulators have confirmed this approximation's validity for frequencies up to the gigahertz range,[8] although it may also need revision as the technology evolves or lossy substrates are used. Nevertheless, the final model used in simulations takes the capacitive effects of junctions, wells, and other parasitic elements into account. However, it does so outside the substrate extraction procedure, using traditional parasitic extraction methodologies. We later tie in those parasitic elements with the circuitry and the extracted substrate model to perform circuit-level verification. This strategy lets us model some frequency-dependent effects related to substrate coupling.

As described, the model doesn't include the parasitic-device-capacitance effects associated with device junctions, which are estimated or computed offline, using an interconnect extraction tool, or directly included in the device models used and in the nonlinear simulator. However, it does account for the linear field-oxide and depletion capacitances of biased wells. If the capacitances introduced to the substrate by the depletion regions of well diffusions and interconnects over the field oxide are also modeled as lumped circuit elements outside the mesh, we can simplify equation 2 and model the substrate as a purely resistive mesh

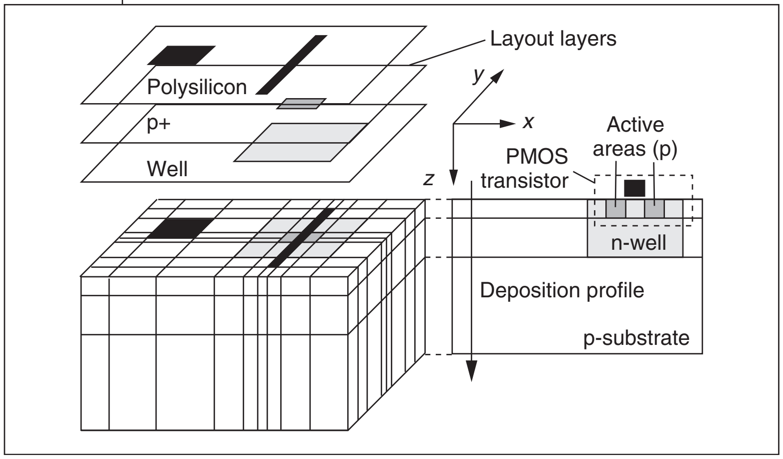

Figure 4. Example of how layout and technology information helps in parameterizing the gridding procedure. The location of features on various layers (polysilicon, well, and p+) determines device locations and types. (Source: Clement et al.[8])

(that is, its macromodel is simply a conductance matrix). Several authors have used this approach successfully.[7,8] Using nodal analysis, we can formulate a set of equations describing the circuit's equilibrium conditions:

$$\boldsymbol{Y}_g \boldsymbol{V}_g = \boldsymbol{I}_g \tag{3}$$

where $\boldsymbol{V}_g, \boldsymbol{I}_g \in \Re^n$ are the vector of potentials and the vector of injected currents at all grid nodes. $\boldsymbol{Y}_g \in \Re^{n \times n}$ is the underlying circuit's admittance representation obtained after discretization, and n is the number of grid nodes with unknown potentials.

Coupling this model directly into a standard simulator with the remaining, possibly nonlinear, circuit elements is straightforward. Unfortunately, that approach is prohibitive in terms of CPU and memory because the 3D-mesh circuit can be very large. From the standpoint of the electrical circuit, only the substrate nodes that directly contact it are relevant. Therefore, computing a macromodel that implicitly encapsulates the substrate mesh's internal node behavior is sufficient. This macromodel's size is thus the number of substrate contacts, which is far smaller than the total number of mesh nodes. The macromodel's size is also independent of the discretization, an important feature from the simulation standpoint. The model can be included in a circuit simulator such as Spice or Spectre in a straightforward manner.

Although the electric field varies nonlinearly as a function of distance, the finite-difference method approximates this variation as a piecewise constant function. We control the approximation's accuracy by carefully choosing the pitch of the grid used to generate the 3D mesh. In areas where the electric field varies rapidly, such as near devices or contacts of depletion regions, finer grids are necessary to accurately approximate the nonlinear behavior. Elsewhere, we can use coarser grids to reduce the overall number of grid points—the Achilles' heel of volume-based methods.

We construct our adaptive, nonuniform mesh on the basis of layout information about the circuit being analyzed, as well as doping profiles of the circuit technology. The layout defines the mesh in the *x*-*y* plane, and the vertical doping profiles, which are assumed known and can be characterized a priori, determine the meshing in the *z* direction. Figure 4, borrowed from Clement et al.,who describe a simple 3D-mesh generation process, shows a gridding strategy based on layout information, the location and type of devices, and other technology characteristics.[8] Layout and technology information are crucial to the extraction process because they determine the exact location and type of substrate contacts. With this information, we generate a coarse 3D mesh and formulate the set of equations that describes the problem in terms of voltages and currents and then solves that system.

Evaluation of the result may necessitate refining the grid in particular spots and reevaluating it. By iteratively repeating this procedure, we obtain an appropriate mesh with an acceptable computation time for most circuits. Other types of mesh refinement, based, for instance, on direct analysis of the layout information and estimation of the electric field gradients, are also possible. These meshing algorithms are simple, but they sometimes generate unnecessarily fine meshes in certain areas. Previous work, however, shows that approximating a linear substrate model with contacts at the surface and possibly in the back plate makes it easy to determine a priori an appropriate grid density to guarantee appropriate accuracy in the model.[7]

Model computation

Given a set of m substrate contacts, we seek a model, $\boldsymbol{Y}_c \in \Re^{m\times m}$, which relates the currents on those contacts, $\boldsymbol{I}_c \in \Re^m$, to their voltage distribution, $\boldsymbol{V}_c \in \Re^m$:

$$\boldsymbol{I}_c = \boldsymbol{Y}_c \boldsymbol{V}_c$$

The matrix $\boldsymbol{Y}_c$ represents the macromodel we hope to obtain from the 3D-mesh circuit to use in a circuit simulator (m is usually much smaller than n; indeed, m is at worst on the order of the number of nodes on the substrate's top surface, whereas n is the number of nodes in the 3D grid). $\boldsymbol{V}_c$ and $\boldsymbol{I}_c$ are subsets of their mesh counterparts $\boldsymbol{V}_g$ and $\boldsymbol{I}_g$ and can easily be derived from them. To understand this, assume that the 3D grid nodes are numbered as follows: nodes 1, …, m are the nodes corresponding to the substrate contacts, and nodes $m + 1$, …, n are the internal grid nodes. Clearly, $\boldsymbol{V}_g = [\boldsymbol{V}_c; \boldsymbol{V}_i]$, where $\boldsymbol{V}_c \in \Re^m$ is the currents on the substrate contacts, and $\boldsymbol{V}_i \in \Re^{n-m}$ is the vector of potentials at all internal grid nodes.

Equation 3, therefore, can be written as

$$\boldsymbol{I}_g = \begin{bmatrix} \boldsymbol{I}_c \\ \boldsymbol{I}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{cc} & \boldsymbol{Y}_{ci} \\ \boldsymbol{Y}_{ic} & \boldsymbol{Y}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_c \\ \boldsymbol{V}_i \end{bmatrix} = \boldsymbol{Y}_g \boldsymbol{V}_g$$

Because only contact nodes can have currents injected into them, $\boldsymbol{I}_i = 0$, and thus

$$\boldsymbol{I}_c = \underbrace{\left[\boldsymbol{Y}_{cc} - \boldsymbol{Y}_{ci} \boldsymbol{Y}_{ii}^{-1} \boldsymbol{Y}_{ic}\right]}_{\boldsymbol{Y}_c} \boldsymbol{V}_c \qquad (4)$$

Equation 4 shows that the resulting model is simply an admittance matrix, and we can easily include it in standard circuit simulators to perform coupled-substrate simulation. From equation 4, however, it also appears that computing the substrate macromodel $\boldsymbol{Y}_c$ requires inversion of $\boldsymbol{Y}_{ii}$, an extremely large matrix. This procedure requires a number of computations that grows roughly with n^2 and is therefore too expensive because n may be very large. Alternatively, we can compute $\boldsymbol{Y}_c$ one column at a time by appropriately setting the voltages at the contacts $\boldsymbol{V}_c$, which are boundary conditions for the mesh problem.

For instance, suppose that the voltage is set to 1 volt on one contact and 0 on all others; in other words, $\boldsymbol{V}_c$ has a single nonzero entry,

$$V_{c_i} = 1, V_{c_j} = 0, \forall j \neq i$$

Then we can compute a Norton equivalent for all nodes connected to that contact. We pack the resulting current sources into $\boldsymbol{I}_g$ appropriately and use equation 3 to solve for $\boldsymbol{V}_g$. From knowledge of the voltages at all grid nodes, we can obtain the currents flowing into the contacts.

Given the form of $\boldsymbol{V}_c$, it is then clear that $\boldsymbol{I}_c$ will equal the ith column of $\boldsymbol{Y}_c$. By repeating this procedure for every contact, we can compute $\boldsymbol{Y}_c$ one column at a time. The algorithm is similar to the standard capacitance extraction problem. Extracting the full model for a system with m contacts requires m linear solutions.

However, because $\boldsymbol{Y}_g$ is the result of a 3D volume meshing, computing the inverse or system solution with a direct method such as Gaussian elimination has at best a cost on the order of n^2, even with reordering and sparse-matrix techniques. For a large n, this cost is prohibitive and would restrict the applicability of this method to small problems or very coarse discretizations.

Iterative methods—for example, Krylov-subspace-based iterative algorithms such as the Generalized Minimal Residual (GMRES) algorithm for solving nonsymmetric linear systems and the Conjugate Gradient (CG) algorithm—offer interesting alternatives to the direct-solution methods. A CG variant using an incomplete Cholesky preconditioner, ICCG, has been applied to the substrate problem with fairly good results.[7] These methods' computational cost per iteration and their memory requirements for the matrix grow linearly with n. Their convergence rate roughly depends on $k^{1/2}$, where k is the condition number of matrix $\boldsymbol{Y}_g$.

Multilevel methods

Formulations based on unbounded operators, such as those resulting from volume meshing of the substrate, generate ill-conditioned linear systems whose condition numbers grow with mesh refinement. When Krylov-subspace algorithms are applied to such problems, con-

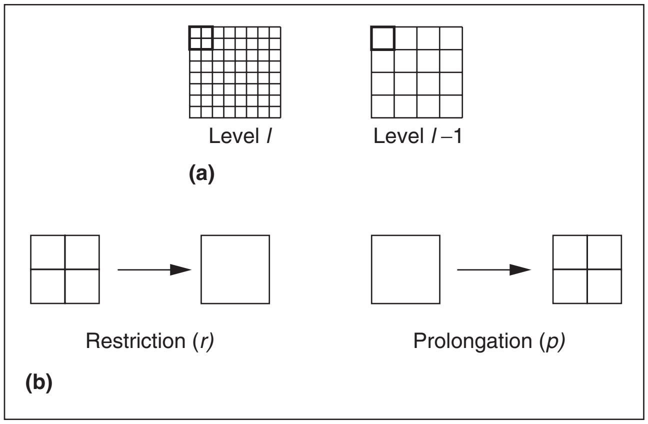

Figure 5. Uniform discretizations (a) produce a fine-grid representation at level l and a coarse-grid representation at level l – 1. The intergrid-transfer operators (b) move data across levels. Restriction operator r changes the grid from level l to level l – 1 (from fine to coarse); prolongation operator p performs the inverse operation.

vergence can become painfully slow and hundreds of iterations (and thus matrix-vector products) may be required per solution. The source of ill conditioning in the linear system is the coexistence of eigenmodes with distinct, characteristic length scales that these methods cannot distinguish.

To overcome this difficulty, we can use multigrid methods, or more generally, multilevel methods, to solve the linear system. Their fast convergence makes multigrid methods the most efficient iterative techniques for solving elliptic partial differential equations.[9,10] They attempt to remove ill conditioning by analyzing the problem at each length scale independently.

These methods operate by decomposing the original problem into a set of subproblems, each associated with a specific length scale, or level. Then, a relaxation, or smoothing, scheme is applied to each subproblem to reduce error components at that length scale. The subproblems communicate with one another through restriction and prolongation operators, collectively called intergrid-transfer operators. The work associated with relaxation at each level decreases geometrically as the problem coarsens. Therefore, a multiple of the work at the finest level bounds the total work required for going through each level once, or for one multigrid sweep.

Furthermore, because the relative error reduction resulting from a relaxation iteration at each level is uniform across all levels, the error reduction for a multigrid sweep equals the error reduction at a single level. Hence, the multigrid convergence rate is independent of discretization; this is the main reason for the efficiency of such methods.

To simplify the description of the method without loss of generality, let us assume that we discretize the entire substrate volume, Ω, using a uniform array of $M \times M \times M$ boxes (clearly this restriction is neither necessary nor convenient in most cases). Furthermore, we assume that $M = 2^l$ for some integer l. The number of cell unknowns, and hence the size of the linear system in equation 3, is then $n = N_l = M^3$. We refer to this discrete system as a level-l, or fine-grid, representation of the problem and denote the problem as

$$\boldsymbol{Y}_{\{l\}} \bullet \boldsymbol{V}_{\{l\}} = \boldsymbol{I}_{\{l\}} \qquad (5)$$

Suppose we also discretize the substrate volume using a coarser, uniform $(M/2) \times (M/2) \times (M/2)$ array of panels, yielding a discrete linear system of size $N_{l-1} = (M^3/8)$. This gives a representation of level $l - 1$, or a coarse-grid representation:

$$\boldsymbol{Y}_{\{l-1\}} \bullet \boldsymbol{V}_{\{l-1\}} = \boldsymbol{I}_{\{l-1\}}$$

Figure 5 shows fine- and coarse-grid representations. Solving the fine-grid problem (equation 5) with direct matrix factorization is prohibitive for a large N_l because $\boldsymbol{Y}_{\{l\}}$ is very large. However, it may be possible to factor the smaller matrix $\boldsymbol{Y}_{\{l-1\}}$ corresponding to the coarse-grid problem since $N_{l-1} = N_l/8$.

This leads to a two-grid method, which solves the problem iteratively at level l with the help of direct solution at level $l - 1$. As we mentioned, the two principal algorithmic components needed are the smoothing operator and the intergrid-transfer, or restriction and prolongation operators pictorially represented in Figure 5b. Our two-grid iteration for solving equation 5 smooths the error in the *k*th iteration, $\boldsymbol{V}_{\{l\}}^{(k)}$, by carefully solving a series of local problems. This first stage, fine-grid smoothing or relaxation, results in an intermediate guess $\boldsymbol{V}_{\{l\}}{}^{*}$. (The asterisk denotes a guess.)

Next, we compute the residual $\boldsymbol{u}_{\{l\}} = \boldsymbol{Y}_{\{l\}} \bullet \boldsymbol{V}_{\{l\}}{}^{*} - \boldsymbol{I}_{\{l\}}$ and project it onto the coarse grid via $\boldsymbol{u}_{\{l-1\}} = r\boldsymbol{u}_{\{l\}}$, where *r* is a restriction operator. Then we explicitly solve the coarse-grid problem $\boldsymbol{Y}_{\{l-1\}} \bullet (\Delta\boldsymbol{V}_{\{l-1\}}) = \boldsymbol{u}_{\{l-1\}}$ for $\Delta\boldsymbol{V}_{\{l-1\}}$, and project the result onto the fine grid via $\Delta\boldsymbol{V}_{\{l\}} = p(\Delta\boldsymbol{V}_{\{l-1\}})$, where *p* is a prolongation operator. Finally, the intermediate guess on the fine grid is updated to yield the $(k+1)$st iteration $\boldsymbol{V}_{\{l\}}^{(k+1)} = \boldsymbol{V}_{\{l\}}{}^{*} - \Delta\boldsymbol{V}_{\{l\}}$. This second stage, coarse-grid correction, is responsible for long-range interactions.

We repeat the fine-grid-smoothing and coarse-grid-correction cycle until the norm of residual $\boldsymbol{u}_{\{l\}}$ is below some tolerance. Fine-grid smoothing reduces only the high-frequency components of the error, and the resulting smoothed error is well represented on the coarse grid where the explicit solution is performed. Therefore, the two-grid scheme effectively decouples the original problem into high- and low-frequency subproblems.

The multigrid method generalizes this two-grid scheme to an arbitrary number of levels. Instead of solving the problem explicitly at level $l-1$, which may still be too expensive, we apply a similar smoothing-correction cycle at level $l-1$. In the same manner, the correction cycle at level $l-1$ becomes a smoothing-correction cycle at level $l-2$, and so on recursively. The substrate volume is now discretized at all levels ($l_{\max}$, …, $l_{\min}$), as Figure 6 shows. Only at the coarsest level, $l_{\min}$, is the system $\boldsymbol{Y}_{\{l\min\}} \bullet \boldsymbol{V}_{\{l\min\}} = \boldsymbol{I}_{\{l\min\}}$ solved explicitly. Such an algorithm is easily described by a recursive function.[10]

As we have said, we require intergrid-transfer operators *r* and *p* between the various grids. Fortunately, constructing such operators is trivial. The restriction operator is simply a local operation in which nearby node values are averaged to obtain the restriction value. For every node, we average all its nearest neighbors by setting an operator as a sparse matrix whose structure reflects the nearest neighbors used for averaging and then multiplying that matrix by the vector of node voltages. Thus, we average all nodes at once. For the prolongation operator, we use standard trilinear interpolation on the coarse-grid nodes to produce the error corrections at the finer-grid level.

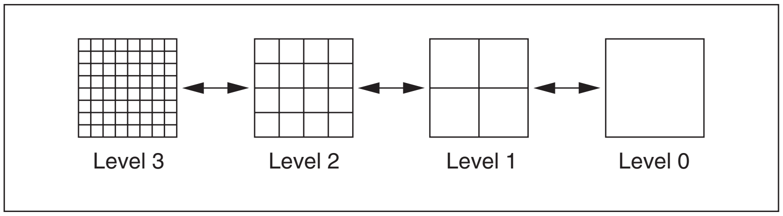

Figure 6. A 2D representation of multilevel discretization.

Experimental and computational results

A simple example shows how the multigrid algorithm outperforms Krylov methods when applied to the extraction of a substrate model using a volume formulation. Figure 7a (next page) shows the layout of a three-stage ring oscillator with a nearby analog transistor.[1,5] We used this setup to analyze the effects of coupling through the substrate. We assumed that coupling occurs mainly through the transistor back gates and direct substrate contacts.

We generated a model for the substrate interaction using the techniques described in the preceding section. In generating the model, we considered the relevant features—namely the substrate connections, the well, and the diffusion areas of the ring oscillator and the analog transistor. Figure 7b shows a simplified diagram of some of these features.

Once we determined the layout features, we used geometric information about them to generate the model, using the technique described earlier. After the numerical computations, the substrate model consisted of the resistor network shown conceptually in Figure 7b.

Extracting the remaining parasitics and the electrical network produced the circuit shown in Figure 7c. We then used Spice to simulate two versions of the circuit—with and without the substrate model. The ring oscillator ran freely, and the analog transistor was biased to deliver a constant current in the absence of substrate noise. We then monitored the analog transistor's body terminal.

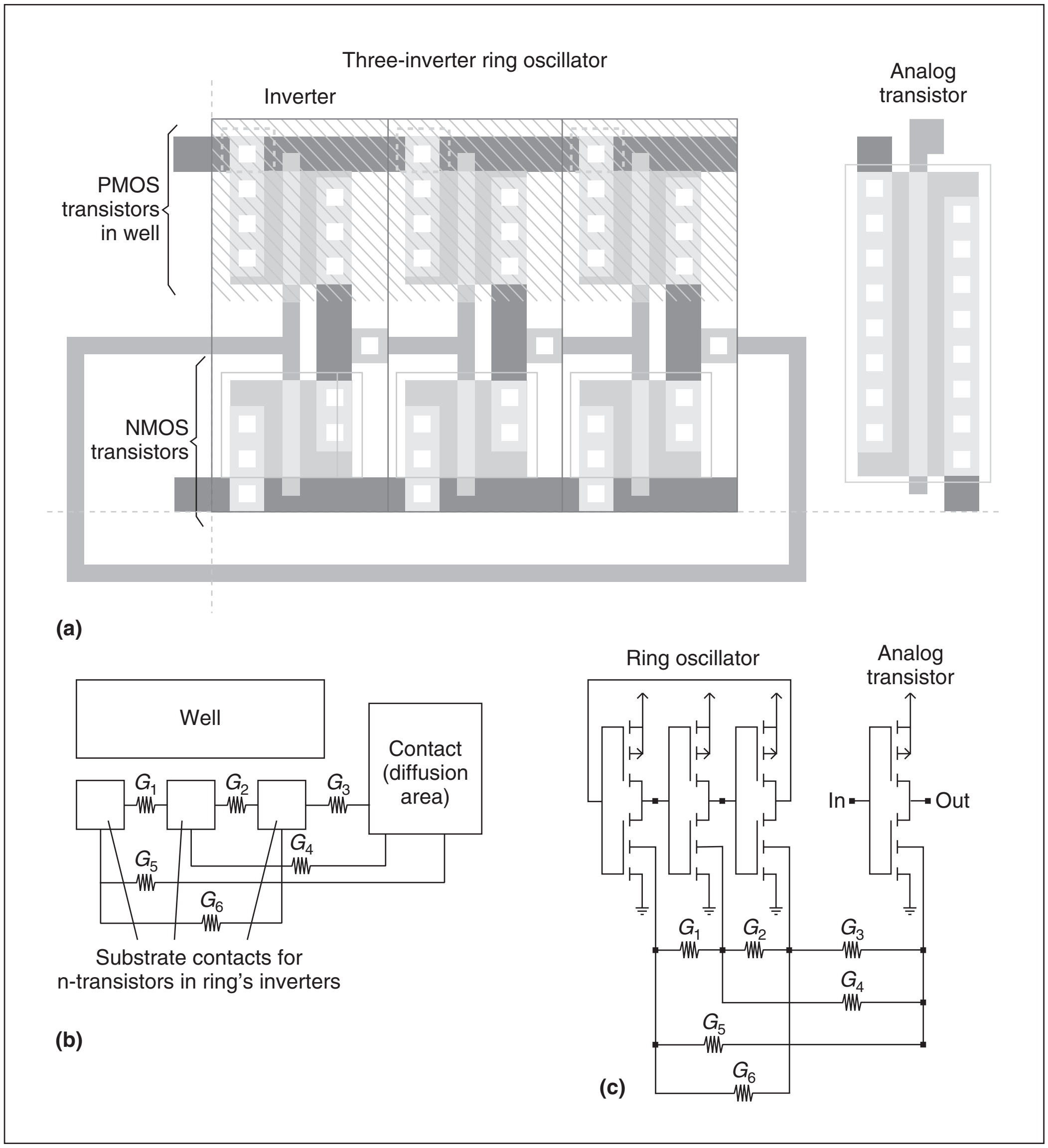

Figure 7. Layout of a small ring oscillator with a nearby analog transistor (a). Extracted substrate model of the example layout (b). For simplicity, couplings to and within the well are not shown. Electrical circuit used for verification of the substrate-coupling effects in the example circuit (c).

Figure 8 shows the simulation results, which are similar to those presented by Costa, Chou, and Silveira.[5] Figure 8a shows the oscillator output nodes' waveforms, which are as we would expect of a fast ring oscillator. When substrate coupling is not accounted for, the analog transistor's body terminal has a constant voltage. However, with the extracted substrate model inserted in the circuit, the voltage at the body terminal oscillates rapidly with significant amplitude, as Figure 8b shows. The peaks in the body terminal of the analog transistor's waveform correspond to the time periods during which the oscillator output nodes change more rapidly.

These oscillations cause threshold-voltage changes, which can degrade the transistor's performance considerably. For correct circuit verification, therefore, the extracted substrate models must be included in any simulation.

To extract the model used in the simulations, we attempted various discretizations and methods for solving equation 3. Figure 9 compares the

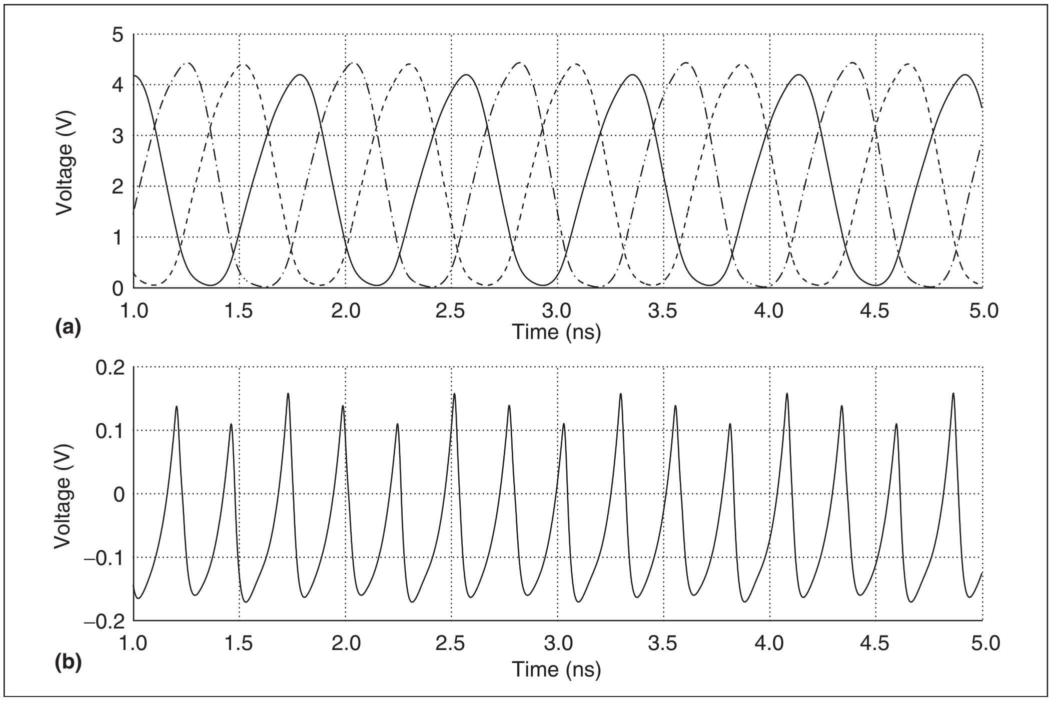

Figure 8. Simulation results: time-domain waveforms for the output nodes of the three inverters in the ring oscillator (a) and the analog transistor's body terminal (b). The differences between the three plots shown in (a) are insignificant.

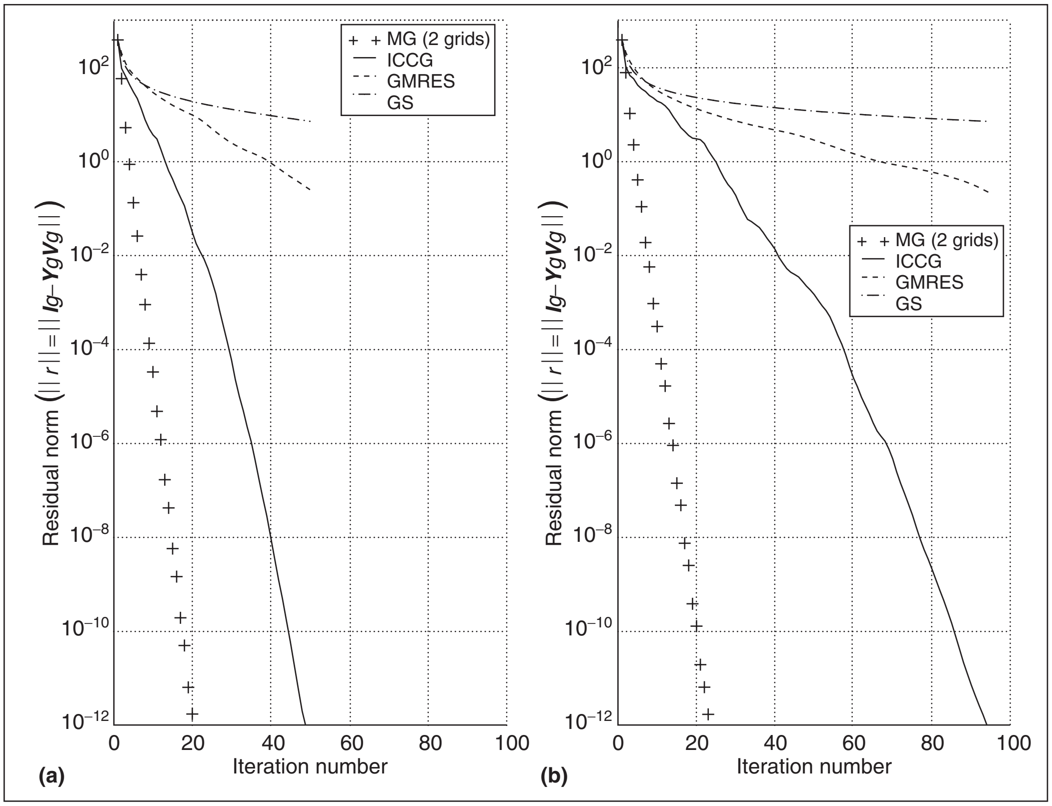

Figure 9. Comparison of convergence rates for the multigrid (MG), ICCG, GMRES, and Gauss-Seidel (GS) methods applied to the example problem. In (a) the discretization size is 17 × 17 × 9; in (b) it is 33 × 33 × 9.

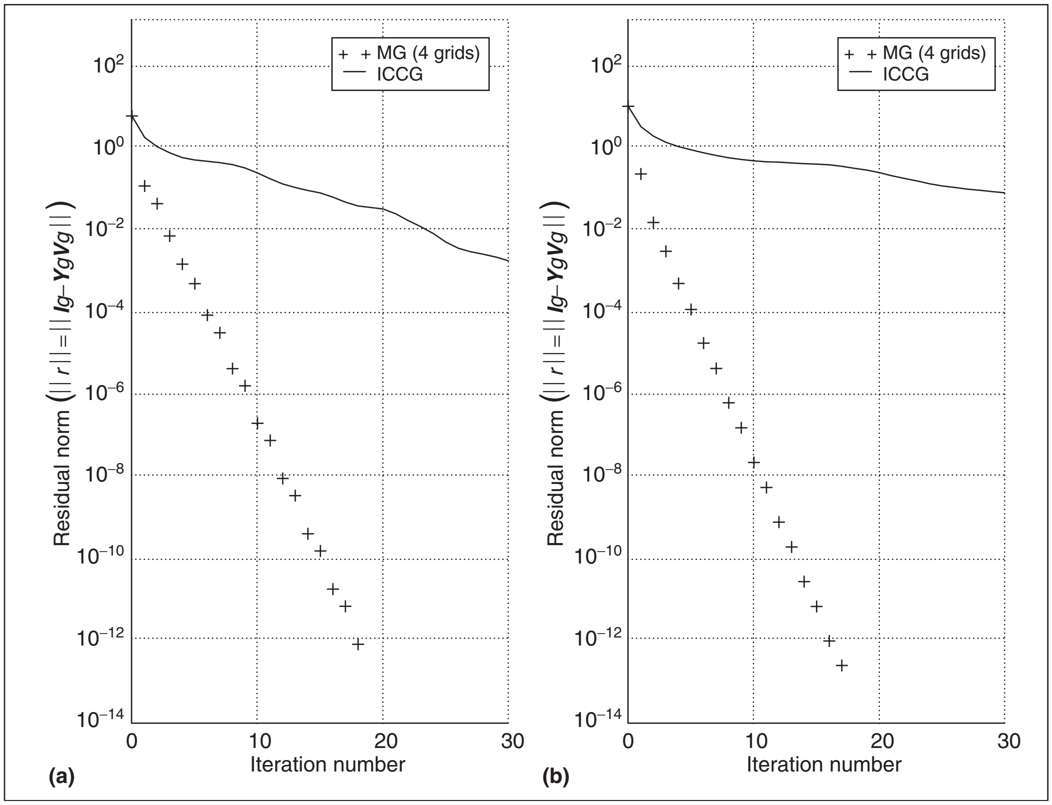

Figure 10. Comparison of convergence rates for the multigrid and ICCG methods applied to the example problem with grid sizes 33 × 33 × 17 (a) and 65 × 65 × 33 (b).

standard Gauss-Seidel (GS) relaxation method, GMRES, ICCG, and our multigrid method (using two grids), as applied to the extraction of the example layout in a small substrate. The figure shows the various algorithms' convergence rates with respect to the iteration number. To obtain the plot in Figure 9a, we used a discretization size of 17 × 17 × 9; for Figure 9b, the discretization size was 33 × 33 × 9. Both discretizations were nonuniform. As the figure shows, GS and GMRES converge very slowly, and are virtually useless. For the coarser discretization, both ICCG and our multigrid method converge fairly quickly, but multigrid is considerably faster. For the finer discretization, however, ICCG's convergence deteriorates considerably (twice as many iterations are required for convergence), whereas the multigrid method shows a convergence rate that seems fairly independent of the discretization size, as we expected.

This property is more evident when realistic discretizations are used for more complex examples, and the advantages of the multigrid algorithm are even more striking. For the comparisons in Figure 10, we reduced the grid size and used discretizations of 33 × 33 × 17 and 65 × 65 × 33. For such fine discretizations, ICCG deteriorated considerably, while the multigrid method (using four grids) showed little convergence deterioration. The multigrid method's behavior facilitates model extraction with higher accuracy in larger problems.

The technique described here can be easily implemented and used with standard extraction technologies. The models produced are accurate and lend themselves to direct inclusion in standard simulation and verification methodologies. Because the technique is still computationally intensive, it is probably not suitable for the intermediate verification steps in the design loop. However, it is efficient enough for use in the final verification steps for accurate characterization of substrate-coupling effects in analog, digital, and mixed-signal designs. ■

Acknowledgments

We thank Joel Phillips of Cadence Design Systems and Jacob White and Mike Chou of the Massachusetts Institute of Technology for many valuable insights. This work was partially supported by the Portuguese programs Praxis XXI and FEDER under contracts 2/2.1/T.I.T/1661/95 and 2/2.1/T.I.T/1639/95.

■ References

1. D.K. Su et al., "Experimental Results and Modeling Techniques for Substrate Noise in Mixed-Signal Integrated Circuits," *IEEE J. Solid-State Circuits*, vol. 28, no. 4, Apr. 1993, pp. 420-430.
2. R. Gharpurey and R.G. Meyer, "Modeling and Analysis of Substrate Coupling in Integrated Circuits," *IEEE J. Solid-State Circuits*, vol. 31, no. 3, Mar. 1996, pp. 344-353.
3. T. Smedes, N.P. van der Meijs, and A.J. van Genderen, "Extraction of Circuit Models for Substrate Cross-Talk," *Proc. Int'l Conf. Computer-Aided-Design* (ICCAD 95), IEEE CS Press, Los Alamitos, Calif., 1995, pp. 199-206.
4. N.K. Verghese, D.J. Allstot, and M.A. Wolfe, "Verification Techniques for Substrate Coupling and Their Application to Mixed-Signal IC Design, *IEEE J. Solid-State Circuits*, vol. 31, no. 3, , Mar. 1996, pp. 354-365.
5. J.P. Costa, M. Chou, and L.M. Silveira, "Efficient Techniques for Accurate Extraction and Modeling of Substrate Coupling in Mixed-Signal IC's," *Proc. Design, Automation and Test in Europe* (DATE 99), IEEE CS Press, Los Alalmitos, Calif., 1999, pp. 396-400.
6. M. Chou and J. White, "Multilevel Integral Equation Methods for the Extraction of Substrate Coupling Parameters in Mixed-Signal IC's," *Proc. 35th ACM/IEEE Design Automation Conf.* (DAC 98), ACM Press, New York, 1998, pp. 20-25.
7. B. Stanisic et al., "Addressing Substrate Coupling in Mixed-Mode IC's: Simulation and Power Distribution Systems," *IEEE J. Solid-State Circuits*, vol. 29, no. 3, Mar. 1994, pp. 226-237.
8. F.J.R. Clement et al., "Layin: Toward a Global Solution for Parasitic Coupling Modeling and Visualization," *Proc. IEEE Custom Integrated Circuit Conf.*, IEEE Press, Piscataway, N.J., 1994, pp. 537-540.
9. A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, vol. 31, no. 138, Apr. 1977, pp. 333-390.
10. W.L. Briggs, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.

Luis Miguel Silveira is an associate professor of electrical and computer engineering in the Instituto Superior Técnico (IST) of the Technical University of Lisbon, a researcher at the Systems and Computers Engineering Institute (INESC-ID) in Lisbon, and a founding member of the Cadence European Laboratories' Lisbon Center. His research interests include computer-aided IC design, with emphasis on parallel computer algorithms and theoretical and practical issues in numerical simulation methods. Silveira has engineer's and master's degrees in electrical and computer engineering from the IST, and MS, EE, and PhD degrees from the Massachusetts Institute of Technology. He is a senior member of the IEEE and a member of Sigma Xi.

Nuno Vargas is a software engineer and a consultant in the fields of networking, interactive television, and voice-over-IP solutions for Novabase SA, in Portugal. His scientific interests include mathematics and integrated circuits and systems. Vargas has a degree in electrical and computer engineering from the IST, Technical University of Lisbon, where he participated in the work described in this article.

■ Direct questions and comments about this article to Luis Miguel Silveira, INESC-ID/Cadence European Laboratories, Dept. of Electrical and Computer Eng., Instituto Superior Técnico, Technical University of Lisbon, Rua Alves Redol, 9, 136 Lisboa, 1000-029 Portugal; lms@inesc-id.pt.

For further information on this or any other computing topic, visit our Digital Library at http://computer.org/publications/dlib.