ELSEVIER

Sensors and Actuators A 58 (1997) 1–11

SENSORS AND ACTUATORS A PHYSICAL

An efficient numerical technique for electromechanical simulation of complicated microelectromechanical structures [1]

N.R. Aluru *, J. White

Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Received 15 August 1996; accepted 26 November 1996

Abstract

An efficient algorithm for self-consistent analysis of three-dimensional (3-D) microelectromechanical systems (MEMS) is described. The algorithm employs a hybrid finite-element/boundary-element technique for coupled mechanical and electrical analysis. The nonlinear coupled equations are solved by employing a Newton-GMRES technique. The coupled algorithm is shown to converge rapidly and is much faster than relaxation for tightly coupled cases.

Keywords: Electromechanical simulations; Microelectromechanical systems; Numerical techniques

1. Introduction

Micromachining technology has enabled the design and fabrication of micromechanical structures such as microsensors, microactuators, accelerometers, and chemical sensors. Although there are many designs which use piezoelectric, thermal, pneumatic, and magnetic actuation, so far the most popular approach is to use electrostatic forces to move micromachined parts. For designers of such microelectromechanical (MEM) devices to investigate design alternatives, they need efficient, robust, and easily used computer simulation tools. Since most of the structures of interest are geometrically complicated, electromechanically coupled, and are inherently three-dimensional, microelectromechanical CAD (MEMCAD) tool developers have been focused on improving the usability, efficiency, and robustness of coupled 3-D electromechanical analysis. The analysis of electrostatically actuated MEM structures involves the tight coupling of mechanical and electrostatic forces. The electromechanical coupling is illustrated through a beam over a ground plane example shown in Fig. 1. When a voltage is applied between the beam and the ground plane, a charge distribution is induced on the surface of the beam. This charge distribution causes the beam to deflect and the charge distribution on the beam surface changes because of the beam deflection. An equilibrium state is obtained when the forces due to the beam deflection and the surface charges balance each other. The beam deflection and the electrostatic charges can be obtained by the solution of coupled problems involving mechanical or elastostatic analysis and electrical or electrostatic analysis.

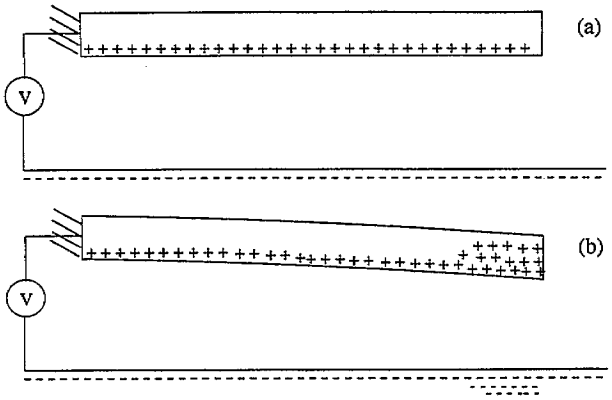

Fig. 1. Illustration of electromechanical coupling problem through a 2-D beam over a ground plane example: (a) applied voltage causes a charge distribution; (b) the deformed structure with charge redistribution.

In order to get the best efficiency for coupled electromechanical analysis, it is desirable to use the fastest methods for each of the domains of analysis. In particular, the fastest 3-D electrostatic analysis programs are based on accelerated boundary-element methods [1,2], and the fastest analysis programs for the large deformations encountered in MEMS are based on Galerkin finite-element analysis. A simple approach to exploiting the existence of these individual solv-

* Corresponding author.

[1] This research was supported by ARPA under ONR contract DABT63-94-C-0053 and FBI contract J-FBI-92-196, by SRC under contract SJ-558, and by grants from IBM and Digital Equipment Corporation.

PII S0924-4247(97)01368-X

ers is to perform the coupled analysis using relaxation [3,4], in which case the major issues are ones of grid interpolation and interprogram communication. Unfortunately, the relaxation approach can converge slowly or not at all, particularly if the structure is very flexible or the electrostatic fields are large, and we shall give examples of this in Section 5. Matrix-free surface-Newton methods were developed to insure convergence, but still use the individual domain solvers as 'black boxes' [5,6]. Although the surface-Newton methods had guaranteed convergence properties, their performance was extremely sensitive to a perturbation parameter used in the matrix-free approach. In addition, each surface-Newton iteration was expensive, requiring a complete nonlinear elastostatic and linear electrostatic solve. In this paper, we present a preconditioned matrix-implicit full Newton method which improves the efficiency and robustness of coupled 3-D electromechanical analysis, but at the cost of requiring some modification of the individual domain solvers.

This paper is organized as follows: the elastostatic and electrostatic analysis are described in Section 2 and Section 3, respectively. A coupled approach to 3-D electromechanical analysis is presented in Section 4. Numerical results are presented in Section 5 and finally conclusions are given in Section 6.

2. Elastostatics

Micromechanical structures undergo large deformation when subjected to electrostatic actuation. The nonlinear structural deformation can be determined by considering the equilibrium of the body in the deformed configuration. Since the deformed structural configuration is not known, the equilibrium equations can be transformed and expressed with reference to either the original undeformed, unstressed configuration or to the last computed deformed configuration. The first approach, where the equilibrium equations are expressed with reference to the original configuration, is defined as the total Lagrangian (or simply Lagrangian) technique and the second approach is defined as the updated Lagrangian technique (see, e.g., Ref. [7] for differences between the two approaches). A key issue with either approach is the appropriate definition of stress and strain measures. In a Lagrangian approach, the technique employed in this paper, the appropriate stress and strain measures are, respectively, the second Piola–Kirchoff stress tensor and the Green–Lagrange strain tensor [8].

Denoting Ω to be the initial configuration occupied by a material body and Γ to be the material boundary, the nonlinear equilibrium equations for finite deformation of a structure, expressed with reference to the initial configuration, are summarized as follows [8]:

$$\nabla\cdot(\nabla\varphi S)+\rho_0 f=0 \text{ in } \Omega \tag{1}$$

$$\varphi=g \text{ on } \Gamma_g \tag{2}$$

$$F_{iA}S_{AB}N_B=h_i \text{ on } \Gamma_{hi} \tag{3}$$

where $F=\nabla\varphi$ is the deformation gradient matrix, φ is the transformation which maps points from the original configuration X to the deformed configuration x, i.e., $x=\varphi(X)$, S is the symmetric Piola–Kirchoff stress tensor, ρ_0 is the density of the material, f is the body force, g are the Dirichlet boundary conditions specified on the boundary Γ_g, the tractions h_i are the natural boundary conditions specified on the boundary Γ_{hi}, $F_{iA}=\partial\varphi_i/\partial X_A$, S_{AB}, $1\le i,A,B\le 3$ denote the components of the deformation gradient and stress tensor, respectively, and N_B is the unit outward normal.

The transformation $\varphi(X)$ can be written as

$$x=\varphi(X)=X+u \tag{4}$$

where u is the unknown displacement vector. The stress components, S_{AB}, are related to the strain components, E_{AB}, by the constitutive equation

$$S_{AB}=\frac{\partial W(E)}{\partial E_{AB}} \tag{5}$$

where $W(E_{AB}(\varphi))$ is the stored energy function of the material. The strain components are related to φ by the relation

$$E_{AB}=\frac{1}{2}\left[\frac{\partial\varphi_i}{\partial X_A}\frac{\partial\varphi_i}{\partial X_B}-\delta_{AB}\right] \tag{6}$$

where δ_{AB} is the Kronecker delta.

Eqs. (1)–(6) summarize the fundamental nonlinear elastostatic model for finite deformation of a structure. The coupling to the electrostatic equations appears through the surface tractions. The surface charges create an electrostatic pressure that acts in the direction of the structure surface normal. The electrostatic pressure on a thick conductor is given by the expression

$$h=\frac{1}{2}E_{\mathrm{n}}{}^{*}\sigma(\mathrm{q}) \tag{7}$$

where σ, the surface charge density, is a function of charge and is determined through a solution of the electrostatic equations, and E_{n} is the normal electric field at the surface.

2.1. Finite-element formulation

The nonlinear elastostatic equations are discretized by employing a Galerkin finite-element method. Denote η to be a test function and let the variational functional spaces $\mathcal{S}$ and $\mathcal{V}$ consist of continuous functions with square integrable first derivatives. The solution space $\mathcal{S}$ is the set of all such functions satisfying the Dirichlet boundary conditions. The weighting or test-function space $\mathcal{V}$ is made up of functions whose value is zero where Dirichlet boundary conditions are specified, i.e.,

$$\{\mathcal{S}=u\,|\,u\in H^1,\ u=g-X \text{ on } \Gamma_g\} \tag{8}$$

$$\{\mathcal{V}=\eta\,|\,\eta\in H^1,\ \eta=0 \text{ on } \Gamma_g\} \tag{9}$$

The weak-form of elastostatics is then stated as follows: given ρ_0, f, g and h_i, find $u\in\mathcal{S}$ such that for all $\eta\in\mathcal{V}$

$$\int_\Omega \nabla\eta:[\nabla\varphi S(E(\varphi))]\,\mathrm{d}\Omega - \int_\Omega \rho_0 f\eta\,\mathrm{d}\Omega - \int_{\Gamma_h} \eta h\,\mathrm{d}\Gamma = 0 \qquad (10)$$

Let $\mathscr{S}^h \subset \mathscr{S}$ and $\mathscr{V}^h \subset \mathscr{V}$ be finite-dimensional approximations to $\mathscr{S}$ and $\mathscr{V}$, respectively. The Galerkin form is then stated as follows: given ρ_0, f, g and h_i, find $u^h \in \mathscr{S}^h$ such that for all $\eta^h \in \mathscr{V}^h$

$$\int_\Omega \nabla\eta^h:[\nabla\varphi^h S^h(E^h(\varphi^h))]\,\mathrm{d}\Omega - \int_\Omega \rho_0 f\eta^h\,\mathrm{d}\Omega - \int_{\Gamma^h} \eta^h h\,\mathrm{d}\Gamma = 0 \qquad (11)$$

To construct a matrix form, the trial and test functions are approximated by linear basis functions, i.e.,

$$u^h = \sum_{a=1}^{nnd} N_a d_a \qquad (12)$$

$$\eta^h = \sum_{a=1}^{nnd} N_a c_a \qquad (13)$$

where *nnd* is the number of finite-element nodes, N_a is the shape function of node a, d_a is the unknown displacement vector of node a, and c_a is the arbitrary weighting function vector. Substitution of Eqs. (12) and (13) into the Galerkin form (11) leads to a nonlinear system of equations which is linearized and solved incrementally for the displacement u. The nonlinear residual equation for a finite-element node, a, is given as

$$R_M^a(u,q) = \int_\Omega \mathscr{B}_a^T \hat{S}\,\mathrm{d}\Omega - \int_\Omega N_a \rho_0 f\,\mathrm{d}\Omega - \int_{\Gamma_h} N_a h(q)\,\mathrm{d}\Gamma \qquad (14)$$

where

$$\mathscr{B}_a = \begin{bmatrix} [\varphi_{,1}^h]^T N_{a,1} \\ [\varphi_{,2}^h]^T N_{a,2} \\ [\varphi_{,3}^h]^T N_{a,3} \\ [\varphi_{,1}^h]^T N_{a,2} + [\varphi_{,2}^h]^T N_{a,1} \\ [\varphi_{,2}^h]^T N_{a,3} + [\varphi_{,3}^h]^T N_{a,2} \\ [\varphi_{,1}^h]^T N_{a,3} + [\varphi_{,3}^h]^T N_{a,1} \end{bmatrix} \qquad \hat{S} = \begin{bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ S_{12} \\ S_{23} \\ S_{31} \end{bmatrix} \qquad (15)$$

$\varphi_{,i}$ denotes differentiation with respect to the ith coordinate and S_{ij} is the i,jth component of the stress tensor. Note that the nonlinear residual equation is a function of displacement and charge.

3. Electrostatics

In electrostatic analysis, the conductor potentials are specified and the potential must satisfy the Laplace equation in the region between the conductors. The charge on each conductor can be determined by solving the integral equation [9]

$$\psi(x) = \int_{\text{surfaces}} \sigma(x')\frac{1}{4\pi\epsilon_0 ||x-x'||}\mathrm{d}a', \quad x \in \text{surfaces} \qquad (16)$$

where $\psi(x)$ is the known conductor surface potential, σ is the surface charge density, $\mathrm{d}a'$ is the incremental conductor surface area, $x, x' \in \mathbf{R}^3$, and $||x||$ is the usual Euclidean length of x given by $\sqrt{x_1^2+x_2^2+x_3^2}$.

A standard approach to solving Eq. (16) numerically for σ is to use a piece-wise constant collocation scheme. That is, the conductor surfaces are broken into n small panels, and it is assumed that on each panel i, a charge, q_i, is uniformly distributed. Then for each panel, an equation is written which relates the known potential at the center of that ith panel, denoted $\bar{p}_i$, to the sum of the contributions to that potential from the n charge distributions on all n panels [10]. The result is a dense linear system,

$$R_E = Pq - \bar{p} \qquad (17)$$

where $P \in \mathbf{R}^{n\times n}$, q is the vector of panel charges, $\bar{p} \in \mathbf{R}^n$ is the vector of known panel potentials, and

$$P_{ij} = \frac{1}{a_j}\int_{\text{panel}_j} \frac{1}{4\pi\epsilon_0 ||x_i - x'||}\,\mathrm{d}a' \qquad (18)$$

where x_i is the center of the ith panel and a_j is the area of the jth panel. Observe from Eq. (17) that the electrostatic residual is a function of conductor geometry and charge, i.e., $R_E(u,q)$.

The dense linear system of Eq. (17) can be solved to compute panel charges from a given set of panel potentials. If Gaussian elimination is used to solve Eq. (17), the number of operations is of order n^3. Clearly, this approach becomes computationally intractable if the number of panels exceeds several hundred. Instead, consider solving the linear system (17) using a conjugate-residual-style iterative method like GMRES [11]. Such methods have the form given below:

- **Algorithm 1**: GMRES algorithm for solving Eq. (17)
- Make an initial guess to the solution, q^0.
- Set $k=0$
- **do** {
  - Compute the residual, $r^k = \bar{p} - Pq^k$.
  - **if** $||r|| < tol$, return q^k as the solution.
  - **else** {
    - **Choose** α's and β in
      - $q^{k+1} = \sum_{j=0}^{k} \alpha_j q^j + \beta r^k$
    - to minimize $||r^{k+1}||$.
    - Set $k = k+1$.
  - }
- }

The dominant costs of Algorithm 1 are in calculating the n^2 entries of P using Eq. (18) before the iterations begin, and performing n^2 operations to compute Pq^k on each iteration. A precorrected fast Fourier transform (FFT) algorithm which, through the use of carefully applied approximations

and transform techniques, avoids forming most of P and reduces the cost of forming Pq^k to the order $n\log n$ operations.

3.1. The precorrected-FFT approach

The precorrected-FFT approach generates an implicit approximation to P which can be used to compute the matrix–vector products Pq rapidly [2]. In this approach, the interaction between nearby panels is computed explicitly. These entries in the potential coefficient matrix, P, are computed by employing Eq. (18).

In a subsequent step, the portion of the matrix–vector product Pq associated with distant interactions is computed by employing an FFT algorithm. Specifically, a 3-D grid is first constructed to include all the panels. The charge in each panel is then projected onto the grid. The potential at the grid points due to grid charges is a 3-D convolution. The convolution can be computed rapidly by employing the FFT algorithm. The grid potentials are then interpolated onto the panels to obtain the potential on each panel. A problem with using this FFT approach to compute distant interactions is that the method unavoidably computes inaccurate approximations to the nearby interactions. Hence, the poor approximation generated by the grid approach to nearby panels is subtracted from the direct interactions. This step is referred to as the precorrection. The matrix–vector product is then obtained by adding the precorrected direct interactions and the grid-approximated distant interactions.

4. Coupled algorithm

In a coupled approach to 3-D electromechanical analysis, the elastostatic and electrostatic equations are solved as a single system. This approach, in comparison to the relaxation algorithm, takes into account the strong coupling between electrical and mechanical systems and in comparison to the surface-Newton technique avoids the several nonlinear elastostatic and linear electrostatic solves during each inner iteration. A coupled approach has been attempted before [12] by employing a finite-element method for both the electrostatic and elastostatic equations. This approach may not be computationally very efficient as a finite-element method, in comparison to a precorrected-FFT accelerated boundary-element method, would require the construction of an exterior mesh for the electrostatic analysis. A computationally efficient approach of a hybrid finite-element/boundary-element method is employed in this paper. A coupled algorithm for electromechanical analysis is shown in Fig. 2. The outer Newton iteration solves the nonlinear coupled system, and the linear system within each Newton iteration is solved using the GMRES algorithm.

A number of issues must be addressed in the coupled approach. First is the efficient computation of the coupled system Jacobian or the matrix–vector product involving the

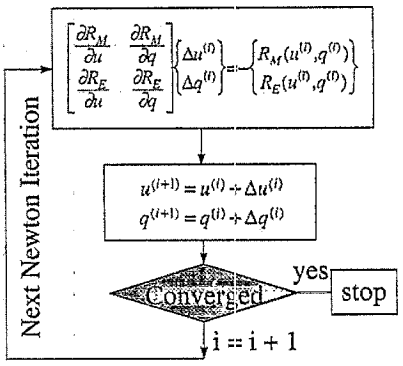

Fig. 2. A coupled algorithm for self-consistent electromechanical analysis.

Jacobian and the displacement/charge vector when employing iterative solution techniques. Second is the storage cost for the coupled system Jacobian. Matrix-free techniques can be used to advantage to minimize storage cost. However, the robustness of the matrix-free technique is very sensitive to the precise choice of the perturbation parameter. If the perturbation parameter is too large, the nonlinearities in the residual will corrupt the derivative estimate. If the perturbation parameter is too small, the small numerical errors in solving the electrostatic and elastostatic problems will corrupt the derivative estimate. Third is the definition of appropriate units for the Jacobian and the residual. The variables in the elastostatic and electrostatic equations represent different units and scales and care should be exercised when computing the Jacobian or the residual. This is easily handled by nondimensionalizing both the elastostatic and electrostatic systems. The issues of storage and efficient computation of the Jacobian are discussed in the following paragraphs.

The coupled system Jacobian can be divided into four parts: the entirely elastostatic part, often referred to as the stiffness or the deformation-coefficient matrix, which determines the change in force due to geometric perturbations; the entirely electrostatic part, which determines the change in potential due to perturbations in surface charge; the electrical to mechanical part, which determines the change in force due to perturbation in surface charge; and the mechanical to electrical part, which determines the change in potentials due to geometric perturbations. The deformation-coefficient matrix is computed by employing a Galerkin finite-element formulation as discussed in Section 2. The formulation accounts for both material and geometric nonlinearities and is summarized in Fig. 3(a). A sparse storage scheme [13] is employed to store the deformation-coefficient matrix. The electrostatic or the potential coefficient matrix is not needed explicitly as the GMRES algorithm, within each Newton iteration, requires only a matrix–vector product involving the matrix, $\partial R_E/\partial q$, and the charge-increment vector, Δq. This matrix–vector product can be computed by employing the precorrected-FFT algorithm discussed in Section 3. In earlier approaches to electromechanical simulations [3–5], a multipole algorithm [14] is employed to compute efficiently the matrix–vector product. The precorrected-FFT algorithm, as

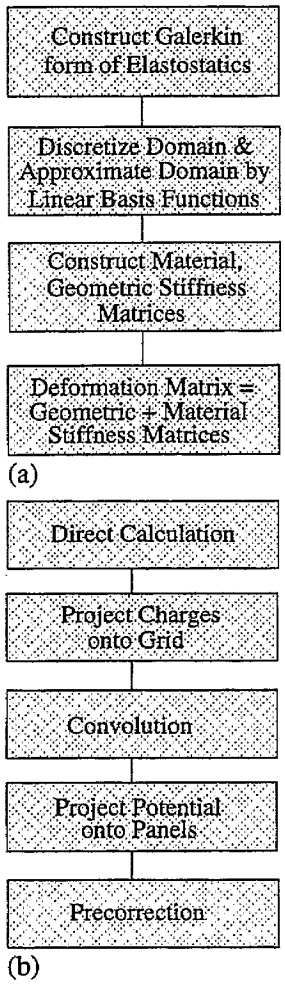

Fig. 3. (a) Computation of the deformation coefficient matrix. (b) Computation of the matrix–vector product employing precorrected-FFT technique.

compared to the multipole algorithm, has been shown to be faster and more memory efficient [2]. The computation of the matrix–vector product with the precorrected-FFT technique is summarized in Fig. 3(b).

4.1. Electrical to mechanical coupling

The electrical to mechanical coupling, $\partial R_M/\partial q$, is obtained from Eq. (14) as

$$\frac{\partial R_M^a}{\partial q} = -\int_{\Gamma_h} N_a \frac{\partial h}{\partial q}\, d\Gamma \tag{19}$$

where R_M^a is the residual equation for the finite-element node a, and noting that $\sigma = q/A$, where A is the panel area, $\partial h/\partial q$ is obtained trivially from Eq. (7).

The computation in Eq. (19) can be performed by computing an equivalent pressure ($\partial h/\partial q$) for each boundary face that belongs to both mechanical and electrical domains. The size of the mechanical to electrical coupling matrix is $3m \times n$ where m is the total number of finite-element nodes on the mechanical domain and n is the total number of surface panels on the electrical domain. This coupling matrix is, however, very sparse as each finite-element node belongs to only a few panels. Hence, only the non-zero entries of the coupling matrix are stored.

4.2. Mechanical to electrical coupling

The mechanical to electrical coupling term can be computed by employing a matrix-free approach. The residual equation for the electrostatic system is given as

$$R_E = Pq - \bar{p} = 0 \tag{20}$$

where P is the potential-coefficient matrix, q is the charge vector and $\bar{p}$ is the vector of applied potentials. An approach to compute the mechanical to electrical coupling term is given as

$$\frac{\partial R_E}{\partial u}\Delta u^{(i)} \approx \frac{P(u+\epsilon\Delta u^{(i)})q - P(u)q}{\epsilon} \tag{21}$$

where ϵ is a small parameter. In order to compute accurately the matrix–vector product, $(\partial R_E/\partial u)\Delta u^{(i)}$, a small value of ϵ is desired and is determined through an optimization problem [15]. For well-scaled residuals, an optimal value of ϵ is $O(\epsilon_m^{1/2})$, where ϵ_m is the machine precision. Unlike the surface-Newton technique, the choice of the matrix-free parameter, ϵ, is not critical and does not affect the robustness or the accuracy of the coupled algorithm [16].

According to Eq. (21), the mechanical to electrical coupling can be computed by performing two matrix–vector products. The first matrix–vector product, $P(u)q$, is straightforward and is computed in the outer Newton loop. The second matrix–vector product, $P(u+\epsilon\Delta u^{(i)})q$, can be obtained by perturbing the panel/conductor geometry from u to $u+\epsilon\Delta u^{(i)}$ and determining the new potential coefficient matrix, $P(u+\epsilon\Delta u^{(i)})$. The second matrix–vector product is computed in the inner GMRES loop by employing the approximations generated by GMRES to $\Delta u^{(i)}$.

The two matrix–vector products required to compute the electrical to mechanical coupling can be obtained by employing either the precorrected-FFT technique or a direct method. In a setup phase, the precorrected-FFT technique computes all the transformation matrices (precorrected direct interactions, projection of panel charges onto grid, FFT and projection of grid potential onto panels). The transformation matrices are a function of the geometry and need to be computed once. In a subsequent evaluation phase, the matrix–vector product can be computed in order $n\log n$ operations. Clearly, the overhead in the precorrected-FFT technique is the setup phase. In order to compute the matrix–vector product, $P(u+\epsilon\Delta u^{(i)})q$, a setup is needed during each iteration of the inner GMRES loop. An additional setup is needed if a preconditioner is to be employed (discussed in Section 4.3). A computationally efficient approach is to employ a direct method, i.e., for nearby panels the potential is computed using the exact analytical formulae [17] and the distant panels are evaluated using quadrature rules. The complexity of the direct

approach is order p^2+2pq, where p is the number of panels on the mechanical domain and $p+q=n$. As will be discussed below, a preconditioner can be applied very efficiently with a direct calculation. In either approach, only two vectors of size n need to be stored to compute the mechanical to electrical coupling.

4.3. Preconditioner

A block diagonal preconditioner of the form

$$\Pi=\begin{bmatrix} R_{Mu}^{-1} & 0 \\ 0 & R_{Eq}^{-1} \end{bmatrix} \tag{22}$$

is applied to the coupled linear system to accelerate the convergence of the GMRES algorithm. In the above equation, Π denotes the preconditioner, $R_{Mu}=\partial R_M/\partial u$ is the deformation coefficient matrix, and $R_{Eq}=\partial R_E/\partial q$ is the potential coefficient matrix. The preconditioned linear system is written as

$$\begin{bmatrix} I & R_{Mu}^{-1}R_{Mq} \\ R_{Eq}^{-1}R_{Eu} & I \end{bmatrix}\begin{Bmatrix} \Delta u^{(i)} \\ \Delta q^{(i)} \end{Bmatrix}=-\begin{Bmatrix} R_{Mu}^{-1}R_M \\ R_{Eq}^{-1}R_E \end{Bmatrix} \tag{23}$$

where $R_{Mq}=\partial R_M/\partial q$ denotes the electrical to mechanical coupling and $R_{Eu}=\partial R_E/\partial u$ denotes the mechanical to electrical coupling.

Consider first the calculation of the right-hand-side vector in Eq. (23). The mechanical and electrical residuals, the deformation coefficient matrix and the transformation matrix (setup phase) of the precorrected-FFT algorithm are computed in the outer Newton loop. The deformation coefficient matrix is stored in sparse ordered and factored form. A sparse solver is then employed to compute $R_{Mu}^{-1}R_M$. $R_{Eq}^{-1}R_E$ is computed by employing the GMRES algorithm. The matrix–vector product needed in the GMRES algorithm is computed by employing the evaluation pass of the precorrected-FFT algorithm. Note that no setup phases are needed inside the GMRES algorithm. Next, consider the computation of $R_{Mu}^{-1}R_{Mq}\Delta q^{(i)}$ and $R_{Eq}^{-1}R_{Eu}\Delta u^{(i)}$ required in the inner GMRES loop. Denoting $v_1^{(j)}$ and $v_2^{(j)}$ to be the jth iteration GMRES approximations to $\Delta u^{(i)}$ and $\Delta q^{(i)}$, respectively, $R_{Mq}v_2^{(j)}$ and $R_{Eu}v_1^{(j)}$ are first computed. A sparse solve is then employed to compute $R_{Mu}^{-1}R_{Mq}v_2^{(j)}$ and a GMRES solver to compute $R_{Eq}^{-1}R_{Eu}v_1^{(j)}$. The preconditioned Newton-GMRES algorithm is briefly summarized in Algorithm 2.

- **Algorithm 2**: Preconditioned Newton-GMRES technique
- **while** not converged / * outer Newton loop * /
  - compute R_M, R_E
  - compute transformation matrices for precorrected-FFT
  - compute and factor R_{Mu}
  - sparse solve to obtain $R_{Mu}^{-1}R_M$
  - use GMRES to compute $R_{Eq}^{-1}R_E$
  - **while** not converged / * inner GMRES loop * /
    - / * steps to compute matrix–vector product in jth GMRES iteration * /
    - compute $R_{Mq}v_2^{(j)}$
    - sparse solve to obtain $R_{Mu}^{-1}R_{Mq}v_2^{(j)}$
    - compute $R_{Eu}v_1^{(j)}$
    - use GMRES to compute $R_{Eq}^{-1}R_{Eu}v_1^{(j)}$
    - / * remaining steps of GMRES not shown * /
  - **end while**
  - update u
  - update q
- **end while**

5. Results

Numerical results are presented for three examples: beam over a ground plane, two silicon bars positioned perpendicular to each other, and a comb drive structure. The performance of coupled and relaxation algorithms is examined for all the examples. In particular, the convergence characteristics and the simulation times are compared.

5.1. Beam examples

Two beam examples over a ground plane structure are considered: the beams are of different dimensions and the convergence characteristics of the relaxation and coupled algorithms are a little different and these are pointed out in the results.

5.1.1. Example 1

The beam example considered here is 500 μm long, 50 μm wide, 14.35 μm thick and is positioned 1 μm above the ground plane. Fig. 4 shows a top view of the beam example. The beam is discretized into 50 parabolic elements and the ground plane is discretized into 250 four-node elements. When a positive potential with reference to the ground plane is applied on the beam, the beam deflects towards the ground plane because of the electrostatic force. As the potential difference increases, the tip of the beam approaches the ground plane, and touches the ground plane for a certain bias defined as the pull-in voltage. The pull-in voltage for the beam considered here is 17.24 V.

Fig. 5 compares the peak deflection obtained from the relaxation and coupled algorithms. The results are identical, verifying the accuracy of the coupled solver. The deflection of the beam for an applied bias of 17.23 V is shown in Fig. 6.

The performance of the relaxation and coupled algorithms for the thick-beam example is summarized in Table 1. Observe that the coupled algorithm takes fewer iterations and is much faster compared to the relaxation algorithm. Fig. 7 compares the convergence of the relaxation and coupled algorithms for the beam and ground plane example. Note that

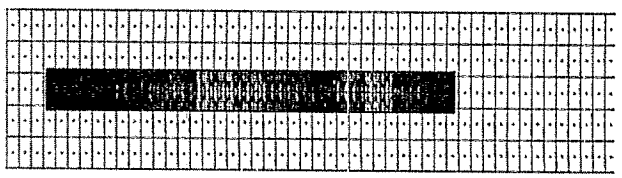

Fig. 4. Top view of a beam over a ground plane example.

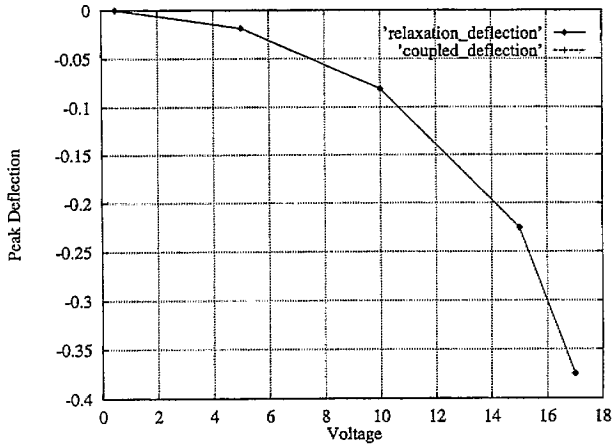

Fig. 5. Comparison of peak deflections from relaxation and coupled algorithms for a beam over a ground plane structure. Note that the two curves overlap.

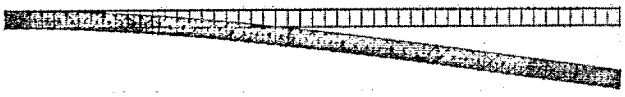

Fig. 6. Deflection of the beam (not to scale) for an applied bias of 17.23 V. Note that the ground plane is not shown in the Figure.

Table 1
Comparison of relaxation and coupled algorithms for number of iterations and CPU (s) for a thick beam and ground plane example

Bias	# Iterations		CPU (s)	
	Relaxation	Coupled	Relaxation	Coupled
2.0	4	2	283.5	368.4
4.0	5	3	381.0	476.2
6.0	6	3	507.7	514.5
8.0	7	3	608.4	572.4
10.0	8	3	710.2	612.4
12.0	10	4	909.5	801.3
14.0	13	4	1244.4	813.4
16.0	20	5	2015.8	1096.0
17.0	41	6	4248.1	1399.3
17.20	94	6	9713.83	1482.5
17.23	200	9	20910.5	2289.8

closer to pull-in the relaxation algorithm converges slowly, but the coupled algorithm still converges rapidly. The slow convergence of the relaxation algorithm near pull-in is due to the increased coupling between elastostatic and electrostatic systems.

The coupled algorithm employs a matrix-free approach to compute the electrical to mechanical coupling. The matrix-free parameter, ϵ, is a concern as the method may not work for all values of ϵ. This issue is investigated and the results are presented in Fig. 8, where the convergence of the coupled algorithm is compared for three different values of ϵ. A value of $\epsilon = 1.0$ requires only one more iteration for convergence as compared to using an optimal ϵ. No noticeable convergence-rate differences are observed between employing an optimal ϵ and an ϵ larger than optimal by a factor of 100.

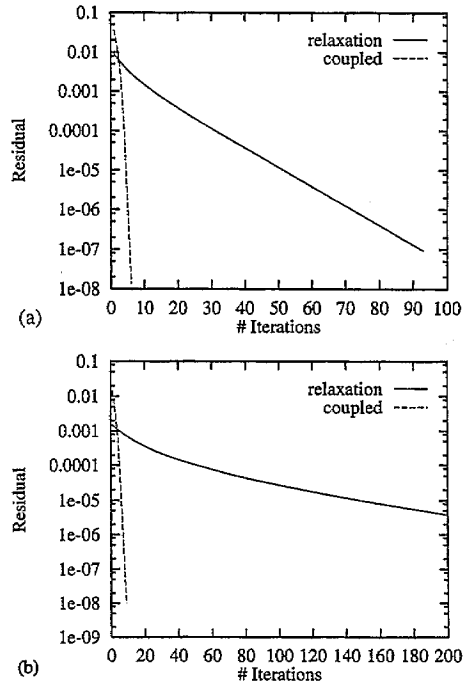

Fig. 7. Convergence of relaxation and coupled algorithms for a beam and ground plane structure: (a) applied bias is 17.20 V; (b) applied bias is 17.23 V.

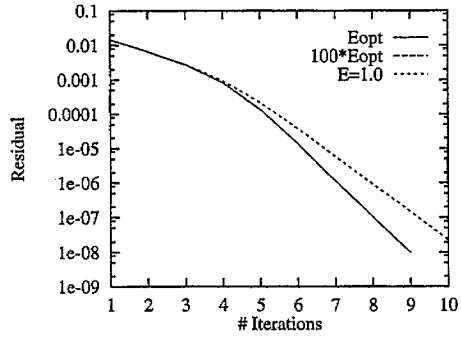

Fig. 8. Comparison of convergence for different values of ϵ. Eopt = sqrt(ϵ_m), 100 * Eopt = 100 · sqrt(ϵ_m), and E = 1 corresponds to $\epsilon = 1.0$. Note that the 100 * Eopt curve overlaps with the Eopt curve.

Hence, the choice of ϵ is not critical for the robustness of the method.

5.1.2. *Example 2*

The beam structure considered is 80 μm long, 10 μm wide, 0.5 μm thick, and is positioned 0.7 μm above the ground plane. Fig. 9 shows a top view of the beam example. The pull-in voltage for the beam considered here is 2.39 V. The beam is discretized into 200 parabolic elements (100 × 2 × 1

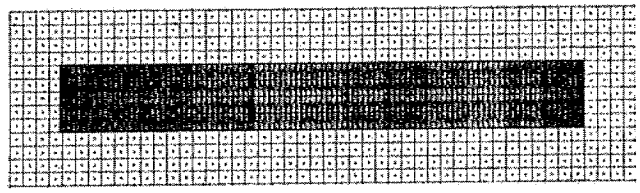

Fig. 9. Top view of a beam over a ground plane example.

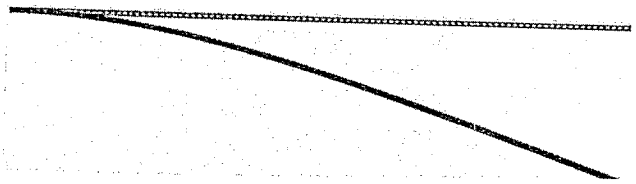

Fig. 10. Deflection of the beam (not to scale) for an applied bias of 2.38 V. Note that the ground plane is not shown in the Figure.

Table 2
Comparison of relaxation and coupled algorithms for number of iterations and CPU (s) for a beam and ground plane example

Bias	# Iterations		CPU (s)	
	Relaxation	Coupled	Relaxation	Coupled
1.0	6	4	3511.4	3965.8
1.5	8	4	4753.5	4443.3
2.0	13	5	7693.5	5532.4
2.25	20	6	11756.6	6908.7
2.35	36	6	20821.9	7406.9
2.38	75	7	42749.0	8801.6

along length, width and thickness, respectively) and the ground plane is discretized into 624 four-node elements. The deflection of the beam at 2.38 V, just before pull-in, is shown in Fig. 10.

The comparison of relaxation and coupled algorithms for the entire bias sweep is summarized in Table 2. As the bias voltage approaches pull-in, the relaxation algorithm converges slowly or fails to converge due to the increased coupling between elastostatic and electrostatic equations. The coupled algorithm, on the other hand, converges faster even as we approach pull-in. Comparison of the simulation times reveals that the coupled algorithm is very competitive with the relaxation algorithm for small bias voltages (see, e.g, the simulation times for an applied bias of 1.0 V). For an applied bias larger than 1.0 V, the coupled algorithm is very efficient and runs much faster compared to the relaxation algorithm. To predict the pull-in voltage for the beam structure, the relaxation algorithm takes a total of 91 285.9 s while the coupled algorithm takes 37 058.7 s. Hence the coupled algorithm is about 2.5 times faster than the relaxation algorithm for this example. The convergence of the relaxation and coupled algorithms for an applied bias of 2.38 V is shown in Fig. 11.

5.2. *Cross bars example*

The cross bars example consists of two silicon bars positioned as shown in Fig. 12. The beam positioned horizontally is the movable part and is positioned 1 μm above the fixed beam in the z-direction. The left-end of the horizontal beam is fixed, so that when a potential is applied on the movable beam, the tip of the movable beam deflects in the z-direction. The movable beam is discretized with 30 parabolic elements and the fixed beam is discretized with 30 linear elements. The deflection of the beam for a potential difference of 2000 V is

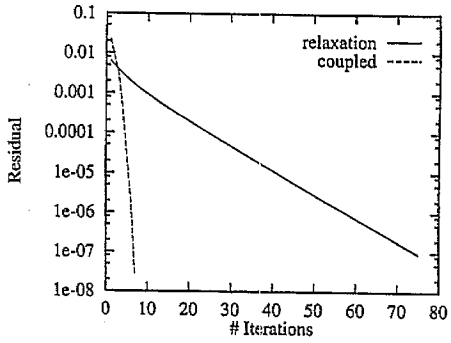

Fig. 11. Comparison of convergence of relaxation and coupled algorithms just before pull-in for the beam and ground plane example.

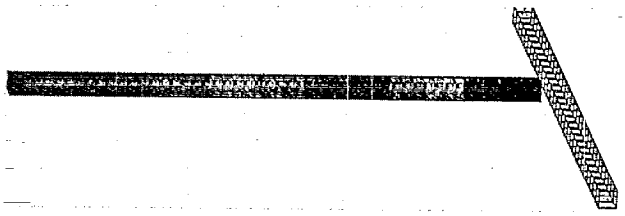

Fig. 12. Two silicon bars at a potential difference of 0 V.

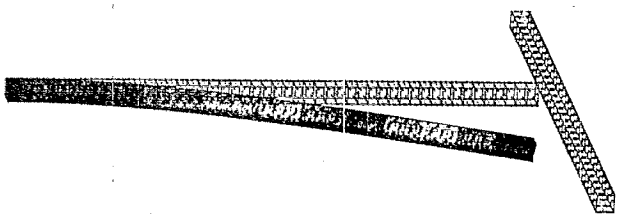

Fig. 13. Deflection of the bar (not to scale) for an applied bias of 2000 V.

Table 3
Comparison of relaxation and coupled algorithms for number of iterations and CPU (s) for the cross bars example (an asterisk indicates that the algorithm fails to converge for the bias)

Bias	# Iterations		CPU (s)	
	Relaxation	Coupled	Relaxation	Coupled
200.0	6	4	449.4	385.8
400.0	11	6	1154.2	597.9
600.0	27	6	3666.8	671.6
700.0	73	6	10984.2	693.1
750.0	200	6	44447.2	682.2
800.0	*	6	*	691.9

shown in Fig. 13. Note that this solution is obtained with the coupled approach as the relaxation algorithm does not converge for a potential difference of more than 750 V.

The performance of the relaxation and coupled algorithms for the cross bars example is summarized in Table 3. The relaxation algorithm fails to converge for a potential difference of more than 750 V across the bars. The convergence of the relaxation and coupled algorithms for 750 V potential difference is shown in Fig. 14(a). Even after 200 iterations the relaxation algorithm fails to converge to less than three orders of magnitude. Fig. 14(b) shows the rapid convergence

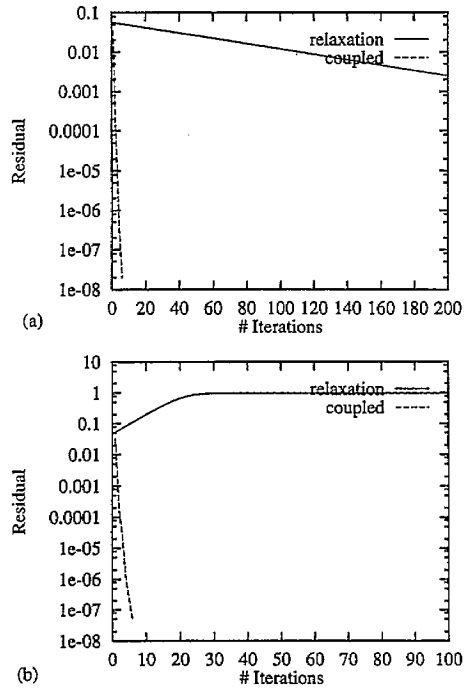

Fig. 14. Convergence of relaxation and coupled algorithms for the silicon bars example: (a) applied bias is 750 V; (b) applied bias is 800 V.

of the coupled algorithm and the breakdown of the relaxation algorithm for an applied bias of 800 V. Table 3 indicates that the coupled algorithm is also very efficient as compared to the relaxation algorithm. For an applied bias of 700 V, the coupled algorithm is about 15.8 times faster, and for an applied bias of 750 V, the coupled algorithm is about 65 times faster compared to the relaxation algorithm.

5.3. Comb drive example

The comb example consists of a deformable comb structure, a drive structure and a ground plane. As shown in Fig. 15, 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. The comb is discretized into 172 parabolic elements, the drive is discretized into 144 linear bricks and the ground plane is discretized into 2688 four-node elements. 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 V is shown in Fig. 16. Note that only the comb structure deforms and the drive and the ground plane do not move.

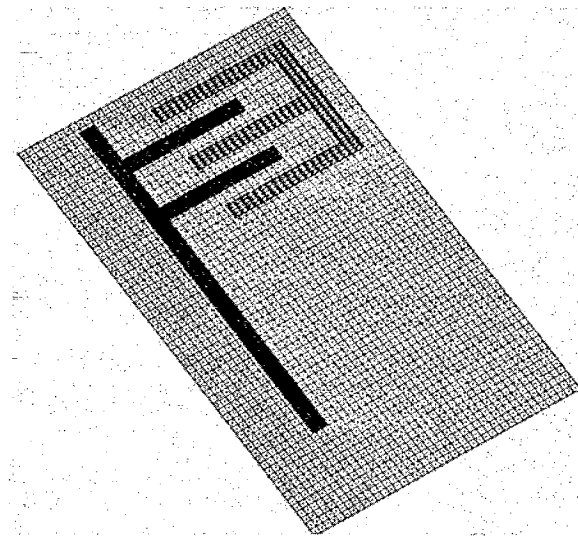

Fig. 15. Comb drive example.

A comparison of the relaxation and coupled algorithms for the comb example is summarized in Table 4. 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, the coupled algorithm takes half as many iterations as the relaxation algorithm and the simulation time for the coupled algorithm is a little longer. For higher voltages, the coupled algorithm con-

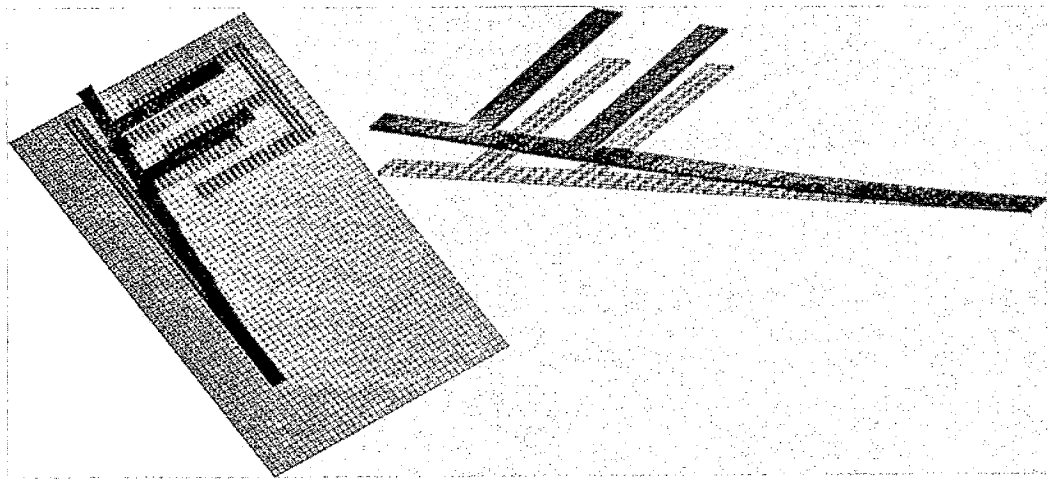

Fig. 16. Deformation of the comb (not to scale) for an applied bias of 85 V. Shown on the left is the complete structure and on the right is just the comb.

Table 4
Comparison of relaxation and coupled algorithms for number of iterations and CPU (s) for a comb drive example (an asterisk indicates that the algorithm fails to converge for the bias)

Bias	# Iterations		CPU (s)	
	Relaxation	Coupled	Relaxation	Coupled
25.0	7	6	3595.4	5589.8
50.0	16	8	9138.0	11833.5
75.0	70	10	42160.3	18590.7
80.0	142	9	81827.0	16670.2
85.0	*	10	*	18490.9

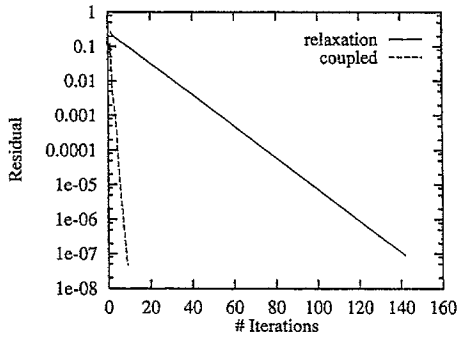

Fig. 17. Comparison of convergence of relaxation and coupled algorithms for a comb example at an applied bias of 80 V.

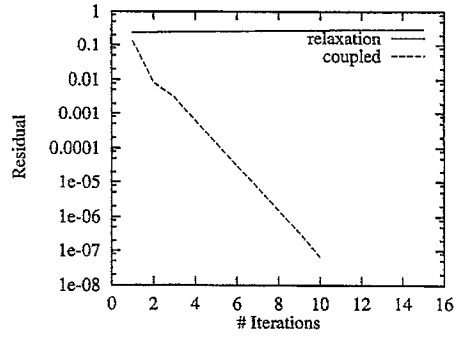

Fig. 18. Comparison of convergence of relaxation and coupled algorithms for a comb example at an applied bias of 85 V.

verges much faster than the relaxation algorithm. For a bias of 80 V, the coupled algorithm is about 4.9 times faster. The convergence of the relaxation and coupled algorithms at 80 V bias is shown in Fig. 17. For an application of 85 V on the drive, the relaxation algorithm fails to converge. The coupled algorithm converges very rapidly and takes only 10 iterations. This is illustrated in Fig. 18.

6. Conclusions

In this paper we presented a coupled algorithm for 3-D electromechanical analysis. The coupled algorithm employs a Galerkin finite-element method for the elastostatic analysis and a boundary-element method with precorrected-FFT acceleration for the electrostatic analysis. The electrical to mechanical coupling term is computed by direct integration and a matrix-free technique is employed to compute the mechanical to electrical coupling term. A block diagonal preconditioner is applied to accelerate the convergence of the GMRES algorithm. Numerical results presented for 3-D electromechanical structures show that the coupled algorithm converges rapidly and is much faster as compared to the relaxation algorithm. Convergence of the coupled algorithm is demonstrated for a cross bars and a comb drive example for which the relaxation algorithm fails to converge.

Acknowledgements

The authors would like to thank Dr John R. Gilbert, Dr Peter M. Osterberg, Joel R. Phillips and Professor Steven D. Senturia for many valuable discussions.

References

[1] K. Nabors and J. White, FastCap: a multipole-accelerated 3-D capacitance extraction program, *IEEE Trans. CAD, 10* (1991) 1447–1459.

[2] J.R. Phillips and J. White, A precorrected-FFT method for capacitance extraction of complicated 3-D structures, *Proc. Int. Conf. Computer Aided Design, Santa Clara, CA, USA, 1994.*

[3] J.R. Gilbert, R. Legtenberg and S.D. Senturia, 3D coupled electromechanics for MEMS: applications of CoSolve-EM, *Proc. MEMS 1995,* pp. 122–127.

[4] X. Cai, H. Yie, P. Osterberg, J. Gilbert, S. Senturia and J. White, A relaxation/multipole-accelerated scheme for self-consistent electromechanical analysis of complex 3-D microelectromechanical structures, *Proc. Int. Conf. Computer Aided Design, Santa Clara, CA, USA, 1993,* pp. 270–274.

[5] 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-electromechanical structures, *Proc. Numerical Process and Device Modeling (NUPAD) V, Honolulu, HI, USA, 1994,* pp. 137–140.

[6] M. Bachtold, J.G. Korvink, J. Funk and H. Baltes, New convergence scheme for self-consistent electromechanical analysis of iMEMS, *Proc. Int. Electron Devices Meet., 1995,* pp. 605–608.

[7] K.J. Bathe, E. Ramm and E.L. Wilson, Finite element formulation for large deformation dynamic analysis, *Int. J. Num. Meth. Engrg., 9* (1975) 353–386.

[8] L.E. Malvern, *Introduction to the Mechanics of Continuum Medium,* Prentice-Hall, Englewood Cliffs, NJ, 1969.

[9] A. Ruehli and P.A. Brennen, Efficient capacitance calculations for three-dimensional multiconductor systems, *IEEE Trans. Microwave Theory Tech., MTT-21* (1973) 76–82.

[10] S. Rao, T. Sarkar and R. Harrington, The electrostatic field of conducting bodies in multiple dielectric media, *IEEE Trans. Microwave Theory Tech., MTT-32* (1984) 1441–1448.

[11] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput., 7* (1986) 856–869.

[12] H.U. Schwarzenbach, J.G. Korvink, M. Roos, G. Sartoris and E. Anderheggen, A microelectromechanical CAD extension for SESES, *J. MEMS, 3* (1994) 162–171.

[13] K.S. Kundert and A. Vincentelli, *Sparse User's Guide: A Sparse Linear Equation Solver*, Univ. of California, Berkeley, 1988.
[14] L. Greengard, *The Rapid Evolution of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA, 1988.
[15] P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, London, 1981.
[16] 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, SC, USA, June 1996*, pp. 54–57.
[17] J.L. Hess and A.M.O. Smith, Calculation of potential flow about arbitrary bodies, *Prog. Aero. Sci., 8* (1966) 1–138.