Simulating the Behavior of MEMS Devices: Computational Methods and Needs

STEPHEN D. SENTURIA, NARAYAN ALURU, AND JACOB WHITE
Massachusetts Institute of Technology

Technologies for fabricating a variety of MEMS devices have developed rapidly, but computational tools that allow engineers to quickly design and optimize these micromachines have not kept pace. Inadequate simulation tools force MEMS designers to resort to physical prototyping. To realistically simulate the behavior of complete micromachines, algorithmic innovation is necessary in several areas.

Micromachining technologies for fabricating tiny sensors, actuators, mechanisms, passive components, and even packages for components—collectively called *microelectromechanical systems* or MEMS—are being developed successfully throughout the world.[1] Examples of these fabrication technologies include bulk micromachining, surface micromachining, laser-etching, microelectrodischarge machining, 3D printing, wafer bonding, the dissolved-wafer silicon-on-glass process, LIGA and high-aspect-ratio metal plating, and micromolding.

These methods have been used to produce many types of MEMS products and prototype devices, including especially pressure sensors and accelerometers, but also scanning probe tips for atomic force microscopes and scanning tunneling microscopes, flow sensors, valves, micromotors, and chemical microanalysis systems on a chip.

Efficient MEMS simulation: Three main computational challenges

Figure 1 shows two examples of silicon microsensors fabricated by wafer bonding. At the left is an accelerometer for out-of-plane motion, and at the right is a resonant rate gyroscope for detecting rotations about an axis normal to the plane of the device. The action of these MEMS devices involves several physical effects: mechanical motion, air damping, electrostatic actuation, and capacitive position detection. Detailed knowledge of all of these effects is a prerequisite for effective and efficient design.

While many types of micromachines are indeed being created, design processes are cumbersome. It is fair to say that the ability of a MEMS engineer to design a new product is still limited in ways that were long ago overcome in integrated circuit engineering. Except for a few classes of pressure sensors, every new MEMS product idea is essentially a research project. During that research project, years can be spent developing prototypes which are then discarded. The result is an unacceptably long product development cycle or unnecessarily conservative design practices.

The first step in reducing design time and allowing for aggressive design strategies is to develop simulation tools that will let designers try “what if” experiments in hours instead of months. Several commercial and academic efforts are underway to devise such simulation systems.[2–7] To provide these tools, three computational challenges are being addressed.

First, faster algorithms are being developed for computing surface forces due to fields or fluids exterior to geometrically complex, flexible three-dimensional structures.

Second, since the performance of most micromachined structures is due to a complicated interaction between structural stresses, electrostatic or magnetic forces, and fluid tractions or pressures, approaches for coupling efficient domain-specific solvers are under investigation.

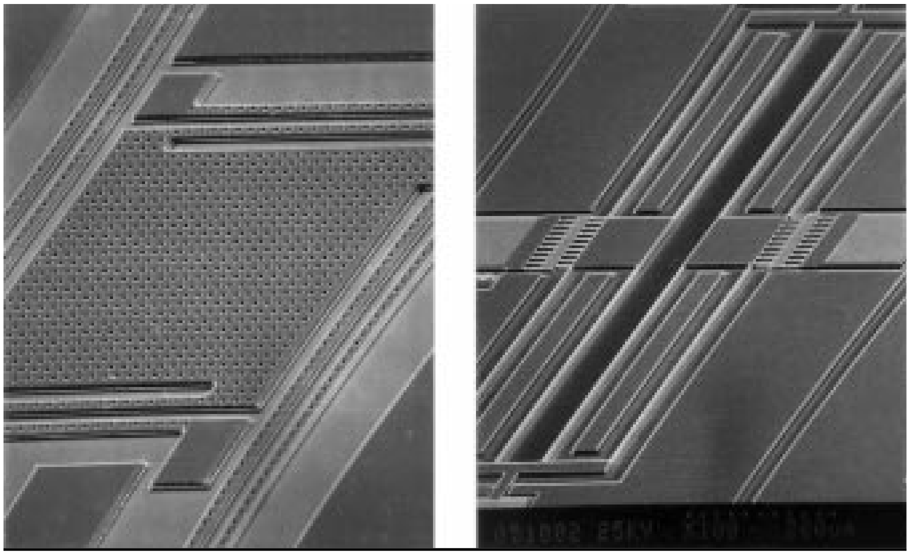

Figure 1. *Left:* silicon microaccelerometer. Notice the holes in the proof mass for control of air damping of the motion. *Right:* silicon resonant rate gyroscope. The comb drives are used to excite the structure into lateral resonance. Rotation causes out-of-plane motion, which is detected capacitively. (Courtesy C. Hsu and M. Schmidt, MIT)

Finally, because the ultimate concern is the functioning of the designed device when it is embedded in a complete system, designers need accurate dynamical models that permit rapid simulation of system performance under a wide variety of inputs and scenarios, such as being inserted into a feedback loop. And since direct simulation of 3D structures involves thousands of degrees of freedom, all coupled together, full nonlinear direct dynamic simulation can be computationally infeasible in a typical workstation environment. Furthermore, designers tend to think in terms of models with only a few degrees of freedom that are well correlated to modifiable parameters like dimensions or material properties. Therefore, the third challenge is to find robust methods to project the results of detailed numerical simulations—with their enormous number of degrees of freedom—onto spaces spanned by a very small number of appropriately selected dynamical variables. We refer to this as the *nonlinear macromodeling* or *nonlinear model order reduction* problem.

We will try to describe some of the current successes and needed efforts associated with these three challenges.

Fast algorithms for exterior problems

Typical micromachined structures, like the examples given in Figure 1, are geometrically complicated and innately three-dimensional. In addition, the performance of these structures depends critically on exterior forces, like electrostatic pressure and fluid traction, in large or semi-infinite domains. Though it is possible to determine these forces by using finite-element or finite-difference discretization of the associated partial differential equations, such an approach has many difficulties.[5] The most obvious problems are generating a well-behaved exterior mesh (particularly if the structure is deforming), truncating the semi-infinite domain while still ensuring accuracy, and solving an unstructured sparse matrix with an enormous number of unknowns.

For many micromachined device applications, the exterior forces can be described by time-independent linear partial differential equations. For example, quasi-static electromagnetic and fluidic forces are frequently well approximated by solutions to Laplace's equation or the time-independent Stokes equation. For such problems, boundary-element methods applied to surface-integral formulations can be used to avoid the exterior meshing and domain truncation problems. Boundary-element methods typically generate dense matrices, but sparsification-accelerated iterative methods can be used to solve such matrices extremely efficiently.[8] We next describe some of the issues associated with using sparsification-accelerated boundary-element methods.

Integral formulations

Many textbooks contain first-kind integral formulations for determining electrostatic surface charge, given conductor potentials, or for determining traction forces from specified velocities in Stokes flow.[9,10] Such first-kind integral equations have strongly elliptic kernels, and therefore have unique solutions when the integral operator is viewed as a map between appropriately selected Sobolev spaces.[11] Standard discretization schemes for this type of first-kind equation converge,[12] but the condition number of the discretized system grows with mesh refinement, making the system of equations progressively more difficult to solve. (In certain cases, it is possible to eliminate this ill-conditioning using specialized discretization schemes based on Sobolev inner products[13]). In general it is possible to derive second-kind integral formulations for Laplace's equation or Stokes flow, and discretizations of these second-kind formulations generate systems of equations whose condition number remains bounded even as the mesh is progressively refined. However, the most easily derived second-kind formulations generate intermediate quantities which must be locally differentiated or globally integrated to compute the desired forces.[14–16]

For electrostatic force calculation in micromachined devices, where the conducting structures enforce equipotential surfaces, a second-kind integral equation can be *directly* derived for the surface charge. To see how this done, consider the first-kind equation for the charge density σ on a single conducting body at unit potential,

$$\int_S G(x,y)\,\sigma(y)\,dS_y = 1, \quad x \in S \tag{1}$$

where S is the surface of the body, σ is the surface charge density, and $G(x,y) = 1/(4\pi\varepsilon_0 |x-y|)$ is the Green's function for Laplace's equation in the three-space. Exploiting the fact that the electric field in the interior of the conducting body is zero leads to a second-kind formulation for the charge density given by[17]

$$\frac{1}{2}\sigma(x) + \int_S \frac{\partial}{\partial n_x} G(x,y)\,\sigma(y)\,dS_y + \int_S G(x_1,y)\,\sigma(y)\,dS_y = 1 \tag{2}$$

where x_1 is any point in the interior of the body. The above second-kind formulation can be extended to multiple bodies, but not to vanishingly thin conductors.[18]

Generating second-kind formulations whose solution directly yields quantities of interest is not a new subject, but the recent development of accelerated boundary-element methods has revived interest in such formulations for a variety of applications beyond simple electrostatics. These methods can be used to investigate surface-integral formulations for fluid flow that includes time-dependence, for dynamic micromachine analysis, and possibly for nonlinearities.[10]

Discretization

Most boundary-element methods for solving surface-integral equations use a basis-function representation of a surface unknown $\sigma(x)$. An example is

$$\sigma = \sum_i^n \alpha_i \theta_i(x), \tag{3}$$

where the θ_i's are compactly supported but not necessarily orthogonal basis functions.[9,19,20] Consider substituting Equation 3 in a general second-kind formulation

$$\frac{1}{2}\sigma(x) + \int_S G(x,y)\,\sigma(y)\,dS_y = f(x), \quad x \in S, \tag{4}$$

where $G(x,y)$ is a problem-specific Green's function. The result of discretization (denoted D) is a linear system of equations which can be written in matrix form as

$$(M^D + G^D)\alpha = f^D, \tag{5}$$

where $\alpha \in \mathbb{R}^n$ is the vector of basis coefficients. In Equation 5, M^D, $G^D \in \mathbb{R}^{n\times n}$, and $f^D \in \mathbb{R}^n$ depend on whether collocation or Galerkin methods are used.[21,22] For collocation,

$$\begin{aligned} M^D &= I, \\ G^D_{i,j} &= \int G(\tilde{x}_i, x)\,\theta_j(x)\,dx, \text{ and} \\ f^D_i &= f(\tilde{x}_i), \end{aligned} \tag{6}$$

where I is the identity matrix and the $\tilde{x}_i$'s are the collocation points. For Galerkin methods,

$$\begin{aligned} M^D_{i,j} &= \int \theta_i(x)\theta_j(x)\,dx, \\ G^D_{i,j} &= \iint G(x',x)\theta_j(x)\theta_i(x')\,dx\,dx', \text{ and} \\ f^D_i &= \int f(x)\theta_i(x)\,dx. \end{aligned} \tag{7}$$

The integrals in Equations 6 and 7 can be evaluated analytically for certain simple geometries and basis functions.[23,24] For higher-order methods and curved surfaces, the integrals are usually evaluated by semianalytic quadrature methods, because the Green's functions of interest are singular, that is, $\lim_{x\to x'} G(x,x') \to \infty$.[25] It is often useful to note that if the basis functions are orthonormal and a single point quadrature formula is used to evaluate the outer integral in Equation 7, then an approximate Galerkin method is generated that is identical to a collocation method.[18]

In order to develop highly accurate codes that are more Green's-function-independent, there has been recent interest in locally corrected quadrature rules.[26,27] The possibility of success in developing such quadrature rules is particularly exciting because then high-order Nystrom methods could be used to solve Equation 4. In such a locally corrected Nystrom method, one would need only to pick n points on a structure's surface and solve n equations of the form

$$\sigma(x_i) + \sum_{j \neq i} w_j G(x_i, x_j)\sigma(x_j) + \sum_{j \text{ near } i} \tilde{w}_{i,j} G(x_i, x_j)\sigma(x_j) = f(x_i), \quad (8)$$

where the x_i's are the quadrature points, the w_j's are the quadrature weights, and the $\tilde{w}_{i,j}$'s are the local corrections.

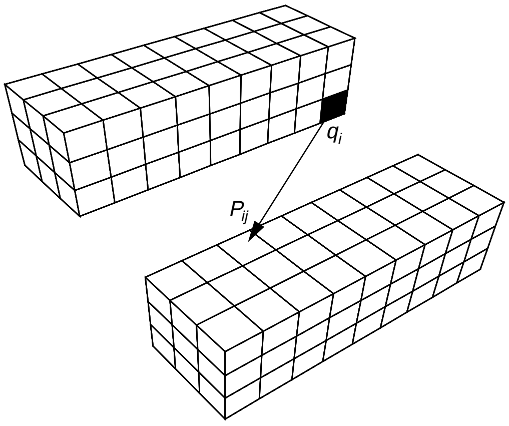

Figure 2. Piecewise-constant collocation discretization of two conductors. Conductor surfaces are discretized into panels that support a constant charge density.

Sparsification

For compact basis functions, M^D in Equation 5 is sparse, but G^D is dense as it represents a discretization of the integral operator. Because the matrices generated by discretized integral operators are typically well-conditioned, Krylov-subspace methods like GMRES[28] are usually used to solve Equation 5. Each iteration of GMRES requires a matrix-vector product, but since G^D is dense, each of these matrix-vector products will require order n^2 operations. It is possible to exploit the structure of G^D so as to compute matrix-vector products with G^D in nearly $O(n)$ operations, provided one does not insist on the computation being exact.

Algorithms that perform fast matrix-vector products are often said to "sparsify" G^D, even though most of these methods only implicitly represent the approximation to G^D. For matrices associated with discretized integral operators, sparsification algorithms rely on a decomposition between nearby and distant interactions. The few nearby interactions are computed directly, and the many distant interactions are computed using approximations.

As a simple example, consider the first-kind integral formulation of an electrostatics problem,

$$\psi(x) = \int_{\text{surfaces}} \sigma(x') \frac{1}{4\pi\varepsilon \|x - x'\|} \, da', \quad (9)$$

where $x \in$ surfaces, and $\psi(x)$ is the known conductor surface potential. The simplest discretization of Equation 9 is to divide the surfaces into flat panels over which the charge density is assumed constant. If the collocation points (the x_i's) are selected at the centroids of each panel, then the discretized system is

$$Pq = \Psi \quad (10)$$

where $q \in \mathbb{R}^n$ is the vector of panel charges, $\Psi \in \mathbb{R}^n$ is the vector of known centroid potentials, and the potential coefficient matrix P is given by

$$P_{i,j} = \int_{\text{panel}_j} \frac{1}{4\pi\varepsilon \|x_i - x'\|} \, da'.$$

This discretization is diagrammed in Figure 2.

If a Krylov-subspace method is used to solve Equation 10, then it will be necessary to compute many matrix-vector products of the form Pq. The matrix P is dense, so computing Pq directly will cost order n^2 operations. Instead, note that computing Pq is equivalent to computing n potentials due to n sources, and consider computing distant interactions by representing clusters of charged panels by a small number of weighted point charges. Specifically, consider dividing a problem domain into a large number of cubes, where each cube contains a few charged panels. The charge in each cube could be represented using a small number of weighted point charges. If the point charges all lie on a uniform grid, then the fast Fourier transform can be used to compute the potential at these grid points due to the grid charges.[29,30] Specifically, Pq may be approximated in order $n \log n$ operations in four steps:

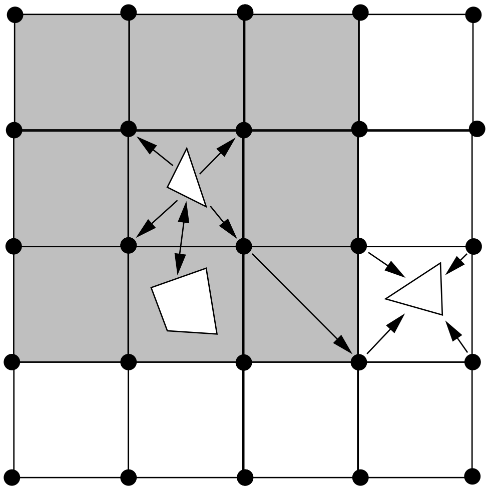

Figure 3. Representation of the four steps of the precorrected-FFT algorithm.

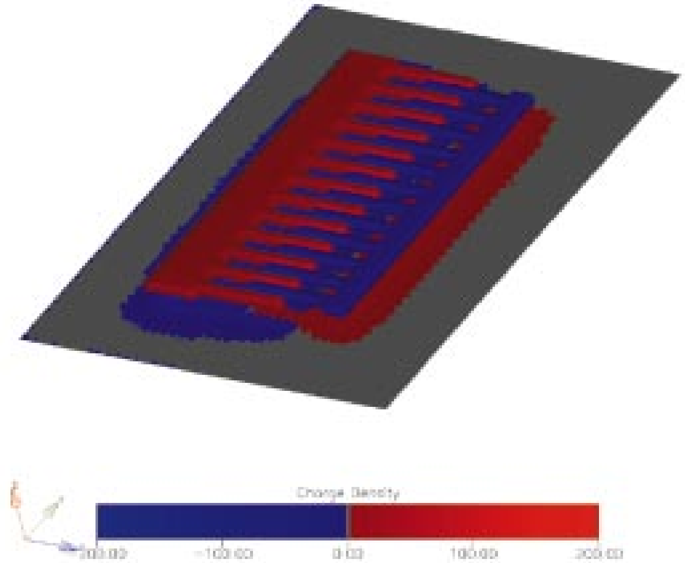

Figure 4. Surface charge distribution on half a comb drive structure. (Courtesy J. Gilbert, Microcosm Technologies)

(1) project the panel charges onto a uniform grid of point charges,

(2) compute the grid potentials due to grid charges using an FFT,

(3) interpolate the grid potentials onto the panels, and

(4) directly compute nearby interactions.

Note that in step 4, it will also be necessary to subtract the inaccurately computed direct interaction associated with the grid projection. The above process is summarized in Figure 3.

There are a wide number of variations on the above basic algorithm, which is quite well known.[31] The FFT is often replaced with a hierarchy of progressively coarser grids on which data are projected.[32] More recent modifications improve computational efficiency by using a hierarchy of progressively finer grids on which data are interpolated.[11,33–37] The advantage of using a grid hierarchy over the FFT is that such approaches can be made sufficiently adaptive to be efficient even for problems with severe geometric inhomogeneities, like a very finely discretized sphere.[32,38,39] The advantage of using the FFT over a hierarchy of grids is that the FFT can be used with oscillatory kernels, like those associated with acoustics or electromagnetic waves.[40]

The other main refinement to the above basic algorithm is the use of higher-order expansions instead of point charges to represent cube distributions.[8,41,42] The advantage of using higher-order expansions is that it is possible to generate efficient schemes that compute the matrix-vector product to machine precision. However, to generate efficient schemes, the interactions between higher-order expansions must be diagonalized, and this can only be accomplished for specific Green's functions.[8,42,43]

The three ideas of using grid hierarchies, higher-order discretization, and higher-order expansions are elegantly combined in wavelet-Galerkin methods, though such methods are problematic for very complicated geometries.[41,44]

Though there are quite a few effective codes that use sparsification, the goal of a Green's-function-independent sparsification procedure that is efficient for arbitrarily inhomogeneous problems remains elusive.

Comb drive example

The combination of an integral formulation, a discretization technique, and a sparsification procedure can lead to very fast algorithms that can analyze extremely complicated structures. In Figure 4 shows an example of simulating the electrostatic forces on half a comb from a micromachined accelerometer.[45] Here the surface charge was computed using the first-kind formulation in Equation 1 and a piecewise-constant collocation scheme with 20,000 unknowns. If direct factorization had been used to solve the generated dense system, it would have required 66.8 CPU hours and 3.2 gigabytes of memory on the DEC Alpha workstation (Model 600 5/333) that was used. If GMRES were used instead of direct factorization, the CPU time would have dropped to 3.8 hours but 3.2 gigabytes of memory would still have been required to store the matrix. Calculating the charge distribution shown in Figure 4 using the precorrected-FFT accelerated method to compute the matrix-vector products, however, required only 6.7 minutes of CPU time and only 220 megabytes of memory.

As the comb drive example demonstrates, accelerated boundary-element methods can be used to solve exterior problems with complicated geometries very rapidly. However, coupling effects, like the mechanical bending induced by electrostatic forces, are extremely important in determining micromachined device behavior. So, these fast techniques for exterior problems must be combined in an efficient way with other simulation algorithms. We discuss this coupled simulation problem in the next section.

Coupled-domain simulation

The performance of most micromachined devices depends critically on the interaction between forces generated by a variety of mechanisms. For example, in microresonator-based gyroscopes, such as in Figure 1 and as described elsewhere,[46] the dynamic performance is due to the coupling between electrostatic, mechanical, and fluidic forces. Simulation of these coupled-domain problems can be accomplished using generic finite-element techniques,[5] but as we described earlier, for Stokes flow or electrostatics there are much more efficient techniques. Therefore, the efficiency of coupled-domain simulation can be substantially improved by using domain-specific solvers, provided the coupling between domains can be handled effectively. We will describe three basic approaches to coupled-domain simulation, and give computational comparisons between relaxation, multilevel Newton, and full-Newton methods for 3D electromechanical analysis.

Domain solvers

Consider the problem of coupled simulation between m domains, where in each domain a discretization with n_i degrees of freedom has been introduced. Let $x_i \in \mathbb{R}^{n_i}$ denote the vector of unknowns associated with domain i, where x_i is computed by solving a possibly nonlinear residual equation

$$R_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_m) = 0. \qquad (11)$$

Here $x_j, j \neq i$, are the unknowns associated with other domains, and are usually treated as inputs when solving the R_i residual equation for x_i. Most domain-specific programs solve their associated residual equations using variants of Newton's method, as in

$$\frac{\partial R_i}{\partial x_i}(x_i^{k+1} - x_i^k) \\ = -R_i(x_1, \ldots, x_{i-1}, x_i^k, x_{i+1}, \ldots, x_m), \qquad (12)$$

where k is the Newton iteration index and

$$\frac{\partial R_i}{\partial x_i} \in \mathbb{R}^{n_i \times n_i}$$

is the Jacobian with respect to x_i. Of course, if the residual equation for domain i is linear, only one Newton iteration is required.

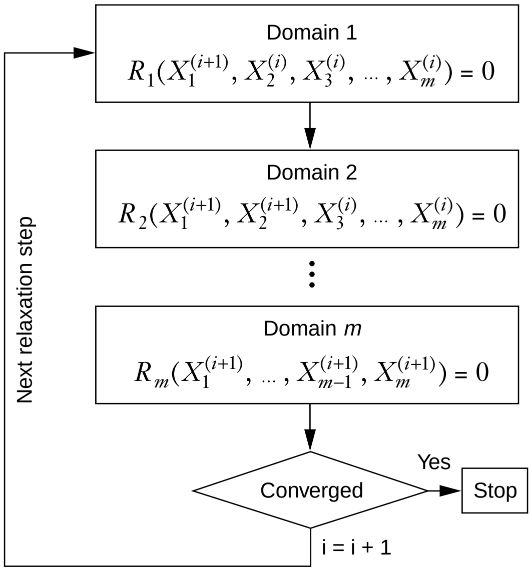
Domain 1
$R_1(X_1^{(i+1)}, X_2^{(i)}, X_3^{(i)}, \ldots, X_m^{(i)}) = 0$
Domain 2
$R_2(X_1^{(i+1)}, X_2^{(i+1)}, X_3^{(i)}, \ldots, X_m^{(i)}) = 0$
Domain m
$R_m(X_1^{(i+1)}, \ldots, X_{m-1}^{(i+1)}, X_m^{(i+1)}) = 0$
Converged
Yes
Stop
i = i + 1
Next relaxation step

Figure 5. A simple relaxation scheme for coupled-domain simulation.

Solution alternatives

One approach to solving the coupled system

$$\begin{aligned} &R_1(x_1, \ldots, x_m) = 0 \\ &R_2(x_1, \ldots, x_m) = 0 \\ &\qquad \vdots \\ &R_m(x_1, \ldots, x_m) = 0 \end{aligned} \qquad (13)$$

is to use the simple nonlinear Gauss–Seidel relaxation scheme diagrammed in Figure 5. The relaxation algorithm does not always converge, particularly when the different domains are tightly coupled.[47] Sufficient conditions for nonlinear relaxation convergence have been exten-

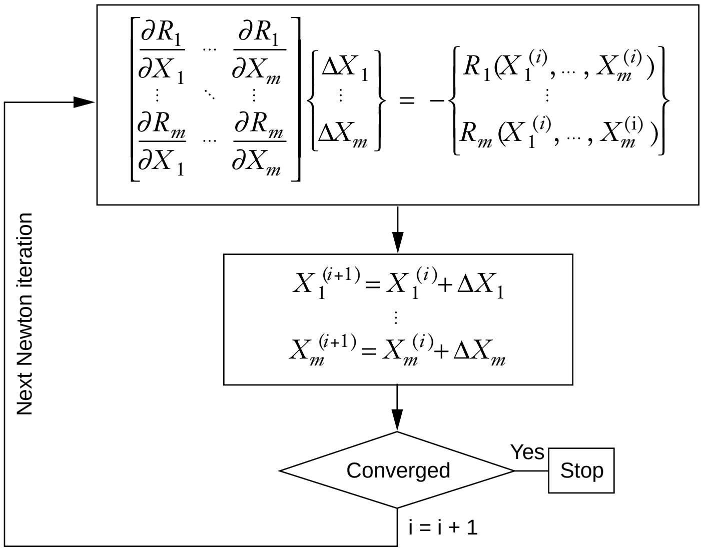
$\begin{bmatrix} \frac{\partial R_1}{\partial X_1} & \cdots & \frac{\partial R_1}{\partial X_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial R_m}{\partial X_1} & \cdots & \frac{\partial R_m}{\partial X_m} \end{bmatrix} \begin{Bmatrix} \Delta X_1 \\ \vdots \\ \Delta X_m \end{Bmatrix} = -\begin{Bmatrix} R_1(X_1^{(i)}, \dots, X_m^{(i)}) \\ \vdots \\ R_m(X_1^{(i)}, \dots, X_m^{(i)}) \end{Bmatrix}$
$X_1^{(i+1)} = X_1^{(i)} + \Delta X_1$
$\vdots$
$X_m^{(i+1)} = X_m^{(i)} + \Delta X_m$
Converged
Yes
Stop
i = i + 1
Next Newton iteration

Figure 6. A full-Newton scheme for coupled-domain simulation.

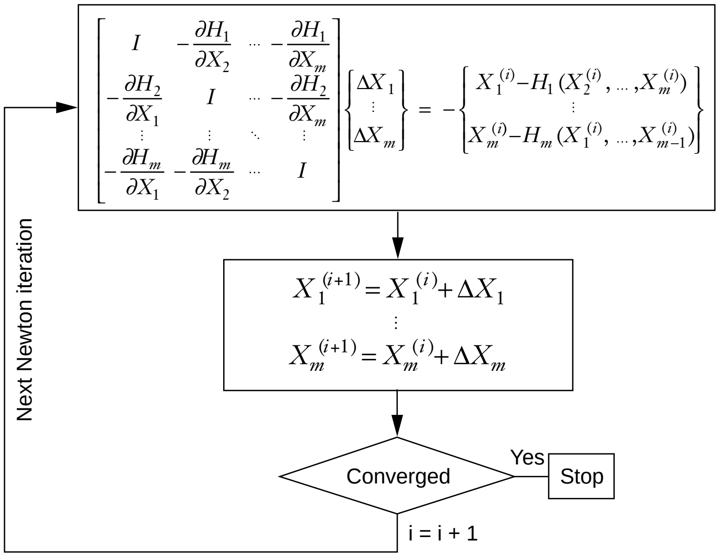
$\begin{bmatrix} I & -\frac{\partial H_1}{\partial X_2} & \cdots & -\frac{\partial H_1}{\partial X_m} \\ -\frac{\partial H_2}{\partial X_1} & I & \cdots & -\frac{\partial H_2}{\partial X_m} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\partial H_m}{\partial X_1} & -\frac{\partial H_m}{\partial X_2} & \cdots & I \end{bmatrix} \begin{Bmatrix} \Delta X_1 \\ \vdots \\ \Delta X_m \end{Bmatrix} = -\begin{Bmatrix} X_1^{(i)} - H_1(X_2^{(i)}, \dots, X_m^{(i)}) \\ \vdots \\ X_m^{(i)} - H_m(X_1^{(i)}, \dots, X_{m-1}^{(i)}) \end{Bmatrix}$
$X_1^{(i+1)} = X_1^{(i)} + \Delta X_1$
$\vdots$
$X_m^{(i+1)} = X_m^{(i)} + \Delta X_m$
Converged
Yes
Stop
i = i + 1
Next Newton iteration

Figure 7. A multilevel Newton scheme for coupled-domain simulation.

sively analyzed.[48]

A more robust approach than nonlinear relaxation is to use a full-Newton method, as diagrammed in Figure 6, possibly combined with a continuation or a homotopy scheme.[49,50] The difficulty with the full-Newton approach is that the off-diagonal, or coupling, derivatives $\partial R_i / \partial x_j$, $i \neq j$, may not be available explicitly. If an iterative method like GMRES is used to solve the linear system, then only matrix-vector products are required. Therefore, the off-diagonal derivatives need not be explicitly computed, and can be approximated by finite differences. For example, if Δu_j is a part of the vector generated by GMRES, then

$$\frac{\partial R_i}{\partial x_j} \Delta u_j \approx R_i(x_j + \Delta u_j) - R_i(x_j), \qquad (14)$$

where we have assumed that the Δu_j vector has been scaled to be small in magnitude.[51]

If Krylov-subspace methods like GMRES are used to solve the system in Figure 6, then the diagonal blocks should be explicitly factored and used as a preconditioner.[52]

Programs that perform domain-specific analysis are *not* usually organized so that one can efficiently determine the residual equation differences required in Expression 14. In addition, the explicitly computed parts of $\partial R_i / \partial x_i$, which are needed for preconditioning the full-Newton iteration equations, are often unavailable. So, to create a robust full-Newton method, the individual solvers must be modified somewhat, and this means the coupled method is not really a "black box" approach in which domain-specific solvers can be easily interchanged. Consider instead that a program which solves a domain-specific residual equation, as in Equation 11, can be thought of as producing x_i given $x_j, j \neq i$. We denote the input/output description as

$$x_i = H_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_m). \qquad (15)$$

With Equation 15 in mind, consider a third approach, the multilevel Newton method given in Figure 7, which can be used to determine the solution to the coupled system. We refer to this as a multilevel Newton method because application of the H_i operator implies solving residual equation R_i, and this is typically done with an inner Newton's method. Note that in the multilevel Newton method, the Jacobian block diagonals are already identity matrices and need not be preconditioned. Also, application of

$$\left(\frac{\partial H_i}{\partial x_i}\right)^{-1} \left(\frac{\partial H_i}{\partial x_j}\right)$$

can be performed using finite differences and multiple calls to the domain-specific solver *without modification*.

An electromechanical example

Again we will use a MEMS comb drive as an example. The device consists of a deformable comb structure, a drive structure, and a ground plane. As shown in Figure 8, the F-shaped finger structure is the comb, the E-shaped finger structure is the drive, and the rectangular-shaped structure is the ground plane. When a positive potential is applied on the drive structure, and zero potential on the comb and the ground plane, the comb structure deforms out of plane. The deformation of the comb structure for an applied bias of 85 volts is shown in Figure 9. Note that only the comb structure deforms and the drive and the ground plane do not move.

This coupled electromechanical problem can be simulated by self-consistently solving for the electrostatic surface charges given the structure deformation, and solving for the elastostatic deformation given the electrostatic pressure. For this example, the exterior electrostatic analysis was performed using the accelerated boundary-element method discussed earlier, and the interior elastostatic analysis was performed with a standard parabolic-brick finite-element method.[52] The comb was discretized into 172 parabolic elements, the drive into 144 linear bricks, and the ground plane into 2,688 four-node elements.

A comparison of the relaxation, multilevel Newton, and full-Newton algorithms for the comb example is summarized in Table 1. At low voltages the deflection of the comb is small, the coupling between the electrical and mechanical systems is weak, and the relaxation algorithm works very well. At low voltages, both the multilevel Newton and full-Newton algorithms take fewer iterations than the relaxation algorithm but the simulation time for the Newton algorithms is a little longer. For higher voltages, the Newton algorithms converge much faster than the relaxation algorithm. For a bias of 80 volts, the multilevel Newton algorithm is about 7.7 times as fast and the full-Newton algorithm is about 5 times as fast as relaxation. The convergence of the relaxation, multilevel Newton, and full-Newton algorithms at 80 V bias is shown in Figure 10. For an application of 85 V on the drive, the relaxation algorithm fails to converge, while the multilevel Newton and full-Newton algorithms converge very rapidly and take 3 and 10 iterations, respectively. This is illustrated in Figure 11.

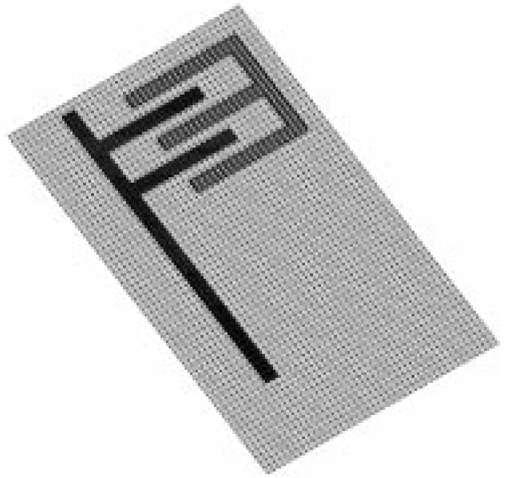
Figure 8. Comb drive example.

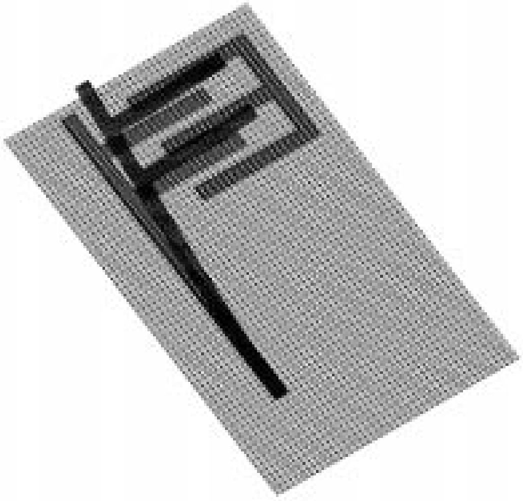
Figure 9. Deformation of the comb (not to scale) for an applied bias of 85 volts.

Table 1. Comparison of three algorithms in simulating a MEMS comb drive. (An asterisk indicates that the algorithm fails to converge for the voltage bias applied.)

	Number of iterations			CPU time (seconds)		
Voltage applied	Relaxation	Multilevel Newton	Full Newton	Relaxation	Multilevel Newton	Full Newton
25	7	3	6	3,595	5,802	5,590
50	16	4	8	9,138	10,195	11,834
75	70	4	10	42,160	12,053	18,591
80	142	3	9	81,827	10,660	16,670
85	*	3	10	*	10,768	18,491

Fast domain-specific solvers and efficient approaches for coupled-domain simulation can

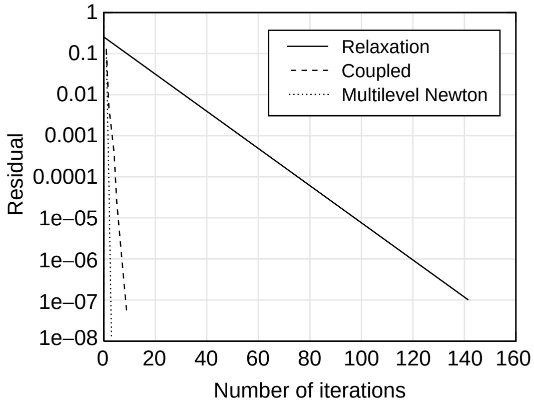

Figure 10. Comparison of convergence of relaxation, multilevel Newton, and full-Newton (shown as "coupled") algorithms for the MEMS comb example at an applied bias of 80 V.

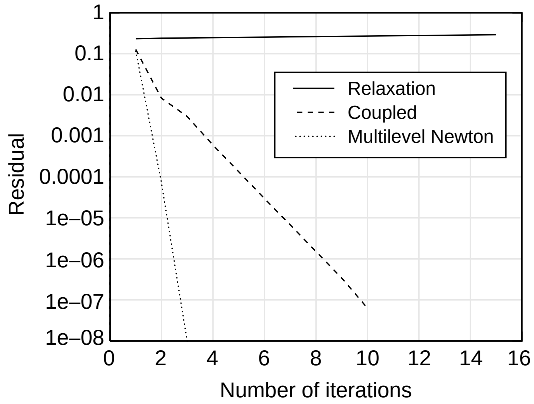

Figure 11. Convergence of the same three algorithms on the comb example at an applied bias of 85 V. The relaxation algorithm does not converge; the others converge quickly.

make it possible to perform realistic analyses of single micromachined devices in a few hours on a scientific workstation. However, such simulation times are still far too long to be used for design exploration or as part of a system-level simulation. Instead, these tools can be used to help generate macromodels.

Nonlinear macromodeling

The modeling level described as *macromodeling* plays several roles in the overall design scheme. First, consider that a designer might want to use a macromodel to explore a design space, that is, to predict easily how behavior will change as dimensions or material properties are changed. This argues strongly for an *analytical* model that has a minimal number of degrees of freedom—essentially a "lumped" model. Second, the macromodel will represent the MEMS device in a system-level simulator. Therefore, the macromodel must be *dynamic*, and must be simple enough to permit hundreds or thousands of dynamical simulations under a variety of excitations in reasonable time. Third, because MEMS devices are usually transducers involving multiple energy domains, the macromodel should correctly account both for *energy conservation* (a quasi-static property), and *energy dissipation* (a dynamic property). Finally, and most important from our point of view, the macromodel should agree with the results of more detailed numerical simulation over some design space of interest, and should be based on approximations that have been compared carefully with experiment on suitably designed test structures.

The only models at present that meet all these requirements are hand-built. Our goal is to find those steps in the macromodel development process that can be effectively automated, and to seek algorithms which speed up macromodel formulation and use. We will first describe what has proved to be a very successful approach to quasi-static macromodeling, and then present some suggestions on how to approach dynamic macromodeling.

Quasi-static macromodeling

We have had very good success in creating quasi-static macromodels starting from simplified analytical formulations.[53,54] The procedure is roughly as follows:

(1) Select an idealized structure that is close to the desired model.

(2) Model the idealized problem analytically, either by solving the governing differential equation, or by approximating the solution with Rayleigh-Ritz energy minimization methods.

(3) Identify a set of nondimensionalized numerical constants that can be varied within the analytical form of the solution.

(4) Perform meshed numerical simulations of the desired structure over the design space of interest, and adjust the nondimensionalized numerical quantities in the macromodel for agreement with the numerical simulations.

Membrane load-deflection. As an example, consider the pressure-deflection behavior of a thin elastic membrane, suspended on a rigid frame (Figure 12). If a differential pressure is applied to this structure, it deforms out of plane. The center-deflection d is related to the applied pressure P through an equation that depends on the *geometry* of the membrane and the *material constants* (Young's modulus E, Poisson ratio ν, and residual stress σ). If the membrane is modeled as a pure membrane (with no bending stiffness), closed-form analytical solutions can be found for circular geometries, and both power-series and Rayleigh-Ritz approximants can be found for square and rectangular geometries. In this case, the Rayleigh-Ritz form is quite helpful, because it yields closed-form expressions with explicit dimensions and material constants. The resulting form of the pressure–deflection relation, including the added nondimensionalized adjustable parameters, is expressed as

$$P = \frac{C_1 t}{a^2}\sigma d + \frac{C_2 f(\nu)\, t}{a^4}\,\frac{E}{1-\nu}\, d^3 \qquad (16)$$

where a is the radius/half-edge length of the membrane and t is the thickness. The dimensionless constants C_1 and C_2 and the dimensionless function $f(\nu)$ are determined from fitting Equation 16 to the results of extensive finite-element simulations over a range of dimensions (length and thickness) and material constants.[53] Note that a factor $(1 - \nu)$ appears in the denominator, with the rest of the ν-dependence captured by $f(\nu)$, which is a slowly varying function, changing only 6 percent for a variation of ν over the physically interesting range 0.3–0.5. The numerical constants obtained for variously shaped membranes are provided in Table 2, in which "rectangular" refers to rectangular shapes with a width-to-length ratio of greater than 8:1.

The load-deflection behavior captured in Equation 16 using the constants in Table 2 has been very successfully applied to a wide variety of microfabricated structures, both for prediction of performance and for the extraction of material constants from test devices.

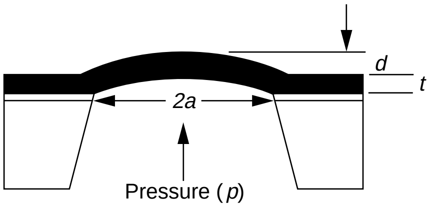

Figure 12. Suspended membrane illustrating load-deflection behavior under a uniform pressure load.

Table 2. Numerical results for the membrane model.

Membrane shape	C_1	C_2	$f(\nu)$
Circular	4.00	2.67	$0.957 - 0.208\nu$
Square	3.41	1.37	$1.446 - 0.427\nu$
Rectangular	2.00	1.33	$(1 + \nu)^{-1}$

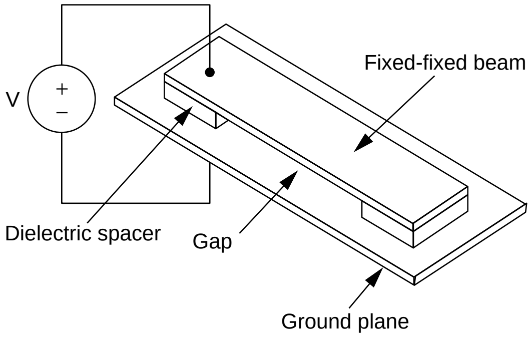

Figure 13. Electrostatically actuated fixed-fixed beam. When $V = V_{PI}$ the beam collapses.

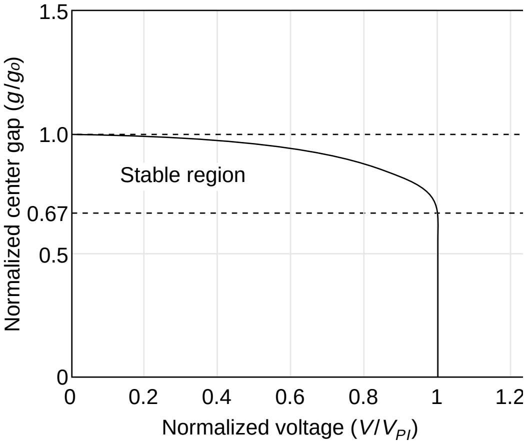

Figure 14. Typical normalized gap versus voltage for a structure like the fixed-fixed beam of Figure 13, showing the sharp collapse.

Electrostatic pull-in of beams. This same method can be used to analyze the instability point of an electrostatically actuated elastic structure. Figure 13 illustrates a conducting beam of thickness t, length L, and width W, clamped at both ends by dielectric supports, and suspended over a ground plane by a gap g_o. When a voltage is applied between the beam and the ground plane, the charges on the beam and the ground plane produce an attractive force between them, which causes the beam to bend toward the ground plane. Figure 14 shows the normalized dependence of the deflection of the center of the beam on applied voltage. At a critical voltage, called the *pull-in voltage* or V_{PI}, the linear elastic restoring force is overwhelmed by the nonlinear attractive force, and the beam collapses. This pull-in event is sharp, and readily observed experimentally. It depends strongly both on the beam's geometry and on its material constants. This has made possible the development of a method for material-property measurement that we call *M-Test*.[54]

The models for the dependence of pull-in voltage on geometry and material properties within M-Test were derived initially from analytical beam theory, with three adjustable constants added for extensive fitting to the results of fully meshed coupled-energy domain simulations, using the procedure described earlier under "Quasi-Static Macromodeling." The result can be summarized as follows:

$$V_{PI} = f_{PI}(S,B,L) \tag{17}$$

where $f_{PI}(S,B,L)$ is a lengthy but closed-form function of the beam length L and two composite parameters, the *stress parameter* S and the *bending parameter* B, given by

$$S = \sigma t g_o^3 \quad \text{and} \tag{18}$$

$$B = \hat{E}t^3 g_o^3, \tag{19}$$

where σ is the residual axial stress in the beam and $\hat{E}$ is the effective stiffness of the structure. (The reader is referred elsewhere for details.[54])

By measuring the variation of V_{PI} with beam length, and fitting the results to the functional form of f_{PI}, numerical values of B and S can be obtained, which when coupled with geometric data on t and g_o permit determination of σ and $\hat{E}$.

Dynamical macromodels

Dynamical macromodels are much more challenging than quasi-static ones, particularly when the design space involves large motions and nonlinear forces. Explicit dynamical formulations for meshed structures are perfectly possible, at least in principle, but have many drawbacks. First, they are typically relatively slow (compared to the better alternatives described below). Second, being fully numerical, explicit dynamical simulations of meshed structures make it difficult for the designer to probe sensitivities to variations in geometry or material constants without simply performing multiple analyses and building up a large database. Third, fully meshed models are too computationally expensive to insert in system-level simulators.

For all these reasons, we have been seeking methods which permit projection of the results of fully-meshed analyses onto physically meaningful reduced variable sets, preferably already containing the appropriate algebraic dependences on structural dimensions and material constants.[55]

One obvious choice, suitable for cases where the device can be decomposed into a combination of rigid bodies, ideal springs, and ideal variable capacitors, is to represent the rigid bodies as lumped elements and the springs as lumped linear elastic elements, and to use 3D electrostatic simulation to build a macromodel for the very nonlinear relationship between electrostatic force and device geometry. In particular, fast electrostatic analysis algorithms, like those described above,[37] can be used to determine the system's capacitance matrix as a function of the displacements and rotations of the rigid bodies relative to one another.[56] Then, a total energy function is built by fitting the various components of the capacitance matrix to suitably formulated analytical expressions with added nondimensionalized constants (as in the previous section), capturing the electrostatic stored energy, elastic stored energy, and kinetic energy, from which either Lagrangian or Hamiltonian mechanics quickly yields relatively simply dynamical equations.

Many structures, however, such as the membrane and beam examples, do not lend themselves to treatment as lumped elements. In that case, we seek to perform some kind of modal expansion. We will describe one example of this.

Nonlinear dynamics using linear modes. The basic approach here is to formulate the dynamical behavior of a device in terms of a finite set of orthonormal spatial basis functions $\{s_n(x)\}$, each with a time-dependent coefficient $q_n(t)$.[57] To derive this basis from an initially meshed structure, we take advantage of the fact that FEM packages typically are very fast at solving for the small-amplitude (linear) modes of a structure. That the structure of interest actually undergoes *large*-amplitude motions and experiences nonlinear forces is not a problem—we are simply using the mode shapes as a convenient *geometric* basis set.

Formally (and for notational simplicity, in one spatial dimension), if $y \in \mathbb{R}^n$ is the vector containing the nodal displacements of a fully meshed structure, where n is the number of nodes, the undamped dynamical behavior of the structure is written as

$$M\ddot{y} + Ky = f_e(y,t), \tag{20}$$

where M is the global inertial matrix, K is the global stiffness matrix, $f_e(y, t)$ is the nonlinear external force, and $\ddot{y}$ is the second derivative of the displacements with respect to the time t (that is, the acceleration).

The *modal matrix* S is the matrix whose columns are the orthonormal mode-shape vectors for this system. It has the useful properties that

♦ $S^{\mathrm{T}}MS = M_G$ (the generalized inertia matrix)
♦ $S^{\mathrm{T}}KS = K_G$ (the generalized stiffness matrix)

where M_G and K_G are diagonal.

If we expand the displacements y in terms of the modes, we obtain

$$y = Sq \tag{21}$$

where components of the vector q are the (time-dependent) mode amplitudes of the n modes. Inserting Equation 21 into Equation 20 and multiplying by S^{T} yields

$$\ddot{q} + M_G^{-1}K_G q = S^{\mathrm{T}}M^{-1}f_e(Sq, t). \tag{22}$$

The advantage of this formulation is that the matrices on the left-hand side are now diagonal and time-independent, so the only coupling of the various q_i's is through the nonlinear force term f_e. Finally, since in practice only a few modes are actually needed to describe the deformation of real structures, the $n \times n$ set of equations represented by Equation 22 is truncated to a much smaller number, m. When applying this to a simple beam example with various placements of electrodes under the beam so as to be certain to excite higher modes, we found that adequate accuracy could be obtained using only four modes. Once this macromodel is constructed, it can be used to simulate structure performance in a wide range of scenarios. And of course, the macromodel is orders of magnitude less expensive to evaluate than performing explicit dynamics on even relatively coarsely meshed structures.[57]

Besides the speedup, which is important, linear modes have the advantage of being sensibly connected to physical features such as dimensions and material constants. Thus, even if one lacks an explicit formula for the exact eigenfrequency for a given mode, one can usually reason analytically how that frequency will depend on a structural dimension or on the value of an elastic modulus. Therefore, these linear modes provide a very useful set of basis functions for dynamical macromodels.

If one seeks to include dissipative properties, such as fluid damping, the problem becomes much more difficult. Quasi-static representations of the fluid cannot typically be used. In this case one is faced with two choices: either include the damping force as an additional term on the right-hand side of Equation 20, or find a new set of geometric basis functions that include the space external to the structure where the fluid damping is occurring. In addition, one would like to improve on the formulation of the right-hand side of Equation 22 because, as stated, a conversion from q-space to y-space is required at each time point in order to evaluate f_e, and then the resulting f_e must be reconverted back into q-space. Our group is working on all three of these issues, and we hope to report positive results in the near future.

In spite of the large challenges ahead, the future of effective micromachine design is very bright. Forecasts predict that the market for micromachine devices and systems will grow exponentially in the coming decade. As total product volume grows, and as the time-to-market in a competitive industry becomes more crucial, there will be an increasing need for effective design tools that permit meaningful "what if" experimentation before committing to costly and slow microfabrication. Successful research in the areas of faster algorithms, coupled-domain simulation, and nonlinear macromodeling, when carried out in close collaboration with process and device development, will accelerate the conversion of MEMS product ideas into real products. ♦

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency under contracts N00014-91-J-1698, J-FBI-88-067, and J-FBI-92-196; by the Office of Naval Research under contract N00014-90-J-1085; by the National Science Foundation under contract MIP-8858764 A02; and by grants from Digital Equipment Corp., Hewlett-Packard, Omron, and IBM.

References

1. J. Bryzek, K. Petersen, and W. McCulley, "Micromachines on the March," *IEEE Spectrum,* Vol. 31, No. 5, May 1994, pp. 20–31.
2. S.B. Crary, O. Juma, and Y. Zhang, "Software Tools for Designers of Sensor and Actuator CAE Systems," *Proc. Transducers '91: Int'l Conf. on Solid-State Sensors and Actuators* [San Francisco], IEEE, Piscataway, N.J., 1991, pp. 498–501.
3. S.D. Senturia et al., "A Computer-Aided Design System for Microelectromechanical Systems (MEMCAD)," *J. Microelectromechanical Systems,* Vol. 1, Mar. 1992, pp. 3–13.
4. J.R. Gilbert et al., "Implementation of a MEMCAD System for Electrostatic and Mechanical Analysis of

Complex Structures from Mask Description," *Proc. Sixth IEEE Workshop on Micro Electro Mechanical Systems* [Ft. Lauderdale, Fla., 1993], IEEE, Piscataway, N.J., 1993, pp. 207–212.

5. J.G. Korvink, J. Funk, and H. Baltes, "IMEMS Modelling," *Sensors and Materials*, Vol. 6, 1994, pp. 235–239.
6. Microcosm Technologies, http://www.memcad.com/.
7. IntelliSense, http://www.intellis.com/.
8. V. Rokhlin, "Rapid Solution of Integral Equations of Classical Potential Theory," *J. Computational Physics*, Vol. 60, No. 2, 1985, pp. 187–207.
9. R.F. Harrington, *Field Computation by Moment Methods*, Macmillan, New York, 1968.
10. C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flows*, Cambridge Univ. Press, New York, 1992.
11. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
12. I. Sloan, "Error Analysis of Boundary Integral Methods," *Acta Numerica*, Vol. 1, 1992, pp. 287–339.
13. G.C. Hsiao and R.E. Kleinman, *Error Control in Numerical Solution of Boundary Integral Equations,* Research Report, Univ. of Delaware, 1995.
14. S.G. Mikhlin, *Integral Equations*, Pergamon Press, New York, 1957.
15. S.G. Mikhlin, *Linear Integral Equations*, Hindustan Publishing Corp., Delhi, India, 1960.
16. A. Greenbaum, L. Greengard, and G.B. McFadden, "Laplace's Equation and the Dirichlet-Neumann Map in Multiply Connected Domains," *J. Computational Physics*, Vol. 105, 1993, pp. 267–278.
17. R. Kress, *Linear Integral Equations*, Applied Mathematical Sciences Vol. 82, Springer-Verlag, Berlin, 1989.
18. J. Tausch and J. White, "Preconditioning First and Second Kind Integral Formulations of the Capacitance Problem," *Proc. 1996 Copper Mountain Conf. Iterative Methods*, 1996.
19. G.P. Muldowney and J.J.L. Higdon, "A Spectral Boundary Element Approach to Three-Dimensional Stokes Flow," *J. Fluid Mechanics*, Vol. 298, 1995, pp. 167–192.
20. H.D. Manier, *A Three Dimensional Higher Order Panel Method Based on B-Splines,* doctoral dissertation, MIT, Cambridge, Mass., Dept. of Ocean Engineering, 1995.
21. C.A. Brebbia, J.C.F. Telles, and L.C. Wrobel, *Boundary Element Techniques*, Springer-Verlag, Berlin, 1984.
22. K.E. Atkinson, "A Survey of Boundary Integral Equations for the Numerical Solution of Laplace's Equation in Three Dimensions," in *Numerical Solution of Integral Equations*, M. Goldberg, ed., Plenum Press, New York, 1990, pp. 1–34.
23. J.N. Newman, "Distribution of Sources and Normal Dipoles over a Quadrilateral Panel," *J. Engineering Math.*, Vol. 20, 1986, pp. 113–126.
24. A.E. Ruehli and P.A. Brennan, "Efficient Capacitance Calculations for Three-Dimensional Multiconductor Systems," *IEEE Trans. Microwave Theory and Techniques,* Vol. 21, Feb. 1973, pp. 76–82.
25. M.H. Lean and A. Wexler, "Accurate Numerical Integration of Singular Boundary Element Kernels over Boundaries with Curvature," *Int'l J. Numerical Methods in Engineering*, Vol. 21, 1985, pp. 211–228.
26. J. Strain, "Locally Corrected Multidimensional Quadrature Rules for Singular Functions," to be published in *SIAM J. Scientific Computing*.
27. S. Kapur and V. Rokhlin, *High-Order Corrected Trapezoidal Rules for Singular Functions,* Tech. Report YALE/DCS/RR-1042, Yale Univ., New Haven, Conn., 1994.
28. Y. Saad and M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Scientific and Statistical Computing,* Vol. 7, No. 3, 1986, pp. 105–126.
29. J.R. Phillips and J.K. White, "Efficient Capacitance Extraction of 3D Structures using Generalized Pre-Corrected FFT Methods," in *Proc. IEEE 3rd Topical Meeting on Electrical Performance of Electronic Packaging,* IEEE, New York, 1994, pp. 253–256.
30. D.T. Borup and O.P. Gandhi, "Calculation of High-Resolution SAR Distributions in Biological Bodies using the FFT Algorithm and Conjugate Gradient Method," *IEEE Trans. Microwave Theory and Techniques*, Vol. 33, 1985, pp. 417–419.
31. R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, New York, 1988.
32. J. Barnes and P. Hut, "A Hierarchical $O(N \log N)$ Force-Calculation Algorithm," *Nature*, Vol. 324, 1986, pp. 446–449.
33. A. Brandt, "Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernels," *Computer Physics Comm.*, Vol. 65, 1991, pp. 24–38.
34. A. Brandt and A.A. Lubrecht, "Multilevel Matrix Multiplication and Fast Solution of Integral Equations," *J. Computational Physics*, Vol. 90, 1990, pp. 348–370.
35. L. Greengard and V. Rokhlin, "A Fast Algorithm for Particle Simulations," *J. Computational Physics*, Vol. 73, 1987, pp. 325–348.
36. C.R. Anderson, "A Method of Local Corrections for Computing the Velocity Field Due to a Distribution of Vortex Blobs," *J. Computational Physics*, Vol. 62, 1986, pp. 111–123.
37. K. Nabors and J. White, "Fastcap: A Multipole Accelerated 3-D Capacitance Extraction Program," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* Vol. 10, Nov. 1991, pp. 1447–1459.
38. J. Carrier, L. Greengard, and V. Rokhlin, "A Fast Adaptive Multipole Algorithm for Particle Simulations," *SIAM J. Scientific and Statistical Computing,* Vol. 9, July 1988, pp. 669–686.
39. K. Nabors et al., "Preconditioned, Adaptive, Multipole-Accelerated Iterative Methods for Three-Dimensional First-Kind Integral Equations of Potential Theory," *SIAM J. Scientific and Statistical Computing,* Vol. 15, No. 3, 1994, pp. 713–735.
40. J.R. Phillips and J.K. White, "Precorrected-FFT Methods for Electromagnetic Analysis of Complex 3-D Interconnect and Packages," *Proc. PIERS '95, Progress In Electromagnetic Research Symp.* [Seattle, Wa.], 1995.
41. G. Beylkin, R. Coifman, and V. Rokhlin, "Fast Wavelet Transforms and Numerical Algorithms," *Comm. Pure and Applied Math.,* Vol. 44, 1991, pp. 141–183.
42. L. Greengard, *The Rapid Evaluation of Potential Fields in Par-*

ticle Systems, MIT Press, Cambridge, Mass., 1988.

43. L. Berman, "Grid-Multipole Calculations," *SIAM J. Scientific Computing,* Vol. 16, Sept. 1995, pp. 1082–1091.
44. T.V. Petersdorff and C. Schwab, "Boundary Element Methods with Wavelets and Mesh Refinement," *Proc. ICIAM '95* (published as suppl. nos. to *Zeitschrift für Angewandte Mathematik und Mechanik [ZAMM],* Vol. 76), Akademie Verlag, Berlin, 1996.
45. S.J. Sherman et al., "A Low Cost Monolithic Accelerometer," *1992 Symp. on VLSI Circuits, Digest of Technical Papers*, IEEE, Piscataway, N.J., 1992, pp. 34–35.
46. M.W. Putty and K. Najafi, "A Micromachined Vibrating Ring Gyroscope," *Proc. Sixth IEEE Solid-State Sensor and Actuator Workshop* [Hilton Head Is., S.C.], IEEE, Piscataway, N.J., 1994, pp. 213-220.
47. H. Yie, X. Cai, and J. White, "Convergence Properties of Relaxation Versus the Surface-Newton Generalized-Conjugate Residual Algorithm for Self-Consistent Electromechanical Analysis of 3-D Micro-Electro-Mechanical Structures," *Proc. Numerical Process and Device Modeling (NUPAD) V* [Honolulu, Hi.], 1994, pp. 137–140.
48. J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
49. E.L. Allgower and K. Georg, *Numerical Continuation Methods*, Springer-Verlag, New York, 1990.
50. N.R. Aluru and J. White, "Direct-Newton Finite-Element/Boundary-Element Technique for Micro-Electro-Mechanical Analysis," *Tech. Digest, Solid-State Sensor and Actuator Workshop* [Hilton Head Is., S.C.], Transducers Research Foundation, Cleveland Heights, Ohio, 1996, pp. 54–57.
51. P.N. Brown and Y. Saad, "Hybrid Krylov Methods for Nonlinear Systems of Equations," *SIAM J. Scientific and Statistical Computing,* Vol. 11, 1990, pp. 450–481.
52. N.R. Aluru and J. White, "A Coupled Numerical Technique for Self-Consistent Analysis of Micro-Electro-Mechanical-Systems," in *Microelectromechanical Systems (MEMS)*, ASME Dynamic Systems and Control (DSC) series Vol. 59, part of *Proc. 1996 ASME Int'l Mechanical Engineering Congress and Exposition*, Am. Soc. of Mechanical Engineers, New York, 1996, pp. 275–280.
53. J.Y. Pan et al., "Verification of FEM Analysis of Load-Deflection Methods for Measuring Mechanical Properties of Thin Films," *Tech Digest, 1990 IEEE Solid-State Sensor and Actuator Workshop* [Hilton Head Is., S.C.], IEEE, Piscataway, N.J., 1990, pp. 70–74.
54. P.M. Osterberg and S.D. Senturia, "M-TEST: A Test Chip for MEMS Material Property Measurement using Electrostatically Actuated Test Structures," submitted to *J. Microelectromechanical Systems*.
55. S.D. Senturia, "CAD for Microelectromechanical Systems," *Proc. Transducers '95: 8th Int'l Conf. on Solid-State Sensors and Actuators*, Elsevier, Netherlands, 1996, Vol. 2, pp. 5–8.
56. H. Yie et al., "A Computationally Practical Approach to Simulating Complex Surface-Micromachined Structures with Fabrication Induced Non-Idealities," *Proc. IEEE Micro Electro Mechanical Systems (MEMS),* 1995, pp. 128–132.
57. G.K. Ananthasuresh, R.K. Gupta, and S.D. Senturia, "An Approach to Macromodeling of MEMS for Nonlinear Dynamic Simulation," in *Microelectromechanical Systems (MEMS)*, ASME Dynamic Systems & Control (DSC) ser. Vol. 59, part of *Proc. 1996 ASME Int'l Mechanical Engineering Congress and Exposition*, Am. Soc. of Mechanical Engineers, New York, 1996.

Stephen D. Senturia is the Barton L. Weller Professor of Electrical Engineering in the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, and is also associated with the Microsystems Technology Laboratories there. He is a founding editor of the IEEE/ASME *Journal of Microelectromechanical Systems* and is associate editor for solid-state sensors of *IEEE Transactions on Electron Devices*. He cofounded the IEEE Solid-State Sensor and Actuator Workshop series and has served on its organizing and program committees many times, as well as serving as program chair in 1990 and general chair in 1992. He has also been an organizer of the Transducers conferences (International Conference on Solid-State Sensors and Actuators), and is program chairman for 1997. Senturia is a fellow of the IEEE. E-mail, sds@mtl.mit.edu.

Narayan Aluru is currently a research scientist in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. He received a bachelor of engineering degree with honors and distinction from the Birla Institute of Technology and Science, Pilani, India, in 1989, an MS degree from Rensselaer Polytechnic Institute, Troy, N.Y., in 1991, and a PhD from Stanford University in 1995. His research interests include microelectromechanical systems, computational mechanics, semiconductor device and process modeling, numerical methods, and parallel processing. Aluru is a member of ASME and SIAM. E-mail, aluru@rle-vlsi.mit.edu.

Jacob White received his BS degree in electrical engineering and computer science from the Massachusetts Institute of Technology and his SM and PhD degrees in EECS from the University of California, Berkeley. He worked at the IBM T.J. Watson Research Center from 1985 to 1987, was the Analog Devices Career Development Assistant Professor at MIT from 1987 to 1989, was a 1988 Presidential Young Investigator, and is an associate editor for *IEEE Transactions on Computer-Aided Design*. He is currently an associate professor in the Electrical Engineering and Computer Science Department at MIT and is also associated with the MIT Research Laboratory of Electronics. His research focuses on theoretical and practical aspects of serial and parallel numerical algorithms for problems in engineering design. White is a member of IEEE.

Corresponding author: Jacob White, Research Laboratory of Electronics, Massachusetts Institute of Technology, 77 Massachusetts Ave. (Room 36-880), Cambridge, MA 02139-4307, USA; e-mail, white@rle-vlsi.mit.edu.