Simulation Tools for MicroElectroMechanical Systems

by

Deepak Ramaswamy

B.Tech (Civil Engineering) 1996, I.I.T. Kharagpur, India
M.S. 1998, M.I.T.

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Information Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Civil and Environmental Engineering
March 13th, 2001

Certified by .....
Jacob White
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by .....
Kevin Amaratunga
Assistant Professor of Civil and Environmental Engineering
Thesis Committee Chairman

Accepted by .....
Oral Buyukozturk
Chairman, Department Committee on Graduate Students

Simulation Tools for MicroElectroMechanical Systems

by

Deepak Ramaswamy

Submitted to the Department of Civil and Environmental Engineering
on March 13^{th}, 2001, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Information Technology

Abstract

In this thesis efficient techniques to solve complex 3-D electromechanical problems are developed. Finite element discretization of complex structures such as the micromirror lead to thousands of internal degrees of freedom. Their mostly rigid motion is exploited leading to a mixed rigid-elastic formulation. This formulation's advantage is apparent when it is incorporated in an efficient coupled domain simulation technique and examples are presented exploring geometry effects on device behavior. Then for system level simulation where full device simulation costs add up we need models with much reduced order with little degradation in accuracy. We describe a model reduction formulation for the electromechanical problem based on implicit techniques which accurately capture the original model behavior.

Thesis Supervisor: Jacob White
Title: Professor of Electrical Engineering and Computer Science

Thesis Committee Chairman: Kevin Amaratunga
Title: Assistant Professor of Civil and Environmental Engineering

Acknowledgments

I would like to extend my sincere appreciation and thanks to my advisor, Professor Jacob White, for his all around guidance, insights and patience from the beginning of my research. Learning from him has been an extremely invaluable experience. I am also very grateful to Professors John Williams and Kevin Amaratunga of the Information Technology program in the Department of Civil and Environmental Engineering for agreeing to serve on my committee, and for their help and advice on my academics.

I am very aware of my eternal debt towards my parents, Krishnamurthi and Lalitha, without whose sacrifices and love, I would not be writing this thesis. Finally I am very thankful for having two amazing long-distance "advisors" in the forms of my brothers Kartik and Arun.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Design of MicroElectroMechanical Systems (MEMS) is often a challenging problem because of the multi-physics involved. This should typically mean that designers would be making extensive use of existing design tools. However this is not the case because the tools take an enormous amount of time to simulate and optimize structures with relatively simple geometry. The aim of this thesis is explore efficient and judicious ways of simulating a specific (and common) multi-physics - electromechanical - device and then efficiently reducing the computational cost of solving the device, making it suitable for system level simulation.

1.1 Outline

In the next section we motivate the study of efficient algorithms by presenting widely used examples. In Chapter 2 we present background material on the GMRES, Newton-Krylov methods, the mechanical and electrostatic solvers which will be used while developing coupled domain algorithms and model order reduction methods. Chapter 3 describes the multi-level Newton algorithm for structural domains treated as being fully elastic. Chapter 4 describes the mixed rigid-elastic formulation for structures with mostly rigid behavior thereby greatly reducing the computation. Chapter 5 extends the rigid-elastic and coupled domain formulations to the dynamics case. Chapter 6 describes model order reduction for our coupled system with fully elastic

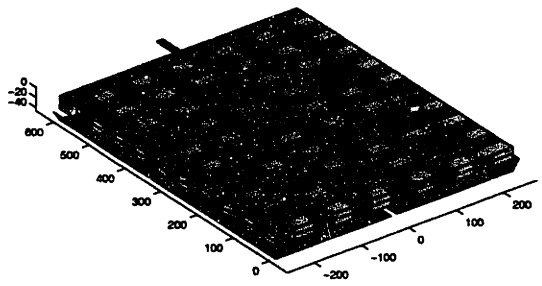


Figure 1-1: Scanning mirror

structures. Here we show how to capture the quadratic dependency of the electrostatic force on the voltage using implicit methods only. Chapter 7 extends the reduction technique of Chapter 6 to rigid-elastic systems to generate rigid-elastic models. Finally we conclude in Chapter 8 by summarizing the main results and implications reached in this thesis. Appendix A relates the reachability space of the ordinary differential equation to the expansion-around-zero based model reduction. Appendix B explores a hypothesis as to why block Arnoldi based reduction performs better than individual Arnoldi reduction.

1.2 Motivation

Some examples of common electromechanical systems are the comb-drive accelerometer, resonator, mirror (Figures 1-1,1-2, 1-3) and angular-rate sensors. But electromechanical systems are a small fraction of an entire array of MEMS devices [1]. For instance there are microfluidic systems such as micromixers,micropippetes and capillary electrophoresis, protein analysis and particle analysis chips, microarrays used as biosensors useful in diagnostics, wireless patient monitoring and genetic testing [2], numerous chemical engineering applications such as microreactors where there are advantages such as increased power efficiency, to be had in scaling down the system

[1] see for eg. http://mems.isi.edu for current happenings in the MEMS industry
[2] see for eg. http://www.lab-on-a-chip.com

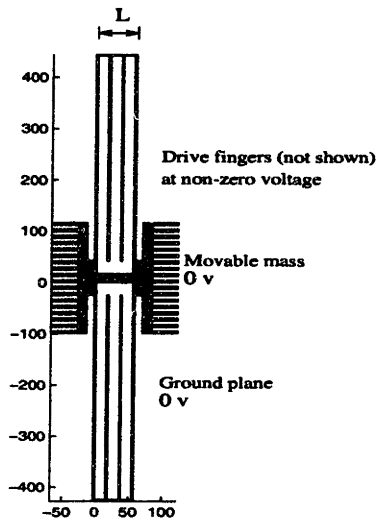


Figure 1-2: Comb drive resonator

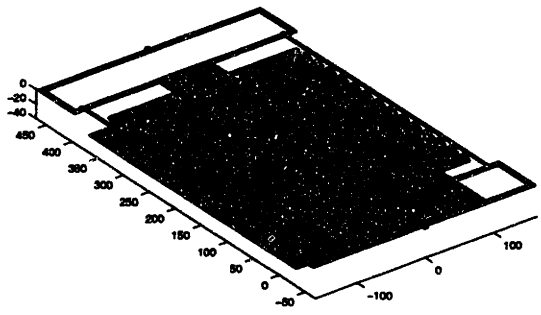


Figure 1-3: Accelerometer

and optical applications (besides the aforementioned mirror) such as variable optical attenuators and tunable optical filters, to name but a few.

The focus of this thesis, however, is on electromechanical systems and the driving force in these devices is the electrostatic pressure generated by keeping the distinct surfaces of the device at different voltages. The accelerometer has applications in car airbags, car and personal navigation, computer peripherals and earthquake detection [3]to name a few. The accelerometer schematic of Figure 1-4 shows a movable proof mass, and drive fingers component which is anchored with the help of tethers. The sense fingers (shown in a lighter color) are also anchored but the fingers can flex. V_I is the input voltage applied to the proof mass system and V_O is the output of interest. The inter-finger gap between the sense and the drive fingers results in a differential capacitor with capacitance $C(u)$ where u is the geometric perturbation (during sudden acceleration) of drive fingers and the movable proof mass system. The capacitance and the position information can be obtained through further processing of V_O and a force feedback achievable through varying V_I can also prevent the fingers from contact (besides standard advantages such as increased bandwidth etc) (for details, see [1] for eg.).

The resonator has potential applications in areas such as wireless systems in the form of filters and oscillators for example, replacing quartz crystals. In Figure 1-5, the central proof mass and tether system will rock back and forth sideways with significant displacement when the frequency of the drive signal V_d is the resonance frequency of the structure and then the sense fingers pick up the motion (see [2] for an introductory treatment).

The mirror in Figure 1-1 will twist because the mirror is kept at 0 v and the pair of thin drive electrodes below it are kept at different voltages. By altering the voltages the mirror can be made to twist to different angles. Micromirrors are simple devices which have proved useful in projection displays and switches for optical networks (Figure 1-6). In the projection display , a display pixel can be lighted up or not depending on the voltage that is applied to the mirror. Similarly the switch can allow

[3]http://www.analogdevices.com/industry/iMEMS

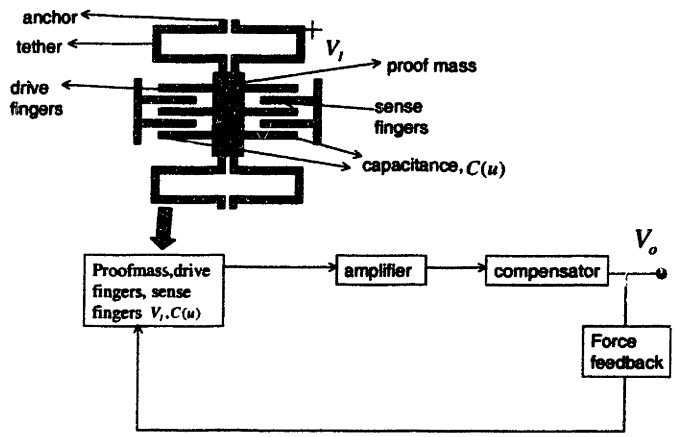


Figure 1-4: Accelerometer schematic (drive components in darker color)

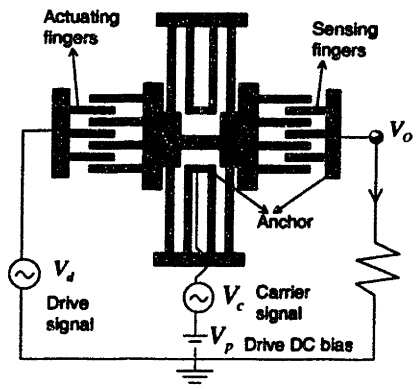


Figure 1-5: Resonator schematic, from [2]

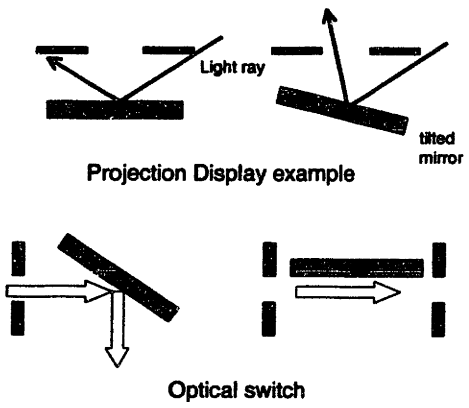


Figure 1-6: Micromirror applications examples - schematics

light to go onto the next fiber or not depending on the applied voltage.

1.3 Self-consistent Solution

In the mirror case for example, one can calculate the forces on the mirror just once when the mirror is in rest position and then use that initial force to calculate the twist of the mirror (either quasistatic or in time). However the problem is that while the electrostatics is a linear problem and the structure could perhaps be approximated as being linear, when one puts them "together" the system is nonlinear. This is illustrated in Figure 1-7 where the forces developed in the original configuration deform the mirror to a position where the external electrostatic forces balance the internal material stress generated forces. But in the deformed position a new set of electrostatic forces are developed which will not balance the internal forces. The problem therefore is of self consistency and therefore may be stated in words as follows - determine the geometry of the structure such that the forces generated for that geometry exactly balance the internal structural forces.

Figure 1-7: Need for self-consistent solution

Chapter 2

Background

2.1 Introduction

In subsequent chapters the multi-level Newton algorithm will be described. The method uses two key sub-algorithms namely, GMRES and Newton-GMRES to solve the coupled domain problem. Therefore we give a brief description of these methods below. Then we describe the mechanical and electrostatic solvers and present model order reduction techniques.

2.2 GMRES

The Generalized Minimal Residual Solver (GMRES) [3] along with the Multigrid based methods is one of the algorithms of choice for iteratively solving large sparse linear systems. To solve the system

$$Ax = b \tag{2.1}$$

at each iteration k, GMRES minimizes w.r.t. y_k the residual

$$r_k = ||b - AQ_k y_k|| \tag{2.2}$$

where Q_k is an orthogonal basis of the Krylov subspace

$$K_k =< b, Ab, A^2b, A^3b, \ldots, A^{k-1}b > \tag{2.3}$$

Additionally we have the relation $AQ_k = Q_{k+1}\hat{H}$ where $\hat{H}$ is a $(k+1) \times k$ upper Hessenberg matrix and $H = Q_k^T A Q_k$ where H is a $k \times k$ upper Hessenberg matrix (truncation of $\hat{H}$). When $r_k = 0$, K_k is invariant i.e. $K_k = K_{k+1} = K_{k+2}$ etc. Note that although we have $||r_0|| \geq ||r_1|| \geq ||r_2|| \ldots \geq ||r_k||$, in general we *do not* have (with $x_k = Q_k y_k$) $||x - x_{k-1}|| \geq ||x - x_k||$ or in other words

$$||x - x_0|| \overset{?}{\geq} ||x - x_1|| \overset{?}{\geq} ||x - x_2|| \ldots ||x - x_k|| \tag{2.4}$$

i.e. convergence to the true solution is not necessarily monotonic.

For GMRES to be computationally attractive the cost per iteration step should be small and the number of iteration steps before $||r_k||$ falls below a tolerance should be small. The cost per step depends on the product of A with some vector in the Krylov subspace. Therefore if A is sparse or if the product can be evaluated using efficient techniques then the cost can be smaller than $O(N^2)$ where N is the dimension of A. The number of iteration steps can be significantly reduced by preconditioning A to have a condition number close to unity. A good preconditioner usually approximates A^{-1} and is cheap to compute and apply. Applying A^{-1} itself as a preconditioner is of course redundant as we can then get the solution directly.

2.3 Newton-Krylov

Consider a nonlinear system of equations $f(x) : R^N \rightarrow R^N$. To determine a solution $x_* : f(x_*) = 0$ we can use Newton iteration defined as

$$x_{k+1} = x_k - Df(x_k)^{-1} f(x_k) \tag{2.5}$$

where D is the differentiation operator. $Df(x_k)^{-1} f(x_k)$ can be computed by

direct factorization of $Df(x_k)$ *or* can be computed iteratively using for example a Krylov subspace based method. In that case the method is called a Newton-Krylov method [5]. We state a standard result from [4] regarding the convergence of Newton's method.

Theorem : Let $closure\{S_0\} \subseteq S$ where S is an open set and S_0 is a convex set. Let $f(x) : S \rightarrow R^N$ be differentiable $\forall x \in S_0$ and continuous $\forall x \in S$. Additionally we define the neighborhood of a point $x_0 \in S_0$ as

$$nbd_r(x_0) = x \ s.t. ||x - x_0|| < r \subseteq S_0$$

and the iteration is defined by (2.5). Then if

$$\begin{aligned} ||Df(x) - Df(y)|| &\leq \gamma||x - y|| \ (Lipschitz \ continuity) \\ ||Df(x)^{-1}|| &\leq \beta \ \forall \ x, y \in C_0 \\ ||Df(x_0)^{-1} f(x_0)|| &\leq \epsilon \\ h = \alpha\beta\epsilon &< 1 \\ r &= \alpha/(1i - h) \end{aligned} \tag{2.6}$$

then $\forall k > 0$,

$$||x_k - x_*|| \leq \alpha \frac{h^{2^{k-1}}}{1 - h^{2^k}},$$

$x_k \in nbd_r(x_0)$, x_k converges to x_* and since $h < 1$ the convergence is at least quadratic. In other words the iteration defined by (2.5) results in quadratic local convergence if the initial guess x_0 is "close enough" to x_*.

However in the case of Newton-Krylov x_{k+1} is not computed exactly as the linear solution iteration is terminated in $i << N$ steps. Therefore the residual(typically $\neq 0$) is

$$r_k = f(x_k) - Df(x_k)(x_{k+1} - x_k) \tag{2.7}$$

The actual expression for x_{k+1} is now

$$x_{k+1} = x_k - Df(x_k)^{-1}(f(x_k) - r_k) \tag{2.8}$$

The Newton-Krylov method falls into a more general class of modified inexact Newton methods (see for eg. [6], [8]). The problem is then to determine a stopping criterion for the linear solution (inner) iteration so as to preserve superlinear convergence as much as possible. [8] shows that linear convergence can be achieved with the stopping criterion for the inner iterative scheme defined as $\frac{||r_k||}{||f(x_k)||} < tolerance < 1$ (provided certain conditions are satisfied). Without being rigorous Newton-Krylov convergence can be explained as follows (see [4], [8] for details).

First we rewrite the third inequality of (2.6) to get the conditions,

$$\begin{aligned} ||Df(x) - Df(y)|| &\leq \gamma||x - y|| \ (Lipschitz\ continuity) \\ ||Df(x)^{-1}|| &\leq \beta \ \forall \ x, y \in C_0 \\ ||Df(x_0)^{-1}[f(x_0) - r_0]|| &\leq \epsilon \end{aligned} \tag{2.9}$$

Based on these conditions it can be shown [4]

$$||f(x) - f(y) - Df(y)(x - y)|| \leq \frac{\gamma}{2}||x - y||^2 \tag{2.10}$$

From (2.10) and if the inner iteration is stopped when

$$\frac{||r_k||}{||f(x_k)||^2} \leq \alpha < 1 \tag{2.11}$$

then

$$\begin{aligned} ||f(x_{k+1}) - f(x_k) - Df(x_k)(x_{k+1} - x_k)|| &\leq \tfrac{\gamma}{2}||x_{k+1} - x_k||^2 \ or \\ ||f(x_{k+1}) - r_k|| &\leq \tfrac{\gamma}{2}||Df(x_k)^{-1}(f(x_k) - r_k)||^2 \ from\ (2.8) \Rightarrow \\ ||f(x_{k+1}|| - ||r_k|| &\leq \tfrac{\gamma\beta^2}{2}||f(x_k) - r_k||^2 \Rightarrow \\ ||f(x_{k+1}|| &\leq ||r_k|| + \tfrac{\gamma\beta^2}{2}(||f(x_k)|| + ||r_k||)^2 \\ &= (1 + \tfrac{\gamma\beta^2}{2}(1 + \alpha||f(x_k)||)^2)||f(x_k)||^2 \Rightarrow \\ ||f(x_{k+1}|| &\leq (1 + \tfrac{\gamma\beta^2}{2}(1 + \alpha||f||_{max})^2)||f(x_k)||^2 \\ ||f(x_{k+1}|| &\leq \delta||f(x_k)||^2 \ where\ \delta = 1 + \tfrac{\gamma\beta^2}{2}(1 + \alpha||f||_{max})^2 \end{aligned} \tag{2.12}$$

where $||f||_{max}$ is the maximum value of $||f(x)|| \ \forall x \in S_0$.

While we will not show whether x_k converges quadratically to x_*, (2.12) shows that the function value, if it decreases, decreases at least quadratically. [7] describes an inexact Newton solver which has better global convergence properties because of backtracking i.e. a step is taken in the Newton direction such that residual decreases w.r.t. a certain tolerance to ensure convergence. Also note that (2.11) is practically overly restrictive and leads to over-solution (Chapter 3 provides an example) at each step and [7] describes methods to get the proper tolerances to get an adequate solution of (2.5).

2.4 Linear elasticity based FEM

We will assume linear elasticity for the mechanical system formulation. For *geometrically* nonlinear structures the nonlinear finite element method is the technique of choice. For now we assume the material is completely elastic and the finite element method is applied to the quasistatic linear elasticity p.d.e.

$$div\ \sigma(u) + f = 0 \tag{2.13}$$

where f is the body force and

$$\sigma_{ij}(u) = \frac{E}{2(1+\nu)}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{E}{(1+\nu)(1-2\nu)}\delta_{ij}\frac{\partial u_k}{\partial x_k}$$

is the Cauchy stress tensor where E is the Young's modulus, ν the Poisson ratio, δ_{ij} the Kronecker delta symbol with all repeated indices are summed, x is a point in the undeformed configuration, u is that point's displacement, and incorporating the stress(Neumann) and displacement(Dirichlet) boundary conditions leading to a set of non-linear force equilibrium equations (Total Lagrangian formulation [9]). Assuming homogeneous boundary conditions for simplicity, let $u, v \in H^1(\Omega) = \{w | w \in L^2(\Omega), \nabla w \in (L^2(\Omega))^3\}, L^2(\Omega) = \{w | \int_\Omega w^2 dV < \infty\}$. Then applying the finite ele-

ment method to the above p.d.e. results in the weak statement

$$a(v,u) = \ell(v) \tag{2.14}$$

where

$$a(v,u) = \int_\Omega \frac{\partial v_i}{\partial x_j}\sigma_{ij}(u)\, dV \tag{2.15}$$

$$\ell(v) = \int_\Omega f_l v_l\, dV \tag{2.16}$$

From the form of $a(v,u)$, we see that the displacements u obtained by solving the above system in turn change the domain of integration Ω . Therefore the problem is geometrically nonlinear. For further details on the Jacobian of the weak form (2.14), see [9]. The discretization breaks up the structure into elements with nodes and through a set of equations relate the node forces (surface and volume forces projected onto nodes) to the nodal displacements which are interpolated to get a displacement solution over the entire domain. The resulting discretized system

$$F(u) = P(u, f, external\ force)^1 \tag{2.17}$$

is sparse. Here u is a discrete vector defined only for the nodes and not continuous.

2.4.1 Dynamics

Over and above the elastostatic case we have the mass matrix contribution as

$$\rho\frac{\partial^2 U}{\partial t^2} \Rightarrow \int_\omega \rho\ddot{w}\, dV \Rightarrow (\int_{\omega_o} \rho_o \phi_i \phi_j\, dV)\ddot{u_j} = M_{ij}\ddot{u_j} \tag{2.18}$$

where ρ, ρ_o are the densities in the current and reference configurations, w the finite element test vector, u the actual solution (u will here on mean fully elastic), $\phi_{i,j}$ the test basis vectors and u_j is the basis (nodal) component.

[1] f as in (2.13)

$$M\ddot{u} + F(u) = P(u, q) \tag{2.19}$$

Note that the mass matrix is constant because the integral in (2.18) involves the original configuration quantities namely ω_o and ρ_o and $\phi_{i,j}$ here depends on the original configuration.

2.5 Fast Electrostatic Solver

The electrostatic solver is based on the precorrected FFT accelerated boundary element solver [10]. This solver discretizes

$$\phi(x) = \frac{1}{4\pi\varepsilon} \int_{surface} \frac{q(x')}{||x - x'||} ds' \tag{2.20}$$

The discretized version of (2.20) is

$$\phi = A(u)q \tag{2.21}$$

where in discretization by collocation $A \in R^{N \times N}$, $\phi, q \in R^N$ and

$$A_{i,j} = \int_{panel_j} \frac{1}{4\pi\varepsilon ||x_i - x'||} ds'$$

where x_i is the i^{th} collocation point. Here the surface of the structure is broken up into panels and the system of equations for the N panel charges is obtained by setting the sum of the potentials due to these charges equal to the given potential at N test or collocation points.

The Dirichlet condition of $\phi(\infty) = 0$ is implicit in this equation. However this tells us that it that the sum of charges on the surfaces of objects located at finite distances will not equal zero and that the force between surfaces depends on the absolute value of the potentials. To have this zero net charge and to have the force depend only the relative potentials (meaning that a zero differential voltage implies zero force), we would need to have $\nabla\phi(\infty) = 0$ (the Neumann condition). We modify the above

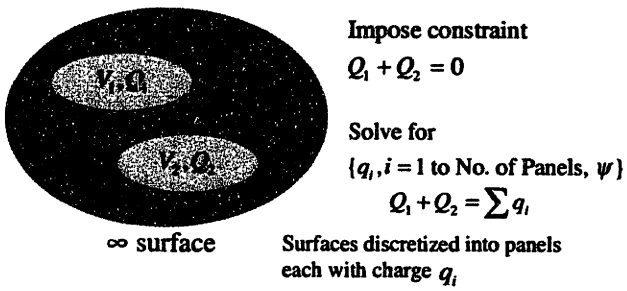


Figure 2-1: Maintaining Charge Conservation

equation to enforce both conditions at infinity to be true. An extra potential ψ is added to the conductors and an additional charge conservation equation is invoked as shown in Figure 2-1.

$A(u)$ is a dense $N \times N$ matrix and is therefore not only expensive to solve via direct factorization but also not suitable for a Krylov-subspace based iterative method without an effective replacement computation of the matrix vector product.

2.5.1 Pre-corrected FFT

The pre-corrected FFT method as described in [10] essentially consists of approximating the "potential coefficient matrix × charge" i.e. (2.21) product by the following steps (Figure 2-2).

Projection Step

$$\hat{q} = Wq$$

where $\hat{q}$ is the grid point charges and q is the panel charge vector. A uniform 3-D grid is overlayed on the problem domain and the charges q are projected onto the grid points with the W operator which is defined as follows. For a grid cell k, the corner grid point charges,$\hat{q}(k)$ are required to match potentials at

test points with the potential due to the charges,$q(k)$ inside the grid cell. The test points typically lie on a sphere encompassing the grid cell (Figure 2-3), $A^{gt}\hat{q}(k) = A^{pt}q(k)$ leading to the definition $W = [A^{gt}]^{\dagger}A^{pt}$ [2] (to be computed only once) Here $A^{gt}_{i,j} = \frac{1}{||x_i - x_j||}$ (or inverse of distance between grid point i and test point j) and $A^{pt}_{i,j} = \int_{panel\ j} \frac{1}{||x_i - x'||} ds'$

Convolution Step

$$\hat{\phi} = H\hat{q}$$

Grid point potentials due to the grid point charges are computed with FFT's. The potential at grid point j due to an unit charge at point i is proportional to $\frac{1}{||x_j - x_i||}$. The H matrix contains this interaction between grid points and is a block circulant matrix whose product with a vector can be exactly computed with FFT's as below. $\tilde{H}$ below is the discrete Fourier transform of H (to be computed only once).

$$\begin{aligned} \hat{Q} &= FFT(\hat{q}) \\ \hat{\Psi} &= \tilde{H}\hat{Q} \\ \hat{\phi} &= FFT^{-1}(\hat{\Psi}) \end{aligned}$$

This is the stage where the pre-corrected FFT algorithm gets it's computational speed over the direct computation with it's $O(nlog\ n)$ complexity where n is the size of the FFT grid.

Interpolation Step

$$\phi = V^T\hat{\phi}$$

In the collocation scheme, the matrix entries $V(k,j)^T$ for grid points of cell k and a point charge j correspond to $V(k,j)$ which projects the point charge j onto the grid points. For Galerkin based methods, $V = W$ is used.

[2] "†" stands for pseudo-inverse

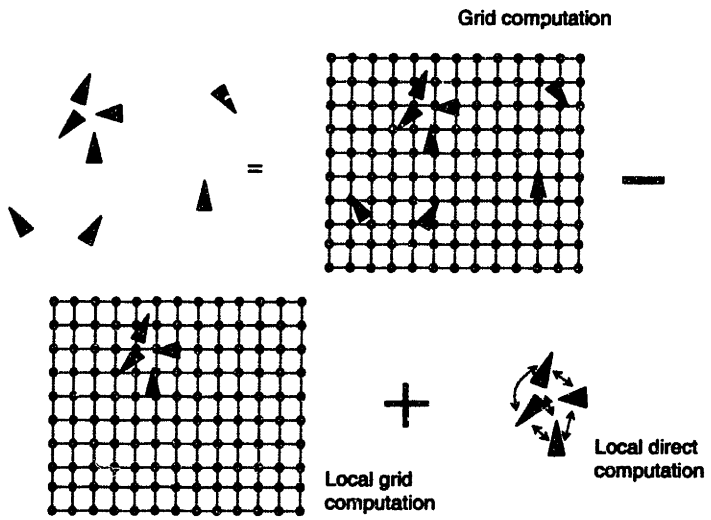


Figure 2-2: Pictorial representation of pre-corrected FFT

Nearby Interactions

$$\phi_{k,\ell} = \phi_{k,\ell} + (A_{k,\ell} - V_k^T H_{k,\ell} W_\ell) q$$

Here $\phi_{k,\ell}$ is the potential contribution of cell ℓ to cell k, $A_{k,\ell}$ represents the part of A which is computed directly, V_k^T represents the interpolation of potential contributions from the grid points of k on to the panels of k, W_ℓ is the part of the W matrix representing the projection of panel charges of ℓ on to the grid points of ℓ and $H(k,\ell)$ is simply that part of the H matrix representing the interactions between grid points of ℓ and the grid points of k. Additionally ℓ is a "neighbor" of k. The grid approximation works well for panels which are located at a relatively large distance w.r.t. each other. However the approximation does not work well for panels in cells which are neighbors. To counter this, the neighboring cell interactions, $A_{k,\ell}$, are computed directly and the nearby cells grid projections, $V_k^T H_{k,\ell} W_\ell$ are removed.

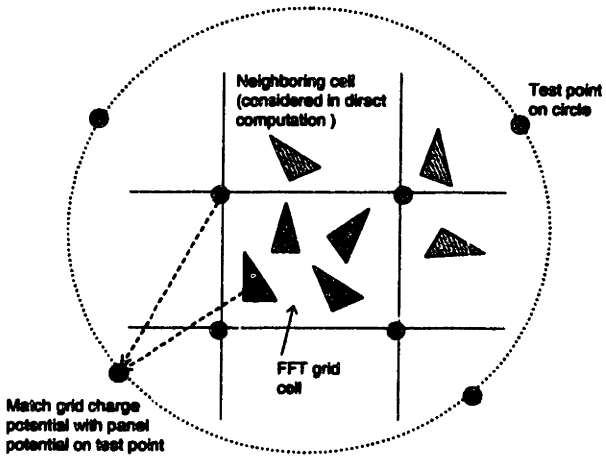


Figure 2-3: Pictorial representation of Pre-corrected FFT matching potentials at test points

Two additional notes are that the accuracy of the scheme (or the order of the method) can be controlled by varying the test sphere radius and choosing the test points (in the projection step above) to be quadrature points of a variable order integration method (see [10] for details) and as the FFT grid is made finer while the FFT cost increases monotonically, the direct panel computation cost decreases monotonically. To find the optimum grid size, while one can attempt to minimize the cost function, from a standard result of the "parameter balancing technique" (see Appendix C), at the grid size for which the cost of the FFTs match the cost of the direct interactions, the overall cost will not be more than a factor of 2 times the minimum cost, so this also might be acceptable.

While solving for problems with a ground plane we use a modified Green's function for (2.20) to avoid having to discretize the ground plane, saving us significant computation.

2.6 Model Reduction Techniques

If we have a model representation of a system which takes an input(s) and returns an output(s), many a time, in higher level system simulation/optimization applications

the computational complexity of getting an output for a given input from the model is simply too high. Thus model order reduction involves reducing the computational complexity of the model by reducing the number of parameters in the original model. If the original model is nonlinear, we can aim to obtain either nonlinear or linear reduced models. If the original model is described by linear ordinary differential equations then a typical approach is to write down the algebraic relation between the input and output for the O.D.E. in the frequency domain and then somehow approximate this relation leading to significantly less computation. Additionally if the original model is stable then we would like to have a stable reduced order model. Then there might also be additional properties we would like to preserve like maintaining passivity in electrical circuits.

If the O.D.E. is given by

$$\begin{aligned} \dot{x} &= Ax + b \\ y &= c^T x \\ Y(s) &= \underbrace{c^T(sI - A)^{-1}b}_{G(s)} X(s) \end{aligned} \tag{2.22}$$

where $A \in R^{N \times N}$, $b, c \in R^N$ and y is the output. The model is stable if and only if all the eigenvalues of A reside in the left half complex plane. There are several approaches for model order reduction described in the literature.

2.6.1 Padé Approximation

One can approximate $G(s)$ by a rational function $\hat{G}(s)$ as

$$\hat{G}(s) = \frac{\sum_0^q d_i s^i}{1 - \sum_1^p a_i s^i}$$

The coefficients a_i and d_i are obtained by matching the first $(p + q - 1)$ [3] Taylor series coefficients of $G(s)$ ($c^T A^{-i} b$ - also called moments) and $\hat{G}(s)$. Directly com-

[3]Typically $q = p - 1$ since one can see from the eigen representation that the denominator of $G(s)$ has an extra power of s

puting the moments is numerically unstable and [14] demonstrates a stable method using biorthogonal Lanczos.

2.6.2 Eigen Analysis

It has been a traditional approach to perform eigenanalysis to generate the the reduced model i.e. to determine a reduced model of dimension r (instead of N), one determines the set of eigenvectors corresponding to the r smallest absolute eigenvalues, V_r. Then

$$x = V_r z \tag{2.23}$$

$$\dot{z} = V_r^T A V_r z + V_r^T b \tag{2.24}$$

represents the reduced model. As a simple example, let

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and we are interested in generating a reduced model of order one. Now assuming $x(0) = 0$ the solution to (2.22) is

$$x(t) = \int_0^t e^{(t-\tau)A^{-1}} A^{-1} b v(\tau) d\tau$$

from which we infer that the reachability space (for x) for the example here is simply a span of

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

However eigenreduction chooses V_r to correspond to the smallest eigenvector of A i.e. 0.5 and therefore

$$V_r = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

V_r is in fact orthogonal to the reachability space and hence (2.23) and (2.24) can

never capture the original model behavior. This example has also indirectly shown us the importance of the "right hand side" b and will be incorporated shortly.

2.6.3 Arnoldi based Methods

[13, 15] are examples of the several publications that have used and developed Arnoldi based model reduction methods. Here we specifically use the approach described by [13].

No Damping Case

The structural dynamics can be described by the O.D.E.

$$\begin{aligned} M\ddot{u} + Ku &= bv \\ y &= c^T u \\ Y(s) &= \underbrace{c^T(s^2M + K)^{-1}b}_{H(s)}\ U(s) \end{aligned} \tag{2.25}$$

where $M \in R^{N\times N}$ is the mass matrix, $K \in R^{N\times N}$ the stiffness matrix, $b \in R^N$ the input force direction and $v \in C$ is the input. Expanding the transfer function in the Taylor series around $s = 0$, the coefficients of the powers of s are

$$c^T(K^{-1}M)^{i-1}K^{-1}b,\ i = 0 \ldots \infty$$

Defining a transformation matrix V containing the Krylov subspace as

$$V \supset < K^{-1}b, (K^{-1}M)K^{-1}b \ldots (K^{-1}M)^{i-1}K^{-1}b > \tag{2.26}$$

and the transformation as

$$\begin{aligned} &u = Vz,\ \tilde{M} = V^TMV,\ \tilde{K} = V^TKV, \\ &\tilde{b} = Vb,\ \tilde{c} = Vc \end{aligned} \tag{2.27}$$

The reduced O.D.E. is

$$\begin{aligned} \tilde{M}\,\ddot{z} + \tilde{K} z &= \tilde{b} v \\ y &= \tilde{c}^T z \end{aligned} \tag{2.28}$$

If V is defined as above, the reduced system is guaranteed to match i nonzero moments of the original system (see [13] for proof).

Damping Case

In the case of damping, the O.D.E. becomes

$$\begin{aligned} & M\ddot{u} + D\dot{u} + Ku = bv \\ & y = c^T u \\ & Y(s) = \underbrace{c^T(s^2M + sD + K)^{-1}b}_{H(s)}\ U(s) \end{aligned} \tag{2.29}$$

where D is the damping matrix. In this case it is not straightforward to have a similar Taylor expansion. Instead we can equivalently convert the system to first order

$$\begin{aligned} & \hat{M} = \begin{bmatrix} I & \\ & M \end{bmatrix},\ \hat{K} = \begin{bmatrix} 0 & -I \\ K & D \end{bmatrix},\ \hat{b} = \begin{bmatrix} 0 \\ b \end{bmatrix} \\ & \hat{M}\,\dot{z} + \hat{K} z = \hat{b} v \end{aligned} \tag{2.30}$$

If as before we define V as

$$V \supset < \hat{K}^{-1}\hat{b}, (\hat{K}^{-1}\hat{M})\hat{K}^{-1}\hat{b} \ldots (\hat{K}^{-1}\hat{M})^{i-1}\hat{K}^{-1}\hat{b} >$$

although we can obtain reduced matrices they are not guaranteed to be stable [4] as in the no-damping case (if the original system is stable). Instead [13] has shown if we take only the top half,

$$\hat{V}^1,\ \hat{V} = \begin{bmatrix} \hat{V}^1 \\ \hat{V}^2 \end{bmatrix}$$

[4] If M, K and D are all symmetric positive definite then the system is stable

this matches exactly i nonzero moments of the unreduced damped system. But the intuition is that if $x = V_1 z$, taking the time derivative $\dot{x} = V_1 \dot{z}$ and therefore we shouldn't really expect to have a separate relation i.e. $\dot{x} = V_2 \dot{z}$. In fact $V_1 \supset V_2$.

and now the reduced matrices are defined as

$$\begin{aligned} &\tilde{u} = \hat{V}^1 u,\ \tilde{M} = (\hat{V}^1)^T M \hat{V}^1,\ \tilde{K} = (\hat{V}^1)^T K \hat{V}^1, \\ &\tilde{D} = (\hat{V}^1)^T D \hat{V}^1,\ \tilde{b} = \hat{V}^1 b,\ \tilde{c} = \hat{V}^1 c \end{aligned} \tag{2.31}$$

and the reduced O.D.E. is

$$\begin{aligned} &\tilde{M}\,\ddot{\tilde{u}} + \tilde{D}\,\dot{\tilde{u}} + \tilde{K}\tilde{u} = \tilde{b}v \\ &y = \tilde{c}^T \tilde{u} \end{aligned} \tag{2.32}$$

Note that even though "K^{-1}" appears twice in

$$\hat{K}^{-1} = \begin{bmatrix} K^{-1}D & K^{-1} \\ -I & 0 \end{bmatrix}$$

K^{-1} needs to applied only once to compute "$\hat{K}^{-1}\times$ a vector". One can also alternatively expand around $s = \infty$ by making the substitution $\acute{s} \rightarrow \frac{1}{s}$. The reduced model obtained through this way will match (2.25) near $t = 0$ (or $\acute{s} = 0$ [5]) as opposed to the $s = 0$ reduced model described here which matches the steady state of (2.25) (because the steady state corresponds to $s = 0$ or the first moments have to match).

[5]Recall the Laplace transform of t is $\frac{1}{s} = \acute{s}$

Chapter 3

Coupled Domain Simulation

3.1 Introduction

The coupled domain problem in this thesis consists of the electrostatic and mechanical domains. After examining the relaxation and Newton based alternative techniques, the multi-level Newton method for coupled domain problems (quasistatic case) will be described along with its efficiency improvements and implementation, particularly, the issue of tolerances. A cantilever beam is then simulated.

3.2 Self-Consistent Solution

Self-consistent electromechanical analysis of micromachined polysilicon devices typically involves determining mechanical displacements which balance elastic forces in the polysilicon with electrostatic pressure forces on the polysilicon surface. Since the surface force on the structure depends on the charge the discretized form of (2.14) can be viewed as

$$F(u) - P(u, q) = 0. \tag{3.1}$$

where u is a vector of finite-element node displacements, F relates node displacements to stresses, and P is the force produced by the vector representing the discretized surface charge q. Note that as the structure deforms, the pressure changes direction

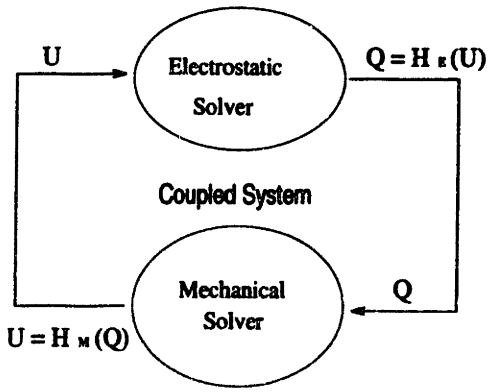


Figure 3-1: Solution by relaxation

and magnitude (since charge density depends on the geometry given a fixed amount of charge for a surface, this will become clearer later), so P is also a function of u.

3.3 Coupled-Domain Techniques

One can view the mechanical analysis as a "black box" which takes an input, q, and produces an output u as in

$$u = H_M(q). \tag{3.2}$$

Similarly the electrostatic analysis is viewed as a "black box" which takes, as input, geometric displacements, u, and produces, as output, a vector of discretized surface charges, q, as in

$$q = H_E(u). \tag{3.3}$$

Self-consistent analysis is then find u and q which satisfy both (3.2) and (3.3) simultaneously.

3.3.1 Relaxation

One can imagine a simple relaxation approach to determine a self-consistent solution to (3.2) and (3.3), by successively using (3.2) to update displacements and then using (3.3) to update charge (Figure 3-1). Applying (3.2) implies solving the nonlinear equation, (3.1), typically using Newton's method.

Algorithm 1 - Nonlinear block Gauss - Seidel relaxation

$k = 0, u_0 = 0$

repeat

compute $q_k = H_E(u_k)$

compute $u_{k+1} = H_M(q_k)$

$k = k + 1$

until $||u_k - u_{k-1}|| < \xi_1$ and $||q_k - q_{k-1}|| < \xi_2$

First we note that (3.3) is well-defined i.e. given a physically "reasonable" u, getting q involves solving a linear system, then from [16](§ 10.3.4) and (3.1) we have the statement that if $\frac{\partial(F-P)}{\partial u}$ and $\frac{\partial P}{\partial q}\frac{\partial q}{\partial u}$ are continuous on an open neighborhood $S_0 \subset D$ of a point $u_* \in D$ where $F(u_*) - P(u_*, H_E(u_*)) = 0$ and $\frac{\partial(F-P)}{\partial u}$ is nonsingular and the spectral radius[1],

$$\varrho([\frac{\partial(F-P)}{\partial u}]^{-1}\frac{\partial P}{\partial q}\frac{\partial q}{\partial u}) < 1$$

then there is an open ball $S \subset S_0$ containing x_* such that the iteration,

$$u_{k+1} = H_M \circ H_E[u_k]$$

[2] is well-defined for any $u_0 \in S$ and the iteration converges at least linearly to u_*. Also note that while solving $u_{k+1} = H_M(q_k)$, the initial guess while solving with Newton's method is u_k.

Apart from the likely slow convergence, the problem with this method is that it may not converge near pull-in [19] - one can imagine it "overshooting" near pull-in. Figure 3-2 shows a spring-capacitor system where the plates come closer in response to an applied voltage. If the applied voltage exceeds V_p then pull-in occurs. Figure 3-3 shows a typical linear looking convergence for the relaxation method and Figure 3-4 shows that even though the equilibrium position lies just inside the pull-in point the

[1] ϱ,the maximum absolute eigenvalue

[2] Note solving (3.2) is $\Leftrightarrow$ to solving (3.1)

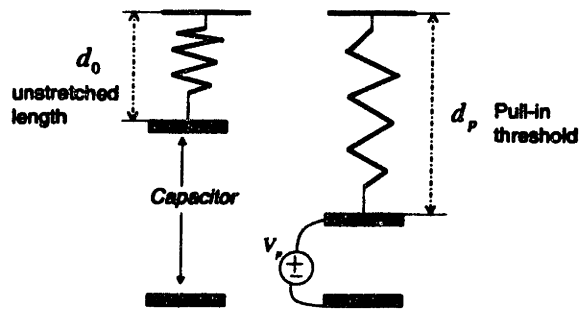


Figure 3-2: Parallel plate capacitor - spring system

plates can still get pulled in.

3.3.2 Explicit Newton

[17] described a technique for getting a self-consistent solution based on Newton's method essentially applied to (3.1). The Newton step for (3.1) is

$$u_{k+1} = u_k - [\frac{\partial(F-P)}{\partial u} - \frac{\partial P}{\partial q}\frac{\partial q}{\partial u}]_k^{-1}[F(u_k) - P(u_k, q_k)] \tag{3.4}$$

where $A(u_k)\phi = q_k$ from 2.21. [17] explicitly calculates $\frac{\partial P}{\partial q}\frac{\partial q}{\partial u}$ equivalently as described below. The electrostatic relation is equivalently cast as

$$\underbrace{G_1(u)}_{system\ matrix}\ \phi = \underbrace{G_2(u)}_{system\ matrix}\ flux$$

and in order to calculate $\frac{\partial P}{\partial q}\frac{\partial q}{\partial u}$, $\frac{\partial flux}{\partial u}$ is needed. Therefore

$$\frac{\partial flux}{\partial u} = G_2(u)^{-1}(\frac{\partial G_1(u)}{\partial u}\phi - \frac{\partial G_2(u)}{\partial u}flux)$$

Now while [17] calculates $(\frac{\partial G_1(u)}{\partial u}\phi - \frac{\partial G_2(u)}{\partial u}flux)$ in $O(N)$ time (justifying by multipole expansions and the fact that $\frac{\partial G_1(u)}{\partial u}$ and $\frac{\partial G_2(u)}{\partial u}$ are sparse) where N is the number

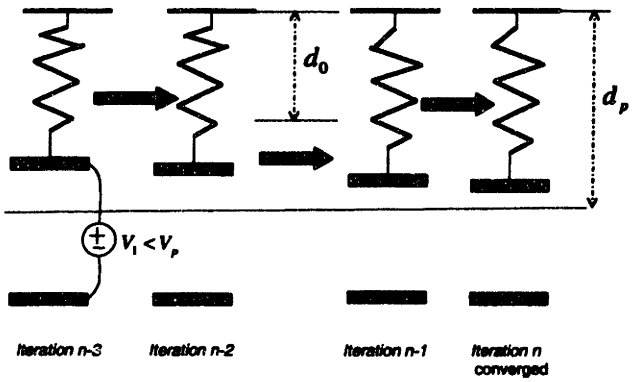


Figure 3-3: Solution by relaxation

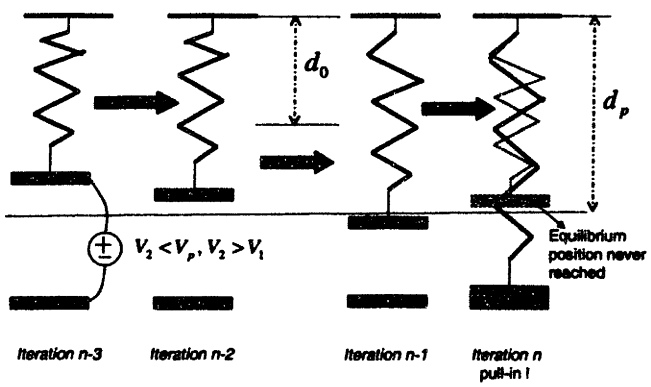


Figure 3-4: Solution by relaxation - pullin

of the panels, applying $G_2(u)^{-1}$ costs $O(N^2)$ overall. which means that this operation is very expensive as it costs at best $O(N^2)$ to explicitly calculate $\frac{\partial P}{\partial q}\frac{\partial q}{\partial u}$ and the cost of calculating for each each Newton step adds up.

3.3.3 Surface-Newton Generalized-Conjugate Residual Algorithm

In order to avoid solving for the large number of internal degrees of freedom at each Newton step [18] proposed solving only for the surface degrees of freedom, u^s, as

$$u^s = H^s_M \circ H_E[u^s] \tag{3.5}$$

with Newton step

$$u^s_{k+1} = u^s_k - [\frac{\partial(u^s - H^s_M \circ H_E[u^s])}{\partial u^s}]^{-1}(u^s - H^s_M \circ H_E[u^s]) \tag{3.6}$$

where

$$H^s_M(q) = Surf[H_M(q)]$$

with $Surf$ extracting u^s from u [3]. Then the GCR [4] algorithm is used to solve the linear system of (3.6). Only the matrix vector product is needed and "$H^s_M \circ H_E[u^s] \times$ a vector" is approximated using finite differences. Since $H^s_M \circ H_E[u^s]$ is never explicitly formed this is called a **matrix-implicit** method. After the equilibrium surface displacements, u^s_* are found then the two solvers are decoupled and solved separately to obtain q_* and u_* (this includes the internal degrees of freedom). [18] has shown that this algorithm converges both faster and more robustly than the relaxation based method. However, the major problem of this approach is that the solution is only an approximation because in the final step when the internal displacements are also computed, the force balance equations on the surface are *not* satisfied anymore(with internal forces also included) and therefore this method works best when the surface

[3] Also note that H_E really depends only on the surface variables u^s.
[4] Another Krylov subspace based method

nodes are weakly coupled to the interior nodes such as in a very elastic structure.

3.3.4 Multi-level Newton

Instead of a direct solution (3.4) we would like to use the Newton-Krylov iterative method to solve the coupled system. Therefore the inverse in (3.4) can be applied iteratively or equivalently we can solve the larger system

$$\begin{aligned} q - A^{-1}(u)\phi &= 0 \\ F(u) - P(u,q) &= 0 \end{aligned} \tag{3.7}$$

with the Newton step

$$\begin{bmatrix} I & -\frac{\partial(A^{-1})}{\partial u}\phi \\ -\frac{\partial P}{\partial q} & \frac{\partial(F-P)}{\partial u} \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} q - A^{-1}(u)\phi \\ F(u) - P(u,q) \end{bmatrix} \tag{3.8}$$

Note that after elimination of Δq from (3.8), the expression is reduced to

$$u_{k+1} = u_k + [\frac{\partial(F-P)}{\partial u} - \frac{\partial P}{\partial q}\frac{\partial q}{\partial u}]_k^{-1}[F(u_k) - P(u_k, q_k) + \frac{\partial P}{\partial q}(q - A^{-1}(u)\phi)]$$

i.e. the RHS here has an extra term over the RHS of (3.4) and hence we can expect different convergence characteristics despite having the same matrix-vector multiply costs when solving (3.4) and (3.8) iteratively. The form of either equation, however, is not very attractive as the condition number may be too large to apply iterative methods. Instead we would like to apply the Newton-Krylov method to the system formulated as black boxes (3.2) and (3.3).

$$\begin{bmatrix} q \\ u \end{bmatrix} - \begin{bmatrix} H_E(u) \\ H_M(q) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.9}$$

in which case the updates to charge and displacement are given by solving

$$\begin{bmatrix} I & -\frac{\partial H_E}{\partial u} \\ -\frac{\partial H_M}{\partial q} & I \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} q - H_E \\ u - H_M \end{bmatrix} \tag{3.10}$$

The above method is referred to as a multi-level Newton method [19] because forming the right-hand side in (3.10) involves using Newton's method to compute H_M. The (3.10) and (3.8) formulations will be compared later in section 3.5.

The Jacobian appears to be well conditioned because of the diagonal identity blocks suggesting an iterative solution method. Here, the GMRES method is used. [5] However, note that the magnitude scales of q and u in the usual units of *Coulombs* and *microns* respectively are not only quite different but the black box natures could also adversely affect the conditioning. In practice, however, the number of GMRES iterations have rarely exceeded 20.

For an iterative solver, an explicit representation of the matrix is *not* required, only the ability to perform matrix-vector products. From (3.10), it is clear that to compute these products one need only compute $\frac{\partial H_M}{\partial q} d_1$ and $\frac{\partial H_E}{\partial u} d_2$. These products can be approximated by finite differences as in

$$\frac{\partial H_M}{\partial q} d_1 \approx \frac{H_M(q + \xi d_1) - H_M(q)}{\xi} \tag{3.11}$$

where ξ is a very small number. Therefore, this matrix-implicit multilevel-Newton method can treat the individual solvers as black boxes. The black box solvers are called once in the outer Newton loop to compute the right hand side in (3.10) and then called once per each GMRES iteration. Computing $H_M(q+\xi d_1)$ means using an inner loop Newton method to solve (3.1), which is expensive. Also varying ξ affects the overall algorithm convergence properties. Therefore a scheme is desirable which is not a function of a user specific factor such as ξ. An advantage of this method, however, is that it is not necessary to modify either the mechanical or electrostatic

[5] The nomenclature is "Outer Newton" for the coupled Newton, "Outer GMRES" for the iterative solution of linear system of the coupled Newton, "Inner Newton" for H_M and "Inner GMRES" for H_E

analysis programs.

Once the mechanical solver is called q is "fixed" till the solver returns u. As the structure deforms while the solver computes u the charge pressure varies in magnitude and direction since the surface varies (more on the map between charges and pressures in "Implementation").

3.4 Modified Multi-level Newton

In order to improve the efficiency of the multilevel-Newton method, one would like to avoid solving (3.1) on every GMRES call. Instead, it is possible to modify the finite-element mechanical solver so that perturbations in displacements due to perturbations in charge can be directly computed. To see this, note that

$$\{(u,q) : u = H_M(q)\} \equiv \{(u,q) : F(u) = P(u,q)\} \tag{3.12}$$

The black box equation is an implicit form and (3.1) is explicit. Then

$$\frac{\partial H_M}{\partial q} = (\frac{\partial (F-P)}{\partial u})^{-1} \frac{\partial P}{\partial q} \tag{3.13}$$

That the black box form exists and (3.13) is true is confirmed with the Implicit Function theorem which states that given $(\frac{\partial (F-P)}{\partial u})$ is full rank, this implies that locally a *unique* map from q to u exists (similar argument as in section 3.3.1).

Given equation (3.13), if the charge is perturbed by δq, and the corresponding perturbation to the displacements δu is given by

$$\delta u = \frac{\partial H_M}{\partial q} \delta q \tag{3.14}$$

$$\Leftrightarrow \frac{\partial (F-P)}{\partial u} \delta u = \frac{\partial P}{\partial q} \delta q \tag{3.15}$$

Using (3.15) to compute $\frac{\partial H_M}{\partial q} \delta q$ is very efficient because $\frac{\partial (F-P)}{\partial u}$ will have already been constructed and factored when computing the Newton right side. This reduces the cost per GMRES iteration from a nonlinear solve to a single backsolve and a mul-

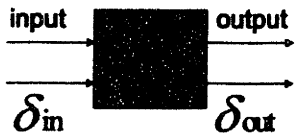


Figure 3-5: Ideal Black Box

tiplication by $\frac{\partial P}{\partial q}$. The multiplication by $\frac{\partial P}{\partial q}$ is inexpensive because $\frac{\partial P}{\partial q}$ is very sparse, only local pressures are effected by local charges (a projected pressure at a node of a rectilinear structure for example is affected by a maximum of 40 charges only) as determined by the map from q to P (more about this map in "Implementation").

The semi-matrix implicit GMRES described above eliminates the nonlinear mechanical black box calls from inside GMRES and leads to better overall outer Newton performance and as much as an order of magnitude in savings in CPU time[6]. We note that the bottleneck inside GMRES has shifted to the electrostatic black box call per GMRES iteration. Therefore an efficient and accurate computation of the electrostatic sensitivity to the geometry needs to be implemented [22].

This brings us to a different paradigm for solving black box systems. In addition to returning an output for an input, the black box should now return a perturbation in output for a perturbation in input (Figure 3-5) to make multilevel Newton competitive with custom coupled solvers. The multi-level Newton algorithm is summarized as shown below

Algorithm 2 - Multi-level Newton

$k = 0, u_0 = 0$

$q_0 = H_E(u_0)$

repeat

solve with GMRES the linear system below for Δq_k, Δu_k

[6]Henceforth, "Multi-level Newton will refer to "Modified Multi-level Newton"

$$\begin{bmatrix} I & -\frac{\partial H_E}{\partial u}|_{q_k,u_k} \\ -(\frac{\partial(F-P)}{\partial u})^{-1}\frac{\partial P}{\partial q}|_{q_k,u_k} & I \end{bmatrix} \begin{bmatrix} \Delta q_k \\ \Delta u_k \end{bmatrix} = - \begin{bmatrix} q_k - H_E(u_k) \\ u_k - H_M(q_k) \end{bmatrix}$$

(here $\frac{\partial H_E}{\partial u} d_1 \approx \frac{H_E(u+\xi d_1)-H_E(u)}{\xi}$ where ξ is a very small number)

$u_{k+1} = u_k + \Delta u_k$, $q_{k+1} = q_k + \Delta q_k$

$k = k + 1$

until $||u_k - u_{k-1}|| < \xi_1$ and $||q_k - q_{k-1}|| < \xi_2$

3.5 Direct Newton and "Black-Box" Newton compared

If we left precondition the linear system of (3.8) with

$$\begin{bmatrix} I & 0 \\ 0 & (\frac{\partial(F-P)}{\partial u})^{-1} \end{bmatrix}$$

then (3.8) becomes

$$\begin{bmatrix} I & -\frac{\partial(A^{-1})}{\partial u}\phi \\ -(\frac{\partial(F-P)}{\partial u})^{-1}\frac{\partial P}{\partial q} & I \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} q - A^{-1}(u)\phi \\ (\frac{\partial(F-P)}{\partial u})^{-1}(F(u) - P(u,q)) \end{bmatrix} \quad (3.16)$$

Note that the left hand side matrix of (3.16) is exactly the same as the Jacobian in (3.10) taking (3.13) into account. However the RHS's are obviously different and the LHS matrix of (3.16) is clearly not the Jacobian of the RHS of (3.16). While their convergence behaviors will indeed be different, in some sense (3.10) is a "preconditioned" version of (3.16).

3.6 Preconditioning

The bottleneck is mostly the electrostatic solver while simulating the coupled system which gets called several times during the outer iterative solve. For an alternative to a fast sensitivity calculation [22], to compute q more rapidly we can think of for ex-

Stage	Number of Iterations
Before applying Q	68
After applying Q	72

Table 3.1: Effect of Arnoldi Preconditioner

ample inverting the direct interactions to give us a preconditioner for the electrostatic GMRES. Specifically a matrix is constructed by just considering the direct interactions and then inverted. Then the matrix entries with rows corresponding to each cell k and columns corresponding to cells which are not neighbors of k are zeroed out to give a local inverse based preconditioner. Another approach would be to recycle the Krylov-subspace from a GMRES solve. During the outer iteration loop several black box calls are made with slightly perturbed geometries

$$H_E(u+\delta u_1), H_E(u+\delta u_2), H_E(u+\delta u_3), \ldots \tag{3.17}$$

So when the RHS is computed $H_E(u)$ is computed and A can thought to be approximated as

$$\bar{A} = QHQ^T + I - QQ^T \tag{3.18}$$

where H and Q are the truncated Hessenberg and Arnoldi matrices from the $H_E(u)$ call (Note : $H = Q^T A Q$) . $\bar{A}^{-1}$ is easy to compute as

$$\bar{A}^{-1} = QH^{-1}Q^T + I - QQ^T \tag{3.19}$$

The term $I - QQ^T$ is needed because without it $\bar{A}$ is singular and this term represents the orthogonal projector onto the orthogonal complement of Q hence removing the singularity. This idea was previously used in the context of restarted GMRES [23]. Unfortunately in practice this preconditioner has shown little or even negative impact - Table 3.1 - on the number of iterations (and may become expensive to apply overall if the dimension of H is large in the first place).

This probably happens because when $dim(Q,1) >> dim(Q,2)$, Q^T is a poor approximation of the inverse of Q. This suggests having two stages of preconditioners with the Arnoldi based preconditioner described here following the first stage. The first preconditioner could bring down the number of iterations to a small number and then the Arnoldi preconditioner preconditioning this preconditioned system could bring down the number of iterations even more.

3.7 Implementation

The electrostatic domain is discretized into surface panels and the mechanical domain into 20 noded brick elements fitted with parabolic basis functions - Each node has a parabolic basis function in each direction (X direction for instance) which is unity valued at that node and zero valued at all other nodes with different X coordinates(two of them). The product of these individual direction specific parabolic basis functions (set to zero beyond the element domain) constitutes the basis function for the node. In this thesis the linearized system was solved using a sparse Gauss elimination based solver since the finite element basis has a support of only a maximum of 20 nodes over each contributing element leading to a typically sparse system. A face of a elastic/rigid brick element is typically broken up into 8 panels, each panel sharing a common central node whose coordinates are the averages of the 8 edge coordinates. The surface pressure on a face is calculated by taking the average of the 8 panel pressures (Figure 3-6).

$$P_f = \frac{10^{-6}}{2 * \varepsilon_0} \frac{\Sigma \frac{q_i^2}{A_i}}{\Sigma A_i} \tag{3.20}$$

where $i = 1...8$, A_i is the area of the ith panel.

To determine $\frac{\partial P}{\partial q}$ for a face, the face equivalent nodal pressures are calculated for a unit pressure on the face and are then individually scaled by each of the 8 charge factors of (3.20). Note that an alternative definition of P_f in terms of the pressure

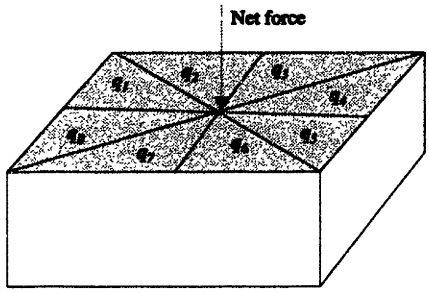


Figure 3-6: Charge panels on brick surface. Individual pressures collected into a net force and turned into an uniform pressure

force due to average charge is also possible but leads to a slight implementation inconvenience (because of extra cross-terms such as $q_1 q_2$ etc.) in the context of generating reduced order models (See Chapter 6).

3.7.1 Tolerances

In Chapter 2 we noted that in the Newton-Krylov method if the iterative method is stopped when the residual norm is roughly equal to a tolerance times the RHS norm squared then the Newton method can converge quadratically locally under certain assumptions. The issue is more complicated here because to compute the matrix-vector product and RHS we call the Krylov subspace based iterative electrostatic black box for which a tolerance needs to be set and the Newton method based mechanical solver which also requires a tolerance setting. Yet another user input as mentioned before is the finite difference factor associated with calculating the charge sensitivity. Note that if the mechanical solution is not computed accurately then the displacement sensitivity calculation is not exact and will affect the outer GMRES computation. Following the notation of Chapter 2 $f(x_k)$ is not computed exactly. Let the approximation be

$$f(x_k) \rightarrow f(x_k) - g_{x_k} \tag{3.21}$$

where g_{x_k} is the error in when x_k is the current state.

Also

$$Df(x_k)(x_{k+1}-x_k) \rightarrow \frac{f(x_k+\xi(x_{k+1}-x_k))-f(x_k)}{\xi} - \frac{g_{x_k+\xi(x_{k+1}-x_k)}-g_{x_k}}{\xi} \quad (3.22)$$

Here we assume that $Df(x_k)(x_{k+1}-x_k) = \frac{f(x_k+\xi(x_{k+1}-x_k))-f(x_k)}{\xi}$ i.e. the finite difference approximation would equal the actual Jacobian $\times$ a vector if $f(x_k)$ was computed accurately. $g_{x_k+\alpha(x_{k+1}-x_k)}$ corresponds to the residual of computing the last matrix vector product after which we stop the iteration. From (2.8), (3.21) and (3.22)

$$Df(x_k)(x_{k+1}-x_k)+f(x_k)-\underbrace{\frac{g_{x_k+\xi(x_{k+1}-x_k)}-g_{x_k}}{\xi}-g_{x_k}-r_k}_{\varpi_k}=0 \quad (3.23)$$

ϖ_k is the new residual and the analysis of Chapter 2 holds as before. If we guarantee

$$||r_k||/||f(x_k)||^2 \leq \frac{1}{3}\alpha \quad (3.24)$$

$$\frac{1}{\xi}||g_{x_k+\xi(x_{k+1}-x_k)}||/||f(x_k)||^2 \leq \frac{1}{3}\alpha \quad (3.25)$$

$$|1-\frac{1}{\xi}|||g_{x_k}||/||f(x_k)||^2 \leq \frac{1}{3}\alpha \quad (3.26)$$

then $||\varpi_k||/||f(x_k)||^2 \leq \alpha$. However instead of $||f(x_k)||$ only $||f(x_k)-g_{x_k}||$ is known. This and the fact that for the electrostatic solver (2.4) will not hold in general makes it difficult to obtain a convergence criterion to satisfy (3.26) since "$f(x)$" is equal to (3.9) for our problem here.

Instead we simply present numerical results. All tolerances are relative.

In Table 3.2 for the lower tolerance case the convergence moves away from being quadratic, and takes an extra Newton Step with slightly more average GMRES iterations per Newton step.

From Table 3.3 and the upper part of Table 3.2 representing the same stages of the computation, it appears that despite a slight loss of superlinear convergence a GMRES

Elec. tolerance = Mech. tol	No. Outer GMRES iters. (tol. = 1e-5)	Outer Newton iterates	Outer Newton RHS
1e-5	8	1.163701e+00	1.652927e-01
	9	8.201623e-02	2.069343e-02
	7	4.181924e-04	6.191509e-04
	9	3.240167e-07	3.211859e-07
1e-3	8	1.159801e+00	1.646205e-01
	9	8.197650e-02	2.044600e-02
	9	1.446459e-03	7.406916e-04
	9	6.711714e-05	3.701086e-05
	7	7.456214e-07	6.361503e-07

Table 3.2: Effect of varying tolerances for inner GMRES and inner Newton

Elec. tolerance = Mech. tol	No. Outer GMRES iters. (tol. = 1e-3)	Outer Newton iterates	Outer Newton RHS
1e-5	4	1.163775e+00	1.652927e-01
	5	8.192913e-02	2.068255e-02
	5	4.081401e-04	6.194835e-04
	5	1.719611e-06	8.905395e-07

Table 3.3: Effect of outer GMRES tolerance

tolerance of 10^{-3} is actually significantly better than a tolerance of 10^{-5} because of the total number of matrix-vector products i.e. 19 compared to 33, provided of course the iteration converges to the right solution.

Finally it should be pointed out that due to finite precision round off errors at various stages of the computation, unfortunately the tolerances cannot be set too low and a small absolute tolerance (varying between 10^{-8} to 10^{-10} for this thesis [7]) is incorporated as a result. The overall effective relative tolerances have been found to be 10^{-5} for H_E, H_M and inner GMRES and 10^{-4} for outer Newton. Also the parameter ξ cannot be set either large or too small to avoid approximation inaccuracy and finite precision roundoff errors respectively.

[7] determined experimentally

Structure	Matrix free CPU time / Semi−matrix free
100 elements cantilever	3.81
100 elements simply supported beam	7.77
196 elements accelerometer	3.48

Table 3.4: Matrix-implicit and Semi-matrix-implicit methods compared

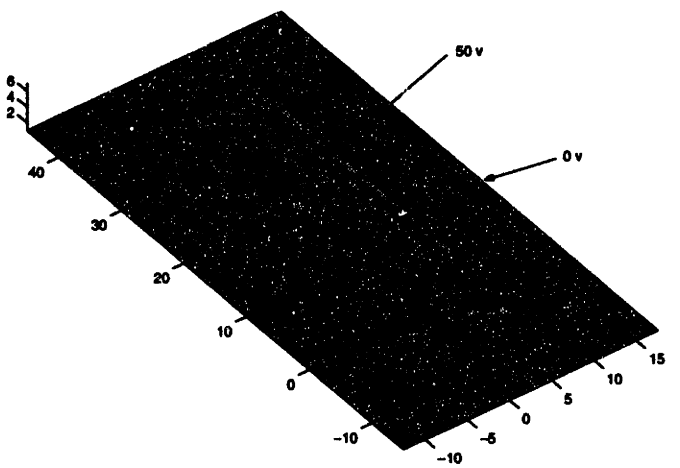


Figure 3-7: Beam over ground

3.8 Results

In this section we present results from our multilevel-Newton coupled electromechanical code. First we compare the matrix-implicit method and the semi-matrix-implicit method in Table 3.4.

The semi-matrix-implicit method is faster not only because the charge sensitivity is calculated faster but also because this accurate computation leads to fewer outer GMRES calls than the matrix-implicit method which calculates the sensitivity approximately.

Figure 3-7 shows a cantilever beam over a ground plane and Figure 3-8 is the self-consistent solution.

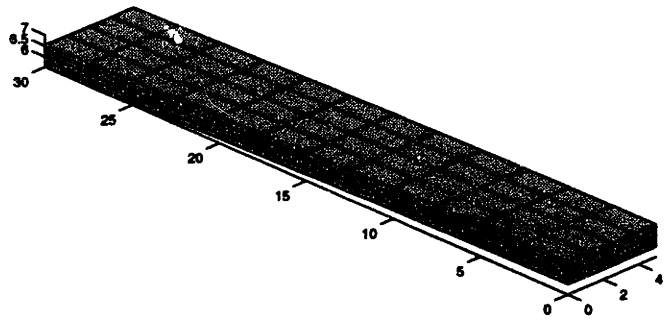


Figure 3-8: Deflected beam

Chapter 4

Mixed Regime Simulation - Quasistatics

4.1 Introduction

In the previous section we developed the coupled method which was based on a linear elasticity based mechanical solver. However, if we were to adopt this solver for devices such as the accelerometer or resonator (See Chapter 1) then it would take a long time to simulate because of the large number of internal degrees of freedom generated (In 3-D the cost of factoring the Jacobian roughly goes as m^2 where m is the number of elements). Instead we would like to exploit the fact that in these structures (Figure 4-1) much of the structure behaves like a rigid body and that these internal degrees of freedom behave as if they were constrained. Hence we describe the mixed regime or rigid-elastic formulation (still quasistatic) in the next section followed by a brief implementation description.

4.2 Rigid-Elastic Formulation

As pointed out in the previous section many finite-element degrees of freedom can be eliminated and replaced with a rigid body with only 6 degrees of freedom $u_{rigid} = \{\theta_1, \theta_2, \theta_3, xR, yR, zR\}$. The u in (3.1) is then $u_{elastic} \cup u_{rigid}$.

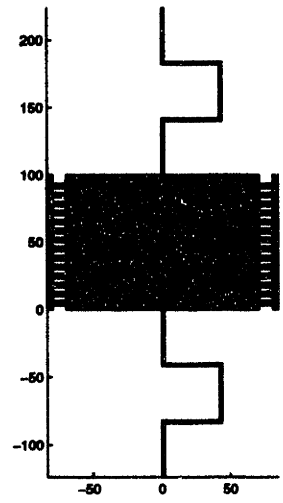


Figure 4-1: Comb drive accelerometer

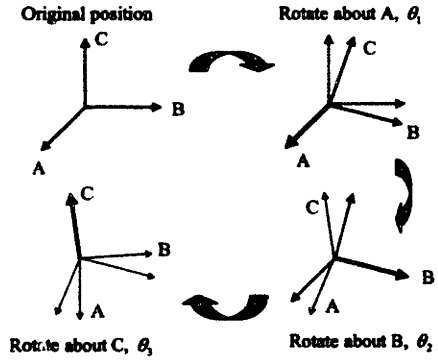


Figure 4-2: Tait-Bryan angles

The current configuration y of a rigid body under displacement is expressed as

$$y = Qx + z \tag{4.1}$$

where x is a point on undisplaced body, Q is a rotation tensor and z a translation vector.

4.2.1 Tait-Bryan Angles Formulation

In terms of Tait-Bryan angles [25] (Figure 4-2)

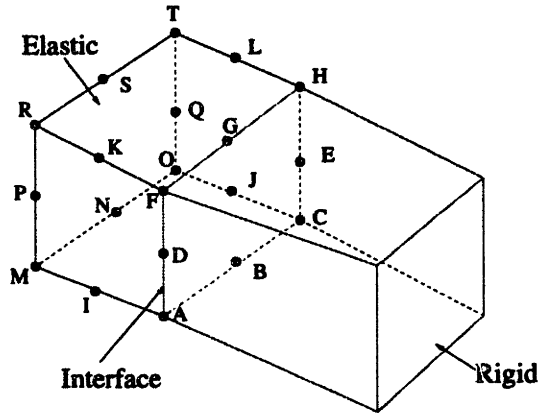


Figure 4-3: Finite Element-Rigid Interface

$$Q = \begin{bmatrix} c\theta_2 c\theta_3 & c\theta_1 s\theta_3 + s\theta_1 s\theta_2 c\theta_3 & s\theta_1 s\theta_3 - c\theta_1 s\theta_2 c\theta_3 \\ -c\theta_2 s\theta_3 & c\theta_1 c\theta_3 - s\theta_1 s\theta_2 s\theta_3 & s\theta_1 c\theta_3 + c\theta_1 s\theta_2 s\theta_3 \\ s\theta_2 & -s\theta_1 c\theta_2 & c\theta_1 c\theta_2 \end{bmatrix} \tag{4.2}$$

where $c = cos$ and $s = sin$.

The equations of equilibrium for a rigid body are

$$F_x^R = F_y^R = F_z^R = M_x^R = M_y^R = M_z^R = 0; \tag{4.3}$$

where F^R is vector of net forces on the body and M^R is the vector of net moments of the body about a selected equilibrium point.

The Jacobian J_F of the F in (3.1) has four distinct parts

$$J_F = \begin{bmatrix} K_{EE} & K_{ER} \\ K_{RE} & K_{RR} \end{bmatrix} \tag{4.4}$$

K_{EE} , the elastic - elastic interaction, is the standard linear elasticity finite-element stiffness matrix, excluding the entries due to the rigid-elastic interface nodes.

K_{ER} is the elastic - rigid interaction. The following relation exists for the equivalent nodal forces at the non constrained nodes $i = I..T$ (Figure 4-3) in terms of the

interface nodes $i = A..H$. For $\xi \in u_{rigid}$

$$\frac{\partial F}{\partial \xi} = \frac{\partial F}{\partial x_A}\frac{\partial x_A}{\partial \xi} + \frac{\partial F}{\partial x_B}\frac{\partial x_B}{\partial \xi} + \cdots + \frac{\partial F}{\partial x_H}\frac{\partial x_H}{\partial \xi} \tag{4.5}$$

Also we know

$$x_i = x_i(u_{rigid}) \tag{4.6}$$

from (4.1).

K_{RE}, the rigid - elastic term, represents the dependence of the forces of the interface on nodes $I..T$ of the elastic element as noted earlier and these forces contribute in the equilibrium of the rigid body. For example the moment M of the rigid body is

$$M = L(F_i^I) \tag{4.7}$$

where L is an operator linear in F_i^I, a interface nodal force component. Then

$$\frac{\partial M}{\partial x} = L(\frac{\partial F_i^I}{\partial x}) \tag{4.8}$$

but $\frac{\partial F_i^I}{\partial x}$ is directly obtained from K_{EE}.

Apart from the external force projections, the only other forces at the interface are the equivalent nodal forces and it is important to realize that these elastic force projections are such that the virtual work on the element due to them equals the virtual work due to element internal stresses and are therefore not an exactly equivalent force system to the interface surface pressure exerted by the elastic element on the rigid element. In other words the rigid body simply mimics the elastic element by imposing element and nodal force equilibrium.

K_{RR} is the pure rigid-rigid interaction term. This exists because external surface forces (eg. charge pressure) on the rigid body vary with u_{rigid} and because the force at an interface node depends on other interface nodes too.

It should be noted that J_F is not symmetric (it is however still structurally symmetric implying benefits for the sparse solver).

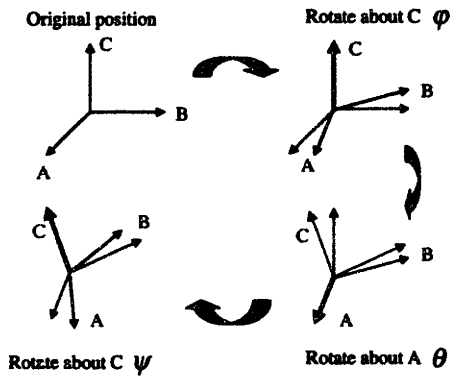


Figure 4-4: Euler angles

4.2.2 Alternative Formulation - Tait-Bryan plus Euler Angles

Another disadvantage is that the TB formulation has a singularity at $\theta_2 = 90°$ (with an infinite number of θ_1 and θ_3 values for the same state). Another choice, an Euler angles based formulation (Figure 4-4) (θ, ϕ, ψ) has a singularity for $\theta = 0°$. At $\theta_2 = 90°$,

$$Q_{TB} = Q_1 = \begin{bmatrix} 0 & \sin(\theta_1 + \theta_3) & -\cos(\theta_1 + \theta_3) \\ 0 & \cos(\theta_1 + \theta_3) & \sin(\theta_1 + \theta_3) \\ 1 & 0 & 0 \end{bmatrix} \tag{4.9}$$

For Euler angles formulation, $\theta = 0$ gives

$$Q_E = Q_2 = \begin{bmatrix} \cos(\phi + \psi) & \sin(\phi + \psi) & 0 \\ -\sin(\phi + \psi) & \cos(\phi + \psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

Comparing Q_1 and Q_2 we see that they can never represent the same rotation. Both represent rotations in terms of 3 angles and therefore lead to the same Jacobian structure. Therefore a scheme is possible which can alternate between these formulations (without reordering the matrix) to avoid singularities.

$\|x_k - x_{k-1}\|$	$\|x_k - x_0\|$	1/Inf. norm condition num.	$\|RHS\|$
1.659364e+02	0.000000e+00	4.579519e-16	1.692242e-04
7.522189e-01	1.659364e+02	2.025080e-15	4.310216e+02
2.117688e+00	1.663536e+02	2.068672e-15	5.192372e+01
6.830594e-01	1.643107e+02	3.023109e-16	7.325284e+00
1.330566e+00	1.649917e+02	5.517946e-16	2.449945e+00
3.149908e-01	1.656975e+02	5.953437e-16	2.561492e+00
1.723040e-01	1.659556e+02	7.218865e-16	9.303514e-01
1.823979e-02	1.658402e+02	1.105089e-15	4.158300e-01
2.510152e-04	1.658378e+02	5.126626e-16	8.560872e-03
4.910356e-07	1.658376e+02	5.126242e-16	6.785718e-04

Table 4.1: Quaternion Formulation Simulation

$\|x_k - x_{k-1}\|$	$\|x_k - x_0\|$	1/Inf. norm condition num.	$\|RHS\|$
1.659566e+02	0.000000e+00	1.155994e-15	1.740751e-04
1.396486e-01	1.659566e+02	6.319913e-15	3.601776e+01
1.052388e-02	1.660733e+02	4.520098e-15	7.480708e+00
2.276828e-01	1.660834e+02	1.212690e-15	5.041451e-03
1.786507e-02	1.658558e+02	1.145242e-15	4.812036e-05
2.666699e-04	1.658379e+02	1.129685e-15	2.969424e-07
2.324695e-08	1.658376e+02	1.129597e-15	1.514945e-09

Table 4.2: Tait-Bryan Formulation Simulation

4.2.3 Alternative Formulation - Quaternion

Alternatively a quaternion (q_1, q_2, q_3, q_4) (a generalization of complex numbers [25]) formulation which is singularity free describing the rotation as

$$Q_{quat} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & q_1q_3 - q_2q_4 \\ 2(q_1q_2 - q_3q_4) & q_2^2 - q_1^2 - q_3^2 + q_4^2 & q_2q_3 + q_1q_4 \\ 2(q_1q_3 + q_2q_4) & q_2q_3 - q_1q_4 & q_3q_3 - q_1^2 - q_2^2 + q_4^2 \end{bmatrix} \tag{4.11}$$

with an additional constraint of

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{4.12}$$

was used. However it was noticed when this formulation was applied to the scanning mirror problem (Tables 4.1 and 4.2) that the Jacobian while not singular, was

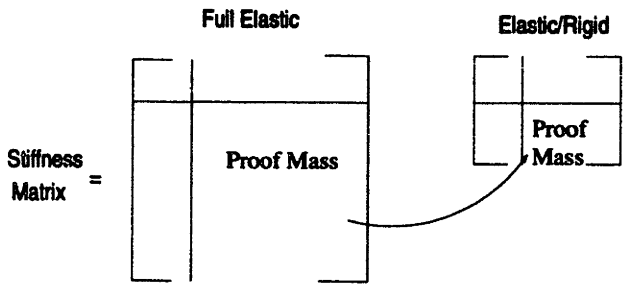


Figure 4-5: Rigid-Elastic matrix reduction

less well conditioned and took more Newton steps than the Tait-Bryan angles scheme in general and hence this scheme was abandoned.

It should also be pointed out the rigid constraints cannot be imposed as a boundary condition on the elastic parts in keeping with the standard finite element formulation. This is because the space of orthogonal tensors is not linear.

The rigid/elastic mechanical solver greatly reduces the size of the stiffness matrix with the bulk shrinking to a dense 6x6 block (Figure 4-5). The surface of the rigid body still has to be discretized finely to properly resolve the electrostatic forces. The rigid/elastic interface should be intruded into the rigid block for a small area (Figure 4-6) around the tether-block mass interface in order to avoid sharp singularities in stress across the interface. Finally since the geometry input to H_E is through the Cartesian coordinates of the surface nodes and not through the rigid body rotation and translation, the coordinates of the rigid body surface are calculated using (4.1) before calling H_E.

4.3 Implementation

An implementation issue is the integration of individual rigid bodies into a single rigid body whenever they behave as a single unit. A necessary input to this process is knowing which elements behave as rigid. For the results in this thesis, the criterion for checking the rigidity is the Young's modulus of the element which is set to a very large value if the element is to be tagged as rigid. Connection to a rigid body in 3-D requires at least three non-collinear common nodes between the rigid body and the

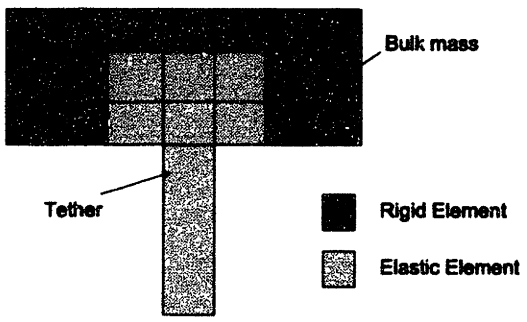
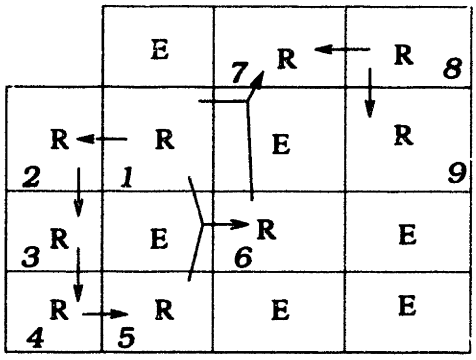


Figure 4-6: Extending elastic domain into rigid body



R - Rigid Element , E - Elastic Element

Figure 4-7: 2-D Rigid body assembly

element. Searching for potential rigid elements to be assembled in a rigid body is done using a depth first search process. If K denotes the number of elements and the $\mathcal{R}$ denotes the set of the rigid elements then the assembly algorithm is written as

$\forall\, i \in K$ check **unexplored**
$\forall\, i = 1$ *to* K
  if *material* $(i) = \infty$ and $i \notin \mathcal{R}$
    $\mathcal{R} = \mathcal{R} \cup i$
    $\forall\, j \in \mathcal{R}$ and j is **unexplored**
      check **explored** for j
      $\forall\, k \in$ *neighbors* (j)
          if *material* $(k) = \infty$ and $k \notin \mathcal{R}$
            $\mathcal{R} = \mathcal{R} \cup k$
          end
      end
    end
  end
end

The 2-D view of the rigid body assembling process starting from element 1 is shown in Figure 4-7. Summation of forces on the rigid body surface is done by

1. finding the centroid of each polygonal surface of a brick exposed to the exterior (the centroid of the polygon is simply the area weighted average of the centroids of the exclusive component triangles).

2. summing the pressure into a force at the centroid (no moment here).

3. shifting the force from the centroid of these individual surface components to the center of mass (with uniform density) of the *rigid structure* creating the same force *plus* a moment there.

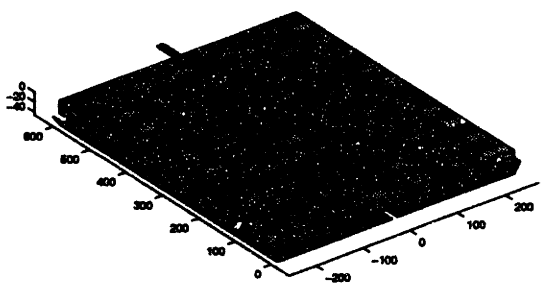


Figure 4-8: Scanning mirror (coarse mesh)

4.4 Results

In this section we present results from our multilevel-Newton coupled electromechanical code.

4.4.1 Scanning Mirror

The solver was tested against the experimental data of a scanning mirror [28] (Figure 4-8) with 12x50x1.1 microns SiN hinges (Young's Modulus = 243.2 GPa, Poisson's Ratio = 0.28) and 500x600x25 microns SiN on Si central plate kept at 0 v. The ground electrodes are kept at $37.5 \pm v$ volts (Figure 4-9). The mirror bulk was discretized into $30 \times 30 \times 3$ blocks (length,width,height) and the hinges were discretized into $3 \times 4 \times 3$ blocks.

The plot (Figure 4-10) shows a close match of the simulation in the linear regime and convergence failure corresponding to pull-in is obtained at 12.13 v (for a side-wall slope of 56 deg) as opposed to 13.4 v (slope was unknown) of the experimental data. Increasing the slope reduces the pull-in voltage but regardless of slope the simulation and experiment match well in the linear regime. On an average each load step took 80 minutes (Digital Alpha 433 MHz). For a coarse mesh the elastic/rigid simulation is compared with the fully elastic simulation (Figure 4-11) to show a close match. The CPU time on a 500MHz Alpha 264DP-2000 for 10 load steps for the fully elastic case was 27.25 hrs as opposed to 34 minutes for the rigid/elastic case.

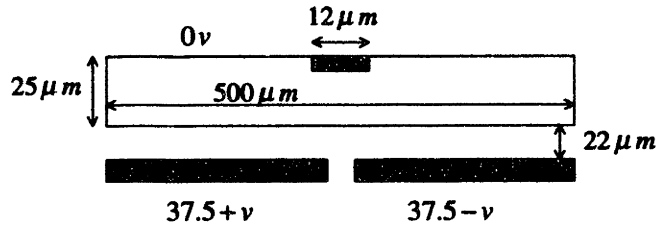


Figure 4-9: Cross-section of scanning mirror

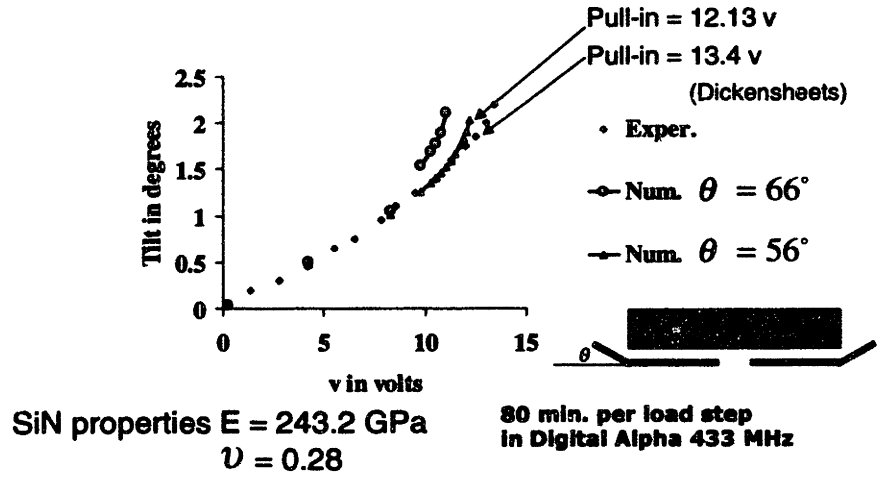


Figure 4-10: Mirror tilt with differential voltage v

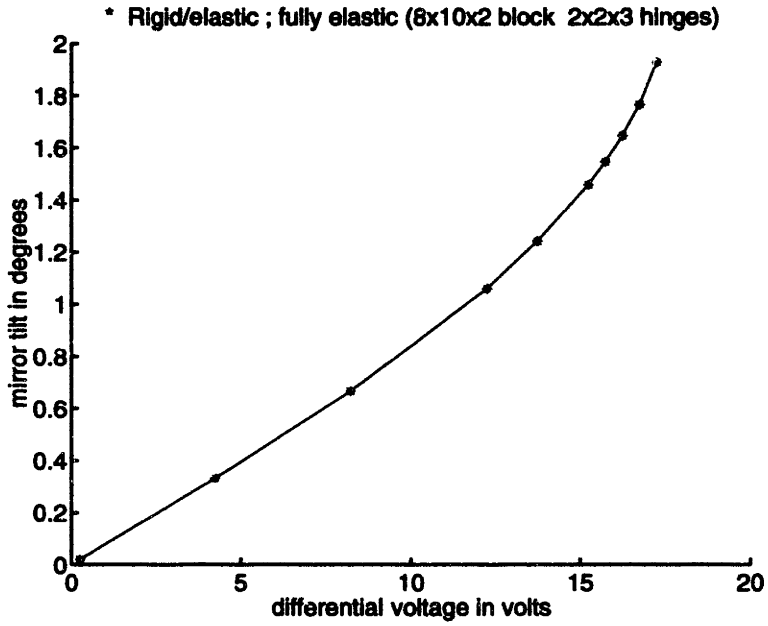


Figure 4-11: Fully elastic and rigid/elastic comparison for coarse mesh block 8x10x2 hinges 2x2x3

4.4.2 Comb Drive Resonator

A 18 finger PolySi resonator [29] (YM = 150 GPa, PR = 0.3) (Figure 4-12) is suspended with 400 microns beams with a uniform depth of 1.94 microns and finger dimensions 13.8x4.6 microns. The movable structure and the ground plane are kept at 0 v and a non zero voltage is applied to the driving electrodes which interdigitate with the movable fingers at the sides. Therefore two sets of same signed charges are created on the resonator and the ground plane which result in a levitating force. Except for the suspension beams everything else is treated as rigid. Varying the width of the suspension beam had almost no effect on the levitation (Figure 4-13) . Each load step in the simulation took about 70 minutes of CPU time (Sun Ultra 30).

The effect of varying the cross-section of the fingers of the resonator and the side supports is shown in Figure 4-17. However before this we must adopt a mesh that is reasonably fine. A convergence study shown in Figure 4-15 is done by varying the number of elements along the fingers (Figure 4-14). Since there is little difference between the cases $n = 12$ and $n = 17$, the $n = 12$ mesh is adopted (the finger corners and edges actually have smaller elements for better accuracy because charge tends

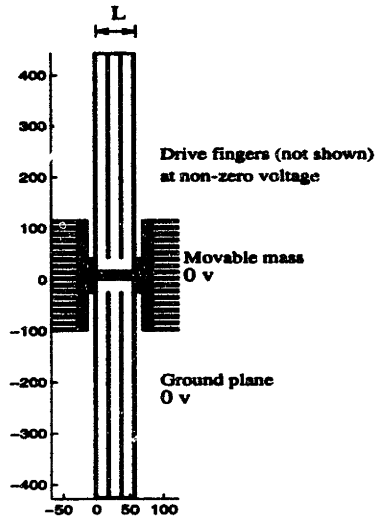


Figure 4-12: Comb drive resonator

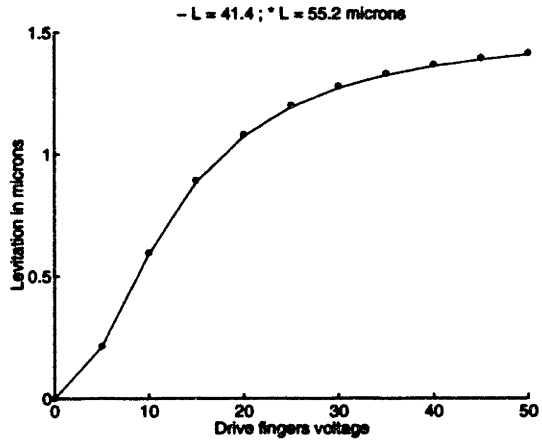


Figure 4-13: Levitation

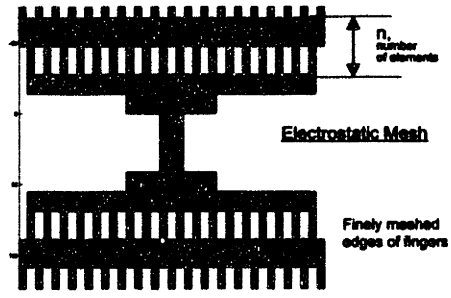


Figure 4-14: Electrostatic Mesh

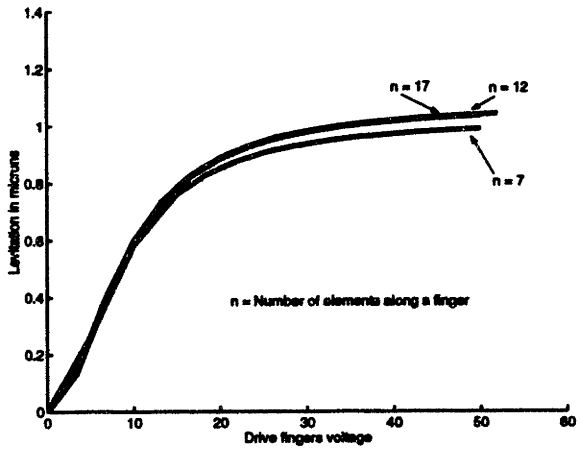


Figure 4-15: Convergence Study

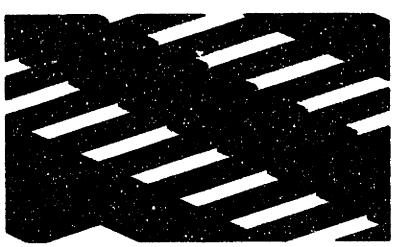


Figure 4-16: Charge distribution - lighter colors represent higher concentration

to accumulate at the corners and edges - Figure 4-16). The curve for $n = 12$ took 33 hrs to simulate on a Sun Ultra 30. The reason why the cross-section might be important is explained by the fact that the electrostatic force always acts outwards (i.e. a pull) and on the walls of the fingers there are vertical components of this force which can either aid or resist levitation depending on the slope. Therefore as the slope changes levitation will range from positive to even negative levitation. This suggests that under-etching might even be a desirable feature since it helps us achieve zero levitation (which is desirable since in-plane horizontal motion is the design requirement).

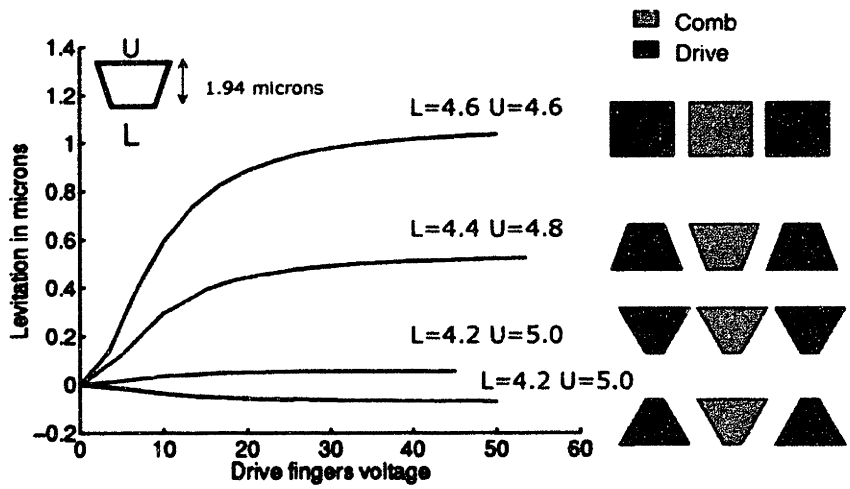


Figure 4-17: Finger cross-section effects on levitation - 36 fingers

Chapter 5

Coupled Domain Simulation - Dynamics

5.1 Introduction

In this chapter we extend the mixed regime solver from the previous chapter and coupled domain simulation from Chapter 3 to the more interesting dynamic structural analysis case and then present results.

5.2 Rigid-elastic Dynamics

The rigid body dynamics in the convected[1] basis [24] is written as

$$\begin{bmatrix} Mass & & \\ & Mass & \\ & & Mass \end{bmatrix} \begin{bmatrix} \ddot{X}_R \\ \ddot{Y}_R \\ \ddot{Z}_R \end{bmatrix} + f = 0 \tag{5.1}$$

for the translation equation where $Mass$ is the rigid body mass and f is the external force(surface force *plus* internal elastic force) on the rigid body and for the rotation equation

[1] "convected" means the rotated basis which is attached to the rigid body and therefore moves with it

$$J\dot{W} + W \times JW + Q^T m = 0 \tag{5.2}$$

where W is the vector of angular velocity, J the inertia tensor, Q the rotation tensor from spatial to body coordinates and m the external moment vector. The convected basis is chosen because J remains constant in this system. In terms of Tait-Bryan angles,

$$W = \begin{bmatrix} c\theta_2 c\theta_3 & s\theta_3 & 0 \\ -c\theta_2 s\theta_3 & c\theta_3 & 0 \\ s\theta_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{5.3}$$

5.2.1 Mass Matrix Displacement Dependent

At the rigid-elastic interface

$$\ddot{u} = \frac{\partial^2 u}{\partial \theta^2}\dot{\theta}^2 + \frac{\partial u}{\partial \theta}\ddot{\theta} \tag{5.4}$$

Lets assume that we have an elastic element with 2 elastic nodes (1&2) and 2 rigidly constrained nodes (3&4). Then the inertial force is given by

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} \ddot{u_1} \\ \ddot{u_2} \\ \ddot{u_3} \\ \ddot{u_4} \end{bmatrix}$$

But $u_3 = u_3(\theta)$ and $u_4 = u_4(\theta)$ where θ represents the vector of rigid body dofs [2]. Therefore the above equation becomes

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} \ddot{u_1} \\ \ddot{u_2} \\ \frac{\partial^2 u_3}{\partial \theta^2}\dot{\theta}^2 + \frac{\partial u_3}{\partial \theta}\ddot{\theta} \\ \frac{\partial^2 u_4}{\partial \theta^2}\dot{\theta}^2 + \frac{\partial u_4}{\partial \theta}\ddot{\theta} \end{bmatrix}$$

[2] "dofs" - degrees of freedom

or for the equilibrium at the elastic nodes,

$$\begin{bmatrix} m_{11} & m_{12} & (m_{13}\frac{\partial u_3}{\partial \theta} + m_{14}\frac{\partial u_4}{\partial \theta}) \\ m_{21} & m_{22} & (m_{23}\frac{\partial u_3}{\partial \theta} + m_{24}\frac{\partial u_4}{\partial \theta}) \end{bmatrix} \begin{bmatrix} \ddot{u_1} \\ \ddot{u_2} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & m_{13} & m_{14} \\ 0 & 0 & m_{23} & m_{24} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{\partial^2 u_3}{\partial \theta^2}\dot{\theta}^2 \\ \frac{\partial^2 u_4}{\partial \theta^2}\dot{\theta}^2 \end{bmatrix}$$

and for the force contribution to f (and also m) for the equilibrium of the rigid body

$$\begin{bmatrix} m_{31} & m_{32} & (m_{33}\frac{\partial u_3}{\partial \theta} + m_{14}\frac{\partial u_4}{\partial \theta}) \\ m_{41} & m_{42} & (m_{43}\frac{\partial u_3}{\partial \theta} + m_{24}\frac{\partial u_4}{\partial \theta}) \end{bmatrix} \begin{bmatrix} \ddot{u_1} \\ \ddot{u_2} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & m_{33} & m_{34} \\ 0 & 0 & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{\partial^2 u_3}{\partial \theta^2}\dot{\theta}^2 \\ \frac{\partial^2 u_4}{\partial \theta^2}\dot{\theta}^2 \end{bmatrix}$$

Therefore the mass matrix of the elastic rows coupled to the interface gets altered in contrast to M in the elastic formulation of (2.19) which is constant and also there is a new velocity term (not damping). Note that

$$\frac{\partial u_3}{\partial \theta}\dot{\theta}^2 \Longleftrightarrow \sum_{i,j} \frac{\partial^2 u_3}{\partial \theta_i \partial \theta_j}\dot{\theta}_i\dot{\theta}_j$$

Additionally when calculating the Jacobian, the derivatives of this term would need to be computed. This certainly complicates the programming. However if the number of internal dofs are much larger than the interface dofs (as is to be expected when using the rigid-elastic formulation) then the cost is negligible. Moreover the Jacobian computation is made easier by the fact that some terms reappear because of the nature of the sines and cosines [3] and more importantly because $\frac{\partial^3}{\partial \theta_i \partial \theta_j \partial \theta_k}$ is the same operator for different permutations of a fixed set of (i, j, k).

[3] i.e. $sin'' = -sin$ etc.

We can write all the equations in standard form

$$M_e(\ddot{u}) + F_e(u) = P_e(u, q) \tag{5.5}$$

$$M_{er}(\ddot{u}, \chi, \theta, \dot{\theta}, \ddot{\theta}) + F_e(u, \chi, \theta) = P_e(u, \chi, \theta, q) \tag{5.6}$$

$$M_r(\theta, \dot{\theta})\ddot{\theta} + F_r(u, \chi, \ddot{u}, \theta) = P_r(\chi, \theta, q) \tag{5.7}$$

where M_e represents the elastic dof - elastic dof mass vector, M_{er} the elastic - rigid dofs mass vector, M_r the rigid body mass vector, and χ and θ the translation and rotation parameters of the rigid body. F_e and P_e represent internal and external forces at an elastic node respectively. F_r and P_r represents non-charge force and charge-force for the rigid body respectively. M_e and M_{er} are linear in u and χ but M_{er} is nonlinearly dependent on θ and its derivatives because of the previous equation. The equations corresponding to the rigid body have a nonlinear mass matrix and also a F_r which depends on the acceleration of the surrounding nodes of the interface because their force contribution is needed for the balance of the rigid body.

J is calculated using the algorithm for polyhedrons described in [27] where the volume integrals are computed by converting them to line integrals.

5.3 Coupled System Dynamics

Most of the interesting performance characteristics in MEMS devices are dynamic. The multilevel scheme in the quasistatic case easily extends over to the dynamics. Charges are still returned by the electrostatic solver as we realistically assume that the structure does not vibrate at a high enough frequency to take into account the current generated due this motion.

Putting $w_t \equiv (u_t, \chi, \theta)$, the electrical system can be written as

$$A(w_t)q_t = \psi \tag{5.8}$$

where $A(w_t)$ is the potential coefficient matrix and ψ is the potential vector. Using the trapezoidal rule discretization for the mechanical system

$$M(w_t, v_t)(\frac{2}{h}(v_t - v_{t-1}) - acc_{t-1}) = (P - F)(w_t, q_t) \tag{5.9}$$

$$2/h(w_t - w_{t-1}) = v_t + v_{t-1} \tag{5.10}$$

where $acc_t = \frac{2}{h}(v_t - v_{t-1}) - acc_{t-1} = M^{-1}(w_t, v_t)(P - F)(w_t, q_t)$ and h is the time step.

Generally writing

$$M(w_t) + F(w_t) = P(w_t, q_t) \tag{5.11}$$

Therefore we have in the discretized form

$$w_t = H_M(q_t) \tag{5.12}$$

$$q_t = H_E(w_t) \tag{5.13}$$

where H_M and H_E describe the relation between q and u at time t.

It should be noted that the Trapezoidal rule is guaranteed to conserve energy and momentum for linear Hamilton's equations only [26] but not for the nonlinear equations here. However the Trapezoidal rule has worked very well in all numerical experiments performed under this thesis.

5.4 Results

Figure 5-1 is a time simulation for the mirror where pull-in occurs for a high enough differential voltage. In Figure 5-2 the vertical levitation (Z_R) is most significant reaching as high 1 μm.

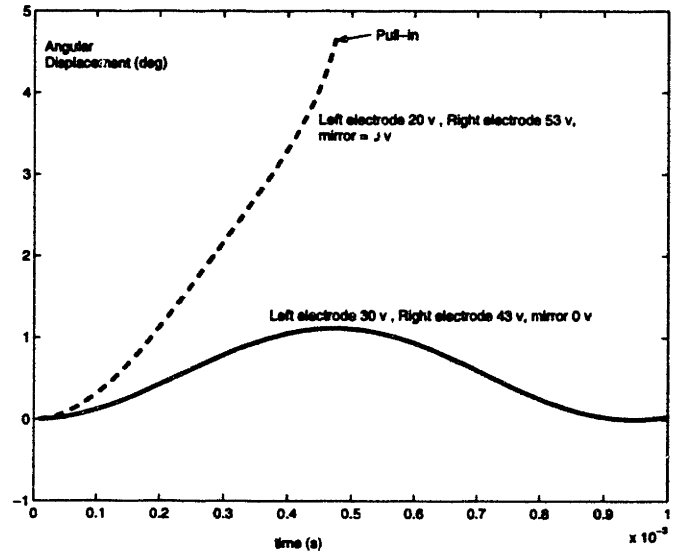


Figure 5-1: Time simulation of mirror. Observe that pull-in occurs for the 20-53 v case

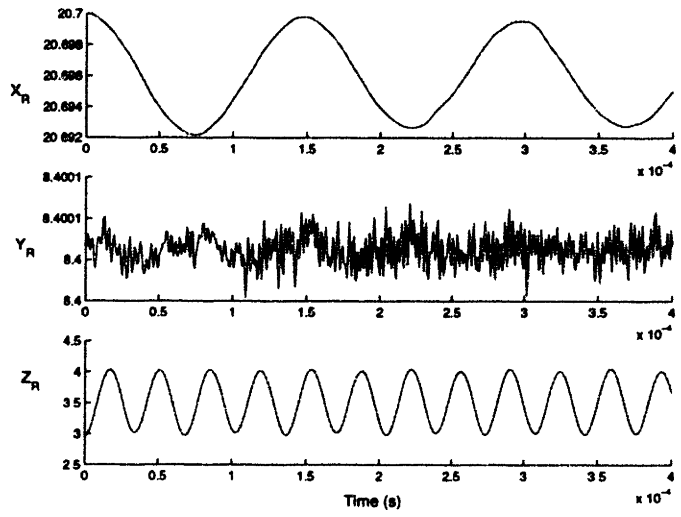


Figure 5-2: Time simulation of resonator kept at 0 v (with a ground) with both side combs at 50 v. Translation parameters - X_R, Y_R, Z_R

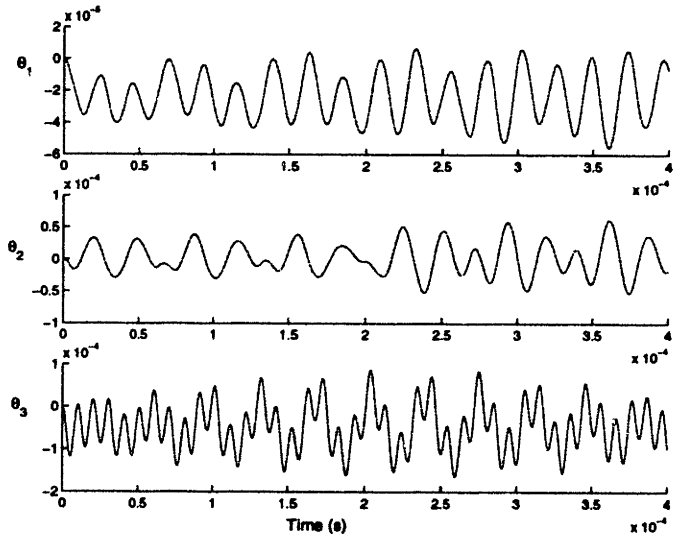


Figure 5-3: Time simulation of resonator kept at 0 v (with a ground) with both side combs at 50 v. Rotation parameters - $\theta_1, \theta_2, \theta_3$

Chapter 6

Model Order Reduction for Fully Elastic Coupled Systems

6.1 Introduction

In the previous chapters we explored efficiency improvements to the coupled domain solver. However when we wish to perform system level simulation or optimization, calling the coupled solver at every iteration will be very expensive. Instead we would like to generate low order models for device simulation which will accurately represent input-output behavior while at the same time are cheap to simulate because of their low order therefore giving us a quick and accurate first stab at system design. We adapt the Arnoldi based model order reduction technique for our coupled problem and then in the results we present several examples.

6.2 Coupled Domain Reduction

Here we adapt the Arnoldi based approach used by [13] and described in Chapter 2 to our coupled domain problem by implicit techniques. In contrast, the approach in [4] produced small-signal models with extremely accurate frequency responses, but exploited the fact that system matrices were explicitly available. In addition we will develop an approach that incorporates *both* the linear and quadratic dependencies of

the electrostatic forces on the input i.e. voltage.

The basic equations of coupled domain system (without damping) are

$$\begin{aligned} M\,\ddot{x} + F(x) &= P(x,q) \\ A(x)\,q &= \phi_0 + \Delta\phi\, v \end{aligned} \tag{6.1}$$

where x is the state space, F the force due to internal stresses, P the external surface force which depends on the charge q, A^{-1} is the potential coefficient matrix, ϕ_0 is the bias potential vector which can undergo a perturbation v in the direction $\Delta\phi$. $\Delta\phi$ consists of non-zero entries corresponding to surfaces on which ϕ_0 is perturbed and zeroes otherwise. Let us the assume that the original fully nonlinear model is completely elastic and that we would like to generate a completely elastic reduced model. To apply the model reduction technique described in section 2.6.3, the strategy here will be to linearize the above equations about the bias point (x_0, ϕ_0).

$$\begin{aligned} M\,\ddot{u} + \underbrace{\left(\frac{\partial F}{\partial x} - \frac{\partial P}{\partial x} - \frac{\partial P}{\partial q}\frac{\partial q}{\partial x}\right)}_{K} u \;=\; & \underbrace{\frac{\partial P}{\partial q}\frac{\partial q}{\partial v}}_{RHS_1} v + \\ & \underbrace{\frac{1}{2}\frac{\partial}{\partial v}\left[\frac{\partial P}{\partial q}\frac{\partial q}{\partial v}\right]}_{RHS_2} v^2 \end{aligned} \tag{6.2}$$

Observe that the right hand side is strictly not a linearization. However the quadratic term is very important as it reflects the essential "squaring" property of electrostatic force.

At the bias point ϕ_0 the charge q_0 is given as

$$q_0 = A^{-1}(x_0)\phi_0$$

Perturbing ϕ_0 by v, the change in q^2 is given by

$$\begin{aligned} q_i^2 &= \phi_0^T A^{-1}(i,:)^T A^{-1}(i,:)\phi_0 \\ \Delta q_i^2 &= 2\phi_0^T A^{-1}(i,:)^T A^{-1}(i,:)\Delta\phi\, v + \Delta\phi^T A^{-1}(i,:)^T A^{-1}(i,:)\Delta\phi\, v^2 \end{aligned}$$

We know from Chapter 3 that the force linearly depends on q_i^2.

This tells us that if the bias ϕ_0 is 0, just collecting the linear term will not give us any force and the quadratic term must be included. As a result we can expect that if the input is a sine wave of frequency ω the response has frequencies ω and 2ω. Note the computation is straightforward. $A^{-1}(i,:)\phi_0$ is simply $q_0(i)$ and $A^{-1}(i,:)\Delta\phi$ is i^{th} coordinate of the charge computed with $\Delta\phi$ as the voltage. Therefore computing the right hand side force term just involves an extra electrostatic black box call over the equilibrium charge calculation. Also note that prior to model order reduction we must have already explicitly computed $\frac{\partial P}{\partial q}$ (which is needed for solution of the outer Newton loop) at the equilibrium point. For convenience we now calculate "$\frac{\partial P}{\partial q^2}$". Also while $\frac{\partial F}{\partial x} - \frac{\partial P}{\partial x}$ is known explicitly $\frac{\partial q}{\partial x}$ is computed using finite differences as in (3.11).

In implementation calculating the corresponding Krylov basis for the above equation is very slow because the inverse of the K matrix has to be applied using an iterative procedure such as GMRES because of its implicit component and this procedure takes a lot of iterations because of the wide range of eigenvalues (or mode shapes) of the stiffness matrix. Therefore since $(\frac{\partial F}{\partial x} - \frac{\partial P}{\partial x})$ is already known explicitly we factor it and use it as a preconditioner for finding K^{-1}. As a consequence the number of iterations for our examples rarely ever exceed four. Although V is orthogonalized while calculating the Arnoldi basis (and this also avoids ill-conditioning of the reduced matrices), in the damping case V_1 is not orthonormal and is therefore orthogonalized.

Also note that with a non-symmetric K, there is no known way to diagonalize both M and K simultaneously. Since we have two right sides we can generate two sets of matrices

$$(\tilde{M}_1, \tilde{D}_1, \tilde{K}_1, \tilde{b}_1, \tilde{c}_1, u_1),\ (\tilde{M}_2, \tilde{D}_2, \tilde{K}_2, \tilde{b}_2, \tilde{c}_2, u_2)$$

i.e. one for each right hand side and the actual solution is $u = u_1 + u_2$ (by linearity). Block Arnoldi which can be used when multiple right hand sides are present is not implemented here but is described in a different multiple right hand sides context in

the rigid-elastic case [Chapter 7].

6.2.1 Stopping criterion

From the user's point of view, the inputs required are the bias direction ϕ_0 and the order of the reduced model. If the order of the reduced model can be variable we need to determine a stopping criterion for fixing the order of the model at the smallest possible number. One way is to stop when the "residual"[1] of the Arnoldi process is small. To gain some insight as to why this might be useful lets assume that we reach an invariant subspace after the i^{th} iteration in the Arnoldi process for the no-damping case i.e.

$$\begin{aligned} &(K^{-1}M)^j K^{-1}b = \\ &< K^{-1}b, (K^{-1}M)K^{-1}b, \ldots, (K^{-1}M)^{i-1}K^{-1}b > \\ &\forall j \geq i \end{aligned} \tag{6.3}$$

or the "residual" is zero.

First note that i moments have matched. The $(i+1)$ moment is given by

$$c^T(K^{-1}M)^i K^{-1}b \tag{6.4}$$

For the reduced order model the $(i+1)$ moment is given by

$$c^T V((V^T K V)^{-1}(V^T M V))^i (V^T K V)^{-1} V^T b \tag{6.5}$$

where V (spanning the columns of the subspace in (6.3))is taken to be orthonormal. But

$$\begin{aligned} &V((V^T K V)^{-1}(V^T M V))^{i-1}(V^T K V)^{-1}V^T b = \\ &(K^{-1}M)^{i-1}K^{-1}b \end{aligned}$$

from the moment matching property (Note that c really doesn't have to be present

[1]The GMRES residual i.e. $r = (K^{-1}b - K^{-1}MVy)$ where y s.t. $||r||$ is minimized

for such types of proofs). Therefore (6.5) is also equal to

$$c^T V(V^T K V)^{-1} V^T M (K^{-1} M)^{i-1} K^{-1} b \tag{6.6}$$

A unique solution ψ to

$$(V^T K V)\psi = V^T M (K^{-1} M)^{i-1} K^{-1} b$$

exists if

$$KV\psi = M(K^{-1}M)^{i-1}K^{-1}b \tag{6.7}$$

There is such a ψ because $(K^{-1}M)^i K^{-1} b \subset V$ from (6.3). Therefore $V\psi = (K^{-1}M)^i K^{-1} b$ and

$$c^T V((V^T K V)^{-1}(V^T M V))^i (V^T K V)^{-1} V^T b =$$
$$c^T (K^{-1}M)^i K^{-1} b$$

This procedure extends for moments $(i+2), \ldots, \infty$. Therefore all moments match when we reach an invariant subspace. This subspace is the same as the invariant subspace which is reached in an expansion-around-∞ Arnoldi process. It follows by a simple manipulation of (6.3) that the subspace

$$< M^{-1}b, (M^{-1}K)M^{-1}b, \ldots, (M^{-1}K)^{i-1}M^{-1}b > \;=$$
$$< K^{-1}b, (K^{-1}M)K^{-1}b, \ldots, (K^{-1}M)^{i-1}K^{-1}b >$$

and is also invariant. This analysis also holds true for the damping case with K etc. replaced by $\hat{K}$ etc. In practice however, we stop the Arnoldi process much before the "residual" becomes very small and because of the error, the above analysis may not be a good indicator. But it has been observed to be a good stopping criterion.

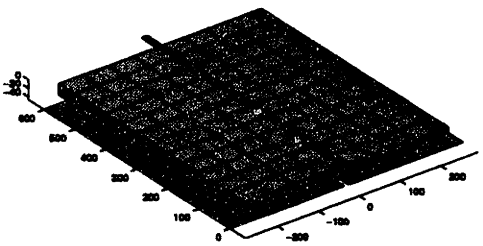


Figure 6-1: 3-D Model of Micromirror has thousands of degrees of freedom. Expensive to Simulate.

6.3 Results

Most MEMS devices are not packaged in vacuum and so we can expect to see fluid(air) damping. But we will not incorporate air damping here. Instead we introduce a fictitious positive definite damping matrix D to test our model reduction. First we reduce a "plain" linearized (i.e. no quadratic term of voltage) cantilever beam to a 15^{th} order model and compare it with the full simulation by taking it to steady state shown in Figure 6-3. The steady state error is large. For a lightly damped model we compare the transient responses of the fully nonlinear model, the 15^{th} reduced models of the "plain" linearized and linearized systems in Figure 6-4. The reduced model of the linearized system matches very well with the full model.

In the micromirror in Figure 6-1, the device input is a differential voltage applied to a pair of plates beneath the mirror, and the output is the micromirror's angular deflection. As shown in Figure 6-2 the quasistatic simulation results are in close agreement with experimental data [28]. The nearly 7000 degrees of freedom mirror is reduced into two 15^{th} order reduced models (one each for the linear and quadratic right hand sides (6.2)) without any damping and we see again that the fully elastic reduced model simulation matches well (Figure 6-5) with the simulation of the full model (rigid-elastic) [20].

Next we examine the frequency responses of the reduced model at different bias points. In the recipe to determine a linear reduced order model the first step is

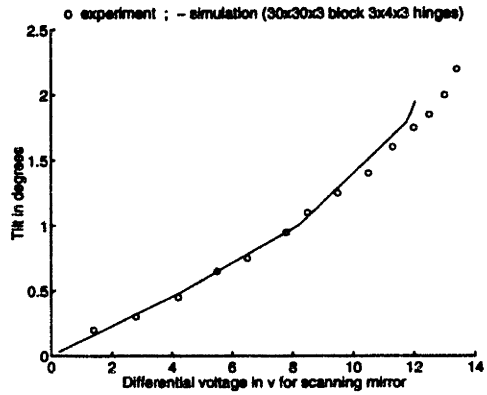


Figure 6-2: Micromirror displacement versus voltage.

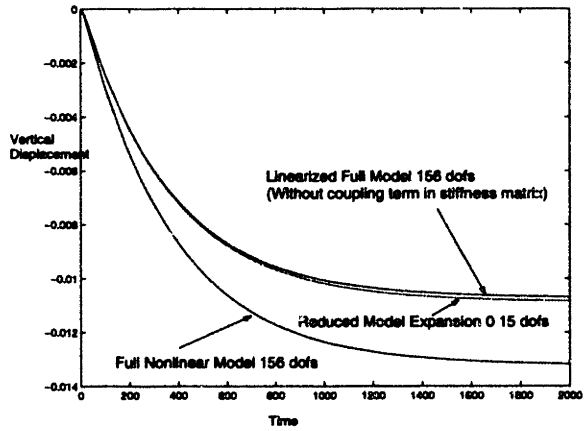


Figure 6-3: Cantilever beam voltage step responses (heavily damped case) using numerical simulation and a generated linear-only macromodel.

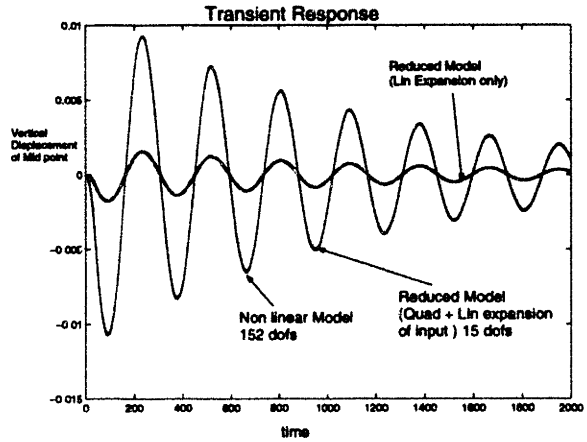


Figure 6-4: Cantilever beam voltage transient responses (lightly damped case) using numerical simulation and two generated macromodels.

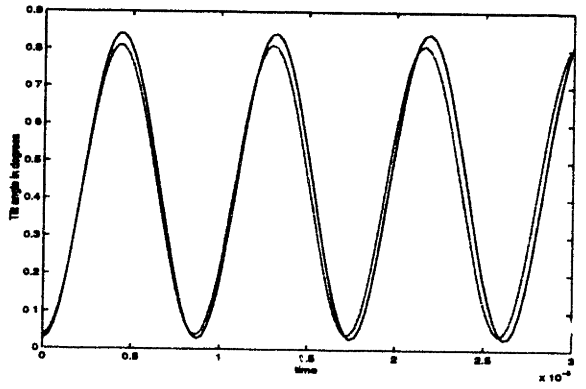


Figure 6-5: Comparing micromirror differential voltage step responses computing using numerical simulation and 15th order macromodels.

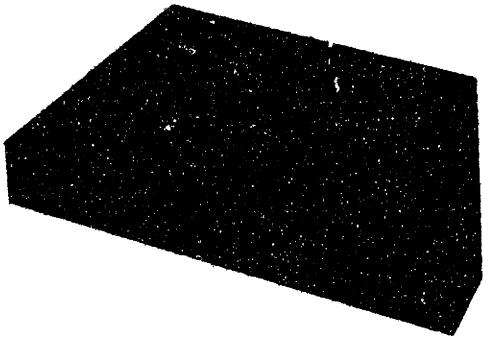


Figure 6-6: Cantilever beam over 0 v plane

to determine the regime in which the original system behaves more or less linearly. The following results were obtained by a modification of the CFD-ACE code of the Computational Fluid Dynamics Corporation, done by the author. The test case is a simple 954 dofs cantilever beam as before (Figure 6-6) and we linearize about the bias points shown in Figure 6-7. Fictitious damping is introduced in the form of Rayleigh damping (D is a linear combination of K and M). From Figure 6-8 we see that the frequency response of both the Arnoldi and Eigen reduced (eigenmodes from K and M only) models (ten Arnoldi vectors and ten Eigen vectors) match well with the full model near the first eigenvalue of the undamped system. This should not be surprising because in an expansion-around-0 reduction we expect the models to match better at lower frequencies.

Figure 6-9 plots the frequency responses for the bias at 100 v and 300 v. The

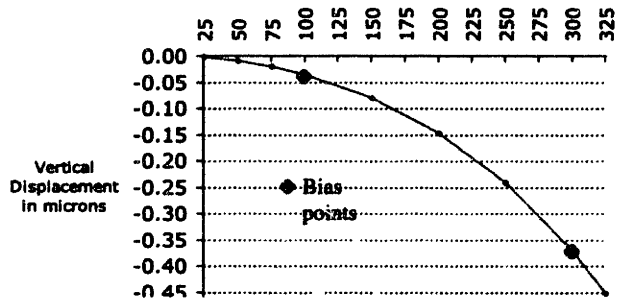


Figure 6-7: Static voltage-displacement curve

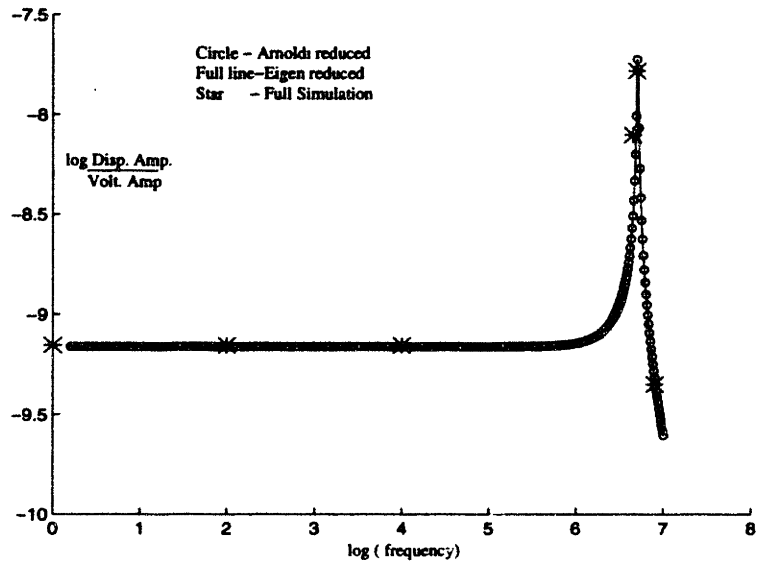


Figure 6-8: Frequency response

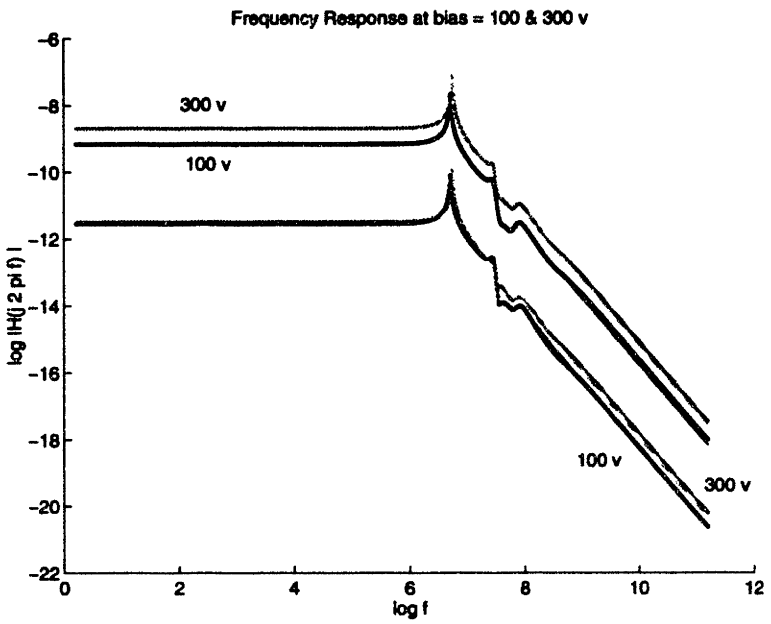


Figure 6-9: Frequency responses for linear and nonlinear inputs (bottom curve) at 100 v and 300 v

response contribution by the nonlinear input is a couple of orders magnitude less than the linear input contribution. Also the Arnoldi and Eigen reduced model plots are not distinguished since they match very well. It is interesting to point out that the second eigenvalue of the original system corresponds to a horizontal mode which is not represented by any of the peaks of the response (since the vertical response that we measure is not affected by horizontal modes. However it does tell us that we needn't have collected the second eigenmode when we collected the 10 smallest eigenmodes for eigenreduction. In fact the Arnoldi method's advantage lies precisely here i.e. efficient capturing of the modes in the response direction. One can see this clearly by noting that the expansion-around-∞ subspace is the same as the reachability space of (2.25)). Figure 6-10 shows the differences between the peak frequencies and values of the biases.

However if the number of mechanical degrees of freedom is large as in the case of the micromirror the reduction process becomes expensive because of the cost of finding a static solution first/cost of factoring. In such cases it is possible to derive a reduced rigid-elastic model directly from the full rigid-elastic model which is explored

Linear input only	Bias = 300 v	Bias = 100 v
Peak value	[illegible]	[illegible]
Peak frequency	5.518488 MHz	5.032921 MHz

Figure 6-10: Peak comparison for bias = 100 v and 300 v

in the next chapter.

6.4 Conclusion

We have successfully demonstrated a fully automatic technique to take a partly implicit system and reduce it to a much smaller explicit system that accurately captures the small signal behavior of the original system.

Chapter 7

Generation of Rigid-Elastic Models

7.1 Introduction

In the last chapter we remarked that it is expensive to generate reduced order models of devices such as the resonator with many degrees of freedom. This and because we treat these devices as rigid-elastic anyway when simulating them in "full" leads to the idea that we can reduce their models in such a way so as to preserve the rigid-elastic behavior.

7.2 Rigid-Elastic Reduction

As in the elastic case one can think of linearizing the nonlinear models in terms of the elastic degrees of freedom and the rigid degrees of freedom namely the three translation variables and the three degrees of rotation. Then we can perform Arnoldi based reduction as before. The problem with this approach is that after linearization(and reduction) the nonlinear behavior of the rigid body is lost i.e. (5.2). The end result will be of the form (2.25) with no guarantee of stability. If the full model is an angles based model, the reduced model although angle based will not contain sines and cosines(as in the full model) and therefore the structure is lost. Instead we would like to retain the nonlinearity of the rigid body motion.

The idea is to retain the six rigid degrees of freedom while reducing the elastic

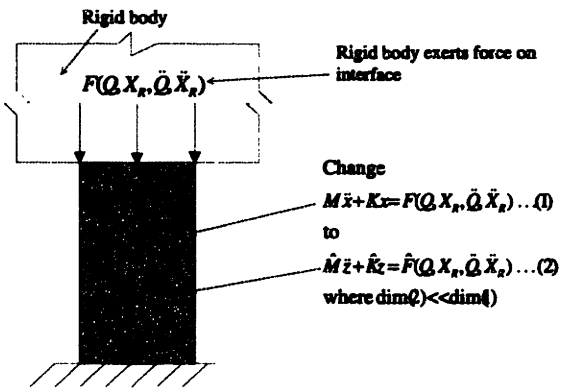


Figure 7-1: Reduction of degrees of freedom of tether

degrees of freedom which still slow down the solver. This reduction of the elastic part of the structure is done using block Arnoldi (see Appendix B).

Any point on the rigid-elastic interface can be expressed as (4.1) and (4.2) or

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} q_1 & q_4 & q_7 \\ q_2 & q_5 & q_8 \\ q_3 & q_6 & q_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix}$$

Here $[X_R Y_R Z_R]^T$ or $(X,Y,Z)_R$ are the translation parameters of the rigid body and $[XYZ]^T$ or (X,Y,Z) are the relative coordinates of an interface point w.r.t. the center of translation and rotation. We can think of (x,y,z) as a linear combination of q_i's and $(X,Y,Z)_R$ which total to 12 in number. Then the linearized system of equations for the elastic part of the structure is written as

$$\begin{aligned} M\,\ddot{x}_{\text{elastic}} + K\,x_{\text{elastic}} &= -\tilde{M}\,\ddot{x}_{\text{interface}} - \tilde{K}\,x_{\text{interface}} \\ M\,\ddot{x}_{\text{elastic}} + K\,x_{\text{elastic}} &= -\sum_{i=1}^{9}(\hat{M}_i\ddot{q}_i + \hat{K}_i q_i) \\ &\quad -\hat{M}_{10}\ddot{X}_R - \hat{M}_{11}\ddot{Y}_R - \hat{M}_{12}\ddot{Z}_R \\ &\quad -\hat{K}_{10}X_R - \hat{K}_{11}Y_R - \hat{K}_{12}Z_R \end{aligned} \tag{7.1}$$

Note that for any node on the interface the q_i's and $(X,Y,Z)_R$ are the same. Only (X,Y,Z) changes from node to node. Now the q_i's,$(X,Y,Z)_R$, $\ddot{q}_i$'s and$(\ddot{X},\ddot{Y},\ddot{Z})_R$ [1]

[1] also represented here as X_R

can be thought of as inputs to the left hand side of (7.1). (7.1) is therefore reduced w.r.t. these inputs and we obtain the transformation

$$x = Vz \tag{7.2}$$

We use (7.2) again for reducing the elastic degrees of freedom in the equilibrium equation for the rigid body. Applying the transformation to the full linearized mass matrix for example looks like[2]

$$\begin{aligned} &\begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} M_{EE} & M_{ER} \\ M_{RE} & M_{RR} \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} V^T M_{EE} V & V^T M_{ER} \\ M_{RE} V & M_{RR} \end{bmatrix} \end{aligned} \tag{7.3}$$

following the notation of Chapter 4. Note that the rigid part is unaffected and that we get a desired reduction in the elastic degrees of freedom. The equation for the elastic part linearized about q_i^0 and $(X^0, Y^0, Z^0)_R$ respectively is

$$\begin{aligned} \bar{M}_{EE}\,\ddot{z} + \bar{K}_{EE}\,z \;=\; & -\sum_{i=1}^{9} (\bar{M}_i \ddot{q}_i + \bar{K}_i (q_i - q_i^0) \\ & -\bar{M}_{10}\ddot{X}_R - \bar{M}_{11}\ddot{Y}_R - \bar{M}_{12}\ddot{Z}_R \\ & -\bar{K}_{10}(X_R - X_R^0) - \bar{K}_{11}(Y_R - Y_R^0) \\ & -\bar{K}_{12}(Z_R - Z_R^0) \end{aligned} \tag{7.4}$$

where $\bar{M}_{EE}$ is $V^T M_{EE} V$ etc.

[2]It might seem as if the rigid body equations need to be linearized w.r.t. the rigid dofs but except for the surface and internal force contribution, actually they are not and the discussion here serves to illustrate the transformation to the Jacobian for both the stiffness and mass terms

The "reduced" translation rigid body equations are

$$\underbrace{\begin{bmatrix} Mass & & \\ & Mass & \\ & & Mass \end{bmatrix}}_{M_{RR}^{force}} \begin{bmatrix} \ddot{X}_R \\ \ddot{Y}_R \\ \ddot{Z}_R \end{bmatrix} + \bar{K}_{RE}{}^{force} z + \tag{7.5}$$

$$M_{RR}^{\text{interface } force} \begin{bmatrix} \{\ddot{q}_i\} \\ \{\ddot{X}\}_R \end{bmatrix} + (K_{RR}^{\text{interface } force} + K_{RR}^{\text{surface } force}) \begin{bmatrix} \{q_i - q_i^0\} \\ \{X - X^0\}_R \end{bmatrix}$$

$$= RHS_1^{force} v + RHS_2^{force} v^2$$

where [3] RHS_1^{force} and RHS_2^{force} represent the linear and quadratic contributions of the electrostatic force [4] (not moment). Similarly the "reduced" rotation equation is

$$\begin{aligned} J\dot{W}(q_i) + W(q_i) \times JW(q_i) \; + \; & Q^T(F^{\text{interface}}) = Q^T(RHS_1^{momt} v + RHS_2^{momt} v^2) \\ F^{\text{interface}} \; = \; & \bar{M}_{RE}^{momt} \ddot{z} + \bar{K}_{RE}^{momt} z + M_{RR}^{\text{interface } momt} \begin{bmatrix} \{\ddot{q}_i\} \\ \{\ddot{X}\}_R \end{bmatrix} + \\ & (K_{RR}^{\text{interface } momt} + K_{RR}^{\text{surface } momt}) \begin{bmatrix} \{q_i - q_i^0\} \\ \{X - X^0\}_R \end{bmatrix} \end{aligned} \tag{7.6}$$

Note that in (7.5) and (7.6) $M_{RR}^{\text{interface } momt,force}$ represents the mass matrix contribution of the interface or internal elastic forces and $K_{RR}^{\text{interface } momt,force}$ is the stiffness matrix contribution of the internal forces and $K_{RR}^{\text{surface } momt,force}$ is the surface force contribution.

It should be pointed out that in this formulation we have not considered for reduction purposes the surface forces on the elastic portion as inputs. It is assumed that the surface forces on the elastic parts (tethers for eg.) are much smaller than the internal forces which the rigid body exerts on them. The surface forces contribute to the elastic stiffness matrix and therefore we need to implicitly apply K_{EE}^{-1}. If 2 vectors

[3] see Chapter 4 for notation

[4] see Chapter 6

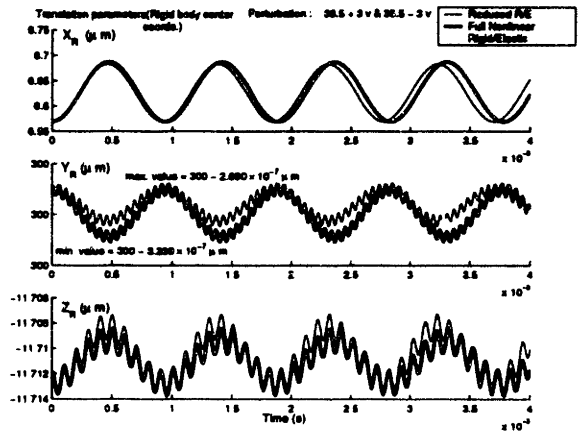


Figure 7-2: Step Response : Scanning mirror rigid body translation parameters comparison

are provided for each of the 24 inputs then this can become expensive as we would need to call H_E 48 times. Therefore this method works fastest when the surface force effects are not included in K_{EE}.

From the user's point of view the only inputs required are the order of the model ($\geq$ 24), the tagging of elements as rigid plus the tolerances required for the coupled domain solver. After that the model is automatically reduced. The reduced model can then be conveniently solved using a simple MATLAB (r) script which calls MAPLE (r) for analytical computations. Note that while solving the reduced model we can replace the q_i's with the Tait-Bryan angles formulation for example (4.2).

7.3 Results

The scanning mirror example of Chapter 6 (Figure 6-2) is reduced with the technique described in the previous section. $K_{RR}^{\text{surface } momt,force}$ should be included otherwise the time period and amplitude can be significantly different from the full simulation.

The device input is a differential voltage applied to a pair of plates beneath the mirror each kept at 37.5 v with the mirror at 0 v, and the output of interest is the micromirror's angular deflection. The **first procedure** as always in a strategy to generate a reduced order model is to obtain the static angle displacement curve - Figure 4-10. Then preferably we would like to **operate** in the **linear regime**. Here

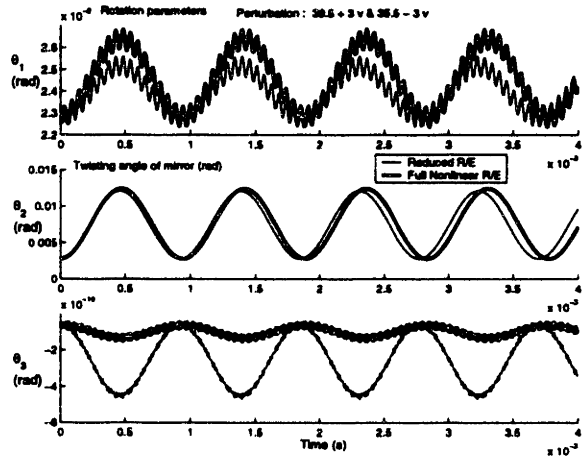


Figure 7-3: Step Response : Scanning mirror rigid body rotation parameters comparison

we choose a bias of 2 v. Figures 6-2 and 6-3 show the comparison between the 54^{th} [5] order reduced and full models for a step input of 3 v. θ_2 is the tilt which reveals a close match between the models (compare with Figure 6-5). For other parameters there is at least a match in the frequencies. For θ_1 and θ_3, the mismatch in amplitudes are not important as the amplitudes are very small. As the input is increased, nonlinear effects become more prominent and the reduced model cannot capture them. The reduction in complexity is quite significant as the the linearized Jacobian for the reduced system is 54×54 in this example compared to 1476×1476 of the original system without the computational costs of many 3-D coupled domain solves.

[5] 6 for the rigid degrees of freedom, 48 for the elastic part in particular 2 for each of the 24 inputs

Chapter 8

Conclusion

In the previous chapters we have described a multi-level Newton technique for coupled domain problems which is most efficient when the black box solvers not only return an output for a given input but also a corresponding perturbation in output for a perturbation in input. One can see that if in the future more physics are involved or better individual solvers are available then one can "plug" them into this architecture. Then when the structure behaves rigidly we have shown that it pays to use a rigid-elastic formulation which also matches the fully elastic formulation very well. However it is the user's decision to tag the elements as rigid so automatic selection remains an incomplete task. With this rigid-elastic formulation we have shown that it is easy and insightful to study the effects of geometry on the structure thus helping us to factor in geometry in design.

Since full 3-D models are expensive to run for system level simulation, accurate and cheap reduced order models are needed and we described the reduction of the electromechanical structure dealing with mostly implicit information based on the popular Arnoldi based methods and in the end we had explicit matrices. This method is extended to generating accurate rigid-elastic models which is more attractive for models with rigid-elastic behavior by reducing only the elastic part of the structure. However we have neglected the more realistic fluid damping and did not consider damping at all for the rigid-elastic case. However for the rigid-elastic it is easily seen that the reduction is independent of computing the damping effect. As a very simple

example the linearized damping matrix can be incorporated by calling the recently developed fast fluid solvers for perturbations in geometry for both the fully elastic and rigid-elastic cases. Finally we have described the formulation for linear model reduction only and research is still ongoing in the development of nonlinear models.

Appendix A

Reachability Space

Converting the second order equation to first order we have

$$\hat{M}\,\dot{z} + \hat{K}\,z = \hat{b}v$$

With initial condition $z = 0$ the solution at anytime t is given as

$$x(t) = \int_0^t e^{(t-\tau)\hat{M}^{-1}\hat{K}}\hat{M}^{-1}\hat{b}\,v(\tau)d\tau$$

Expanding the exponential using the Cayley-Hamilton theorem the reachable space is obtained as

$$< \hat{M}^{-1}\hat{b}, (\hat{M}^{-1}\hat{K})\hat{M}^{-1}\hat{b}, (\hat{M}^{-1}\hat{K})^2\hat{M}^{-1}\hat{b}, \ldots, (\hat{M}^{-1}\hat{K})^n\hat{M}^{-1}\hat{b} >$$

where n is the dimension of the system. This is the same as series obtained by the expansion-around-∞ Taylor series of $H(s)$ (2.25). This would suggest that perhaps we should use this expansion instead of expansion-around-0. However there is no practical use of using this series unless we reach an invariant subspace with dimension much smaller than n. However if an invariant subspace is reached we saw in Chapter 6, it is identical to an equal dimension expansion-around-0 subspace which is also invariant. This justifies using expansion-around-0 which also matches

the steady state.

Appendix B

Block Arnoldi

When multiple RHS inputs are present as in the first order system below with three inputs v_1, v_2 and v_3

$$\begin{aligned} \dot{x} &= Ax + b_1 v_1 + b_2 v_2 + b_3 v_3 \\ y &= c^T x \\ Y(s) &= c^T (s - A)^{-1} (b_1 V_1(s) + b_2 V_2(s) + b_3 V_3(s)) \end{aligned} \tag{B.1}$$

we can consider each input separately (superposition of solutions) and generate three different reduced order models. Each reduced model will match a set of moments with the original model for the *corresponding* input.

$$\begin{aligned} Y_1(s) &= c^T Q_1 (s - Q_1^T A Q_1)^{-1} Q_1^T b_1 V_1(s) \\ Y_2(s) &= c^T Q_2 (s - Q_2^T A Q_2)^{-1} Q_2^T b_2 V_2(s) \\ Y_3(s) &= c^T Q_3 (s - Q_3^T A Q_3)^{-1} Q_3^T b_3 V_3(s) \\ Y_a(s) &= Y_1(s) + Y_2(s) + Y_3(s) \end{aligned} \tag{B.2}$$

For block Arnoldi Q_1, Q_2 and Q_3 are combined together into a single matrix $Q = [Q_1 Q_2 Q_3]$ and assuming Q is full rank the reduced model is

$$Y_{ba}(s) = c^T Q (s - Q^T A Q)^{-1} Q^T (b_1 V_1(s) + b_2 V_2(s) + b_3 V_3(s)) \tag{B.3}$$

Now both (B.2) and (B.3) match exactly the same moments but results have appeared in the literature [30] which suggest that for the same point-wise accuracy (B.3) requires lesser computation because of cheaper and fewer matrix-vector products compared to individual Arnoldi and Pade'. Note that the combined cost of factoring the individual matrices of (B.2) is $3i^3$ compared to the $27i^3$ cost of factoring the reduced matrix of (B.3). It is tempting to think that for $i = 1, 2, 3$

$$||c^T Q_i (s - Q_i^T A Q_i)^{-1} Q_i^T b_i - c^T (s - A)^{-1} (b_i|| \stackrel{?}{\geq} ||c^T Q (s - Q^T A Q)^{-1} Q^T b_i - c^T (s - A)^{-1} b_i|| \quad \forall s \in [-j\infty, j\infty]$$

because Q has more vectors (spanning[1] either A^{-1} or A) than Q_i. A script is written in MATLAB (r) to test whether this hypothesis is ever violated.

```
% To see if Block Arnoldi reduces the norm of the error in each
% component of the transfer function
function [] = testblkarnoldi(N,order)

A = randn(N);
eyeN = eye(N);
b = randn(N,1);
c = ones(N,1);
[L,U] = lu(A);
% Determine Arnoldi basis                                          10
[V] = MORgmres_krylov(L,U,eyeN, A\b(:,1), order-1,norm(A\b(:,1)));
Ar1 = V'*A*V;
br1 = V'*b;
cr1 = V'*c;
eyeNr1 = eye(size(Ar1,1));

% To include the effect of Block Arnoldi,
```

[1] expansion around 0 or ∞

```
% add some vectors spanning A^{-1}. randn(N,3) like other right hand sides
V2 = orth([V A\randn(N,3)]);
Ar2 = V2'*A*V2;                                                    20
br2 = V2'*b;
cr2 = V2'*c;
eyeNr2 = eye(size(Ar2,1));

counter = 0;
for s=0:0.01:10
  counter = counter+1;
  X(counter) = s;
  % 'i' is really the imaginary number 'j'
  w = i*s;                                                         30
  Y1(counter) = norm(c'*((w*eyeN-A)\b) - cr1'*((w*eyeNr1-Ar1)\br1));
  Y2(counter) = norm(c'*((w*eyeN-A)\b) - cr2'*((w*eyeNr2-Ar2)\br2));
  Y3(counter) = norm(c'*((w*eyeN-A)\b));
end

clf;
subplot(2,1,1);
%Individual Arnoldi
plot(X,Y1,'b');
hold on;                                                           40
%"Block" Arnoldi
plot(X,Y2,'r-');
subplot(2,1,2);
%original |H(j omega)| - transfer function
plot(X,Y3);
```

From (B-1) we can see that the addition of vectors such as generated from additional RHS vectors while decreasing the error for most of s , actually increase the

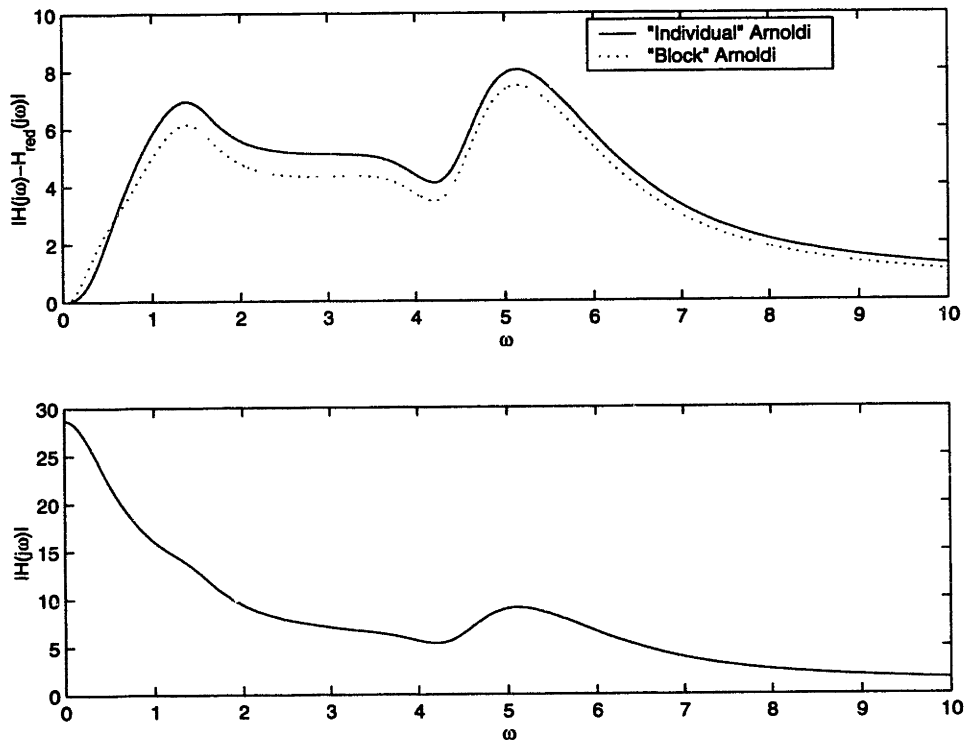


Figure B-1: Individual Arnoldi and Block Arnoldi comparison

error in this case near $s = 0$ (where the reduced and the full models match best).

Appendix C

Parameter Balancing

The parameter balancing technique as described for example in [11] is useful in determining an estimate of the minimum complexity when the cost function has monotonically increasing and decreasing parts. Specifically let the cost function

$$c(n) = f(n) + g(n)$$

where $f(n)$ is monotonically increasing in n and $g(n)$ is monotonically decreasing in n. Let $c(n_*)$ be the minimum value of $c(n)$ at $n = n_*$. Also let n_0 be that value of n at which $g(n)$ equals $f(n)$ i.e.

$$g(n_0) = f(n_0)$$

Now for $n_* > n_0$

$$c(n_0) = g(n_0) + f(n_0) = 2f(n_0) \leq 2f(n_*) \leq 2c(n_*) \tag{C.1}$$

since $f(n)$ is monotonically increasing and both $f(n)$ and $g(n)$ are positively valued for $n > 0$.

Similarly for $n_* < n_0$

$$c(n_0) = g(n_0) + f(n_0) = 2g(n_0) \leq 2g(n_*) \leq 2c(n_*) \tag{C.2}$$

Therefore from (C.1) and (C.2) $c(n_0) \leq 2c(n_*)$. Now if it is much more simpler to compute n_0 than n_*, then by selecting $n = n_0$ we will not be off the minimum by more than a constant.

Bibliography

[1] B. E. Boser and R. T. Howe,"Surface Micromachined Accelerometers", IEEE Journal of Solid-state Circuits, Vol. 31, No. 3, March 1996.

[2] N. Maluf,An Introduction to Microelectromechanical Systems Engineering, Artech House Inc,2000.

[3] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric linear systems", SIAM J. Sci. Statist. Comput., 7(3), pp. 105-126,1986.

[4] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, § 5.3, 2nd Ed.,Springer.

[5] P.N. Brown and Y. Saad, "Hybrid Krylov Methods for Nonlinear Systems of Equations",SIAM J. Sci. Statis. Comput., 11, pp. 450-481,1990.

[6] R.S. Dembo, S.C. Eisenstat and T. Steihaug, "Inexact Newton Methods", Siam J. Numerical Anal., 19, pp. 400-408, 1982.

[7] M. Pernice and H.F. Walker, "NITSOL: A Newton Iterative Solver for Nonlinear Systems",SIAM J. Sci. Comput., Vol. 19, No. 1, pp.302-318, Jan. 1998.

[8] B. Morini, "Convergence Behaviour of Inexact Newton Methods", Mathematics of Computation, Vol. 68, No. 228, pp. 1605-1613, Mar. 1999.

[9] K.J. Bathe, Finite Element Procedures, Prentice-Hall Inc., Englewood Cliffs, NJ,1996.

[10] J.R. Phillips and J. White, "A Precorrected-FFT for Electromagnetic Analysis of Complex 3-D Structures", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 16, No. 10, pp. 1059-1072, Oct. 1997

[11] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, Network Flows - Theory, Algorithms and Applications,Prentice-Hall Inc.,New,Jersey,1993.

[12] K. Nabors and J. White, "Fastcap: A Multipole Accelerated 3-D Capacitance Extraction Program",IEEE Transactions on CAD of Integrated Circuits and Systems,Vol. 10, pp. 1447-1459, Nov. 1991.

[13] T. Su and R. R. Craig Jr., "Model Reduction and Control of Flexible Structures Using Krylov Vectors", J. Guidance, 260-267, 1989.

[14] P. Feldmann and R.W. Freund, "Efficient Linear Circuit Analysis by Pade' Approximation via the Lanczos Process", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 14, No. 5, 639-649, May 1995.

[15] E.J. Grimme, "Krylov Projection Methods for Model Reduction", PhD. Thesis, University of Illinois at Urbana-Champaign, 1994.

[16] J.M. Ortega and W.C. Rheinboldt, Interative Solution of Nonlinear Equations in Several Variables ,Academic Press, New York,1970.

[17] M. Bachtold, J.G. Korvink and H. Baltes, "New Convergence Scheme for Self-Consistent Electromechanical Analysis of iMEMS",Proc. IEDM, pp. 605-608, Washington, 1995.

[18] 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",NUPAD V.,International Workshop on,1994.

[19] N.R. Aluru and J. White, "A Coupled Numerical Technique for Self-Consistent Analysis of Micro-Electro-Mechanical Systems", Microelectromechanical Sys-

tems (MEMS), ASME Dynamic Systems and Control (DSC) series, New York, Vol. 59, pp.275-280,1996.

[20] D. Ramaswamy, A. Aluru and J. White, "Fast Coupled-Domain Mixed-Regime Electromechanical Simulation," Proc. Transducers '99, The 10th International Conference on Sold-State Sensors and Actuators.

[21] D. Ramaswamy, N.R. Aluru, and J. White,"A mixed rigid/elastic formulation for an efficient analysis of electromechanical systems",Proc. MSM98, 1998 International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators.

[22] J. Wang and J. White, "Fast Algorithms for Computing Electrostatic Geometric Sensitivities", Proceedings of SISPAD 97, Boston MA, Sep, 1997.

[23] J. Baglama, D. Calvetti, G.H. Golub, L. Reichel, "Adaptively Preconditioned GMRES algorithms", SIAM Journal on Scientific Computing,Vol. 20, Num. 1,pp.243-269,1998.

[24] V.I. Arnold, Mathematical Methods of Classical Mechanics, Springer, New York, 1988.

[25] R.E. Roberson, R. Schwertassek, Dynamics of Multibody Systems, Springer-Verlag,1988.

[26] J.C. Simo, N. Tarrow and K.K. Wong,"Exact energy-momemtum conserving algorithms and symplectic schemes for nonlinear dynamics",Computer Methods in Applied Mechanics and Engineering, No. 100, pp.63-116, 1992.

[27] B. Mirtich, "Fast and accurate computation of polyhedral mass properties", Journal of Graphics Tools, 1(2), pp.31-50, 1996.

[28] D. L. Dickensheets and G.S. Kino, "Silicon-Micromachined Scanning Confocal Optical Microscope", Journal of Microelectromechanical Systems , Vol. 7, No. 1, March 1998.

[29] W.C. Tang, M.G. Lim, and R.T. Howe,"Electrostatic Comb Drive Levitation and Control Method",Journal of Microelectromechanical Systems, Vol. 1, No. 4, December 1992.

[30] L.M. Silveira, M. Kamon and J. White,"Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances associated with 3-D Interconnect Structures",pp.534-538, European Design and Test Conference, 1995.

[31] F. Wang and J. White,"Automatic Model Order Reduction of a Micro-device using the Arnoldi Approach", International Mechanical Engg. Congress and Exposition, pp.527-530, Nov. 1998.

THESIS PROCESSING SLIP

FIXED FIELD: ill. _____ name _____

index _____ biblio _____

► **COPIES:** (Archives) Aero Dewey (Barker) Hum

Lindgren Music Rotch Science Sche-Plough

TITLE VARIES: ► ☐ _____

_____

_____

NAME VARIES: ► ☐ _____

_____

IMPRINT: **(COPYRIGHT)** _____

_____

► **COLLATION:** _____

_____

► **ADD: DEGREE:** _____ ► **DEPT.:** _____

► **ADD: DEGREE:** _____ ► **DEPT.:** _____

SUPERVISORS: _____

_____

NOTES:

cat'r: date:

page: 179

► **DEPT:** C.E.

► **YEAR:** 2001 ► **DEGREE:** Ph.D.

► **NAME:** RAMASWAMY, Deepak

_____