Analysis of Realistic Large MEMS Devices

Per Ljung, Martin Bächtold, Mirko Spasojevic
Coyote Systems Inc.

ABSTRACT

A high-speed high-accuracy 3D field solver for the solution of coupled multi-physics encountered in MEMS is presented. The software AutoMEMS enables automatic model generation from layout, automatic meshing, adaptive mesh refinement of large complex geometries encountered with realistic MEMS devices. Arbitrary types of materials, general geometries, and all types of boundary conditions are supported to solve coupled electro-mechanical 3D fields.

Keywords: 3D field solver, BEM, boundary element method, adaptive refinement, coupled fields

1 INTRODUCTION

Because MEMS devices utilize multiple energy domains, they are inherently difficult to design, analyze and optimize. This work addresses the verification of user-created designs by simulating the interacting 3D physical fields. Only by numerically solving the partial differential equations (PDE) describing these fields can an accurate representation of the behavior of a MEMS device be obtained [1] [2] [3].

The premise of this work is to robustly automate the numerical solution of 3D fields, allowing both beginner and expert MEMS designers to rapidly design and optimize their devices. This entails that the simulator must be capable of modeling and analyzing entire MEMS devices, which are large systems with complex geometries. Further, to guarantee computational feasibility and ease of use, the multi-physics PDE solver must be fast, accurate and fully adaptive.

2 ROBUST AUTOMATION

Simulating large, realistic MEMS models has previously been prohibitive due to both the extensive user effort required to generate models and the computational cost of such analysis. This work robustly automates efficient and accurate numerical modeling of MEMS by eliminating substantially all user interaction required to generate accurate 3D simulations of devices described by photolithographic masks. This automatic procedure allows large, realistic MEMS devices to be accurately simulated in less than one hour on personal computers [4] using the AutoMEMS™ computer aided design (CAD) software.

To accomplish these goals, it is necessary to automate several steps, including: (1) generating a 3D model from photolithographic masks, (2) applying boundary conditions and material properties to the model, (3) meshing the model, and (4) solving the (coupled) PDEs on the model. The steps shown in Figure 1 have been successfully automated in AutoMEMS by using a boundary element method (BEM) [5] discretization and solver. The ovals depict user input, while rectangles are automated steps that do not require any user interaction. Other MEMS simulators [6] [7] [8] use a combination of the finite element method (FEM) and BEM to simulate coupled multi-physics problems but are unable to automate each step. Further, the computational scaling of these previous simulators prevents their use in solving large, realistic MEMS devices.

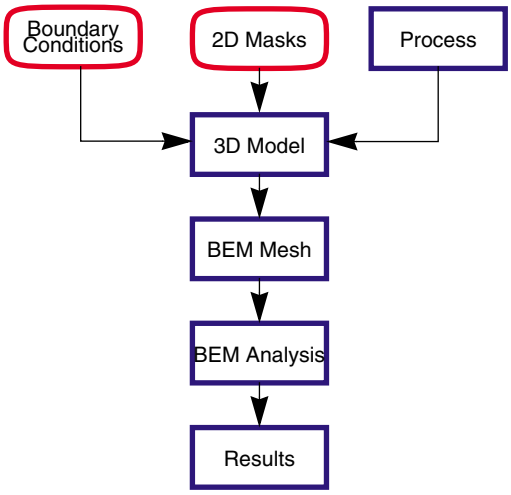

Figure 1: Data flow to simulate a MEMS device.

3 MESHED 3D MODELS FROM LAYOUT

By emulating MEMS fabrication processes, it is possible to generate realistic 3D models [9] [10] [11] [12] [13] of a MEMS device given a 2D mask layout and process description. The model generator [14] used by AutoMEMS creates geometries containing both planar and conformal layers from a standard IC or MEMS mask layout (e.g. CIF or GDSII format) in conjunction with a table-based description of the fabrication. The process information includes deposition order, deposition thickness and material name. There is no limitation on the number of layers, dielectrics or boundary conditions.

4 ADAPTIVE MESHING

BEM calculates very accurate solutions to PDEs. There are essentially only three sources of error in these calculations: (1) numerical integration accuracy and (2) geometric fidelity and (3) mesh discretization. accuracy. Geometric fidelity errors result since the generated AutoMEMS model may not exactly coincide with an actual fabricated device because some effects (e.g. non-homogeneous sputtering, voids, chamfered edges,

curved surfaces, non-vertical edges) are ignored. In this tool, the model may have curved surfaces (approximated as a p-order polynomial). The mesh discretization error is minimized by using adaptive meshing so that the state or gradient may vary nonlinearly (approximated as a p-order polynomial).

To ensure consistently accurate simulations, Coyote has developed an automatic method to mesh and refine the BEM model. This allows the user to specify a desired accuracy (e.g. 2% error on the states/gradients, or a 1% error on the capacitance) and then let the software refine the mesh until this accuracy is achieved.

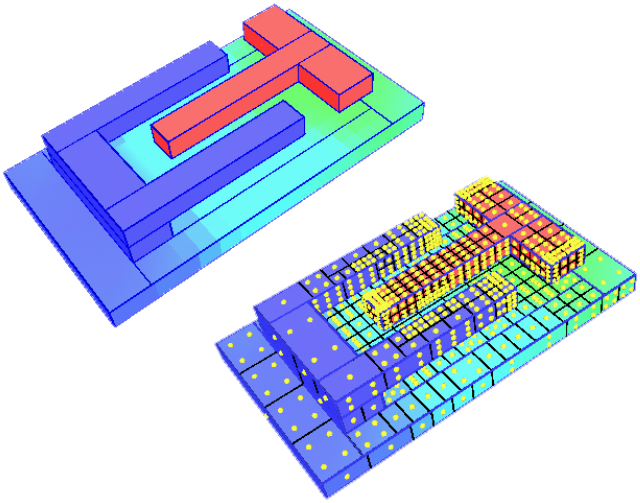

Figure 2: Automatic mesh refinement using error indicators

4.1 Error Indicators

To achieve an automatic mesh refinement, an initial mesh is solved and then post-processing evaluates the states and gradients at points inside each element. For example, if a particular element is a triangle with a linear shape function, then the state/gradient error indicator evaluations [15] on the surface should map closely to a plane face. If they do not, then this indicates that the size of the element should be reduced (h-refinement) or that the polynomial shape function of the element should be increased (p-refinement). If all error indicators show that the elements are well-formed, then the adaptive refinement is finished and the results are reported to the user. If some elements have too large errors, then the model is again refined until all errors are below the user-specified limits.

4.2 Element Refinement

To illustrate the automatic adaptive mesh refinement, consider the initial BEM mesh of a MEMS comb-finger shown in Figure 2. The number of dots on each element indicate the order of the shape function. A single dot indicates that the element is a constant element, 4 dots indicate a bilinear shape function, 9 dots indicate a biquadratic shape function and 16 dots indicate a bicubic shape function.

After 6 iteration cycles, the refined mesh is visible in Figure 2. Note that intuitively the electrostatic flux will vary greatly along the comb-finger tips, and the automatic mesh refinement has captured this behavior by refining these elements. Elements with small errors are not refined. This error indicator driven refinement approach is substantially faster than refining all elements for large models.

5 ELECTROSTATIC SIMULATION

The meshed model of the ADXL50 accelerometer [16] is easily generated using [14]. Note the high fidelity of the model, including the etch-holes.

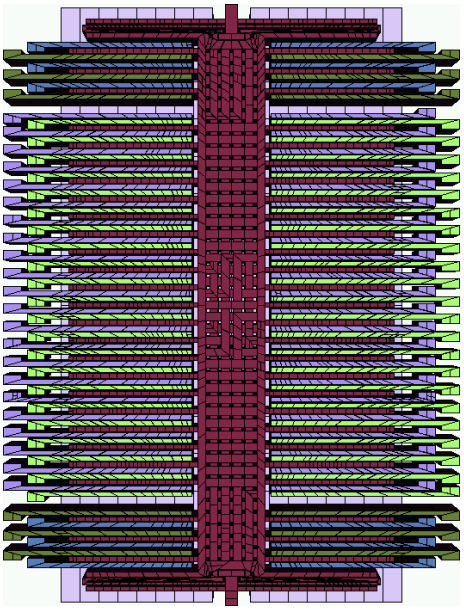

Figure 3: ADXL50 model generated from layout

As in the comb-drive resonator example, it is expected that the electrostatic flux will be concentrated along the edges and corners since physics dictates that there is a flux singularity along sharp edges. The capacitance of the fingers is obtained by integrating the electrostatic flux, therefore it is important to efficiently model the charge-density on the comb-fingers.

Using the automatic error indicators described in “Element Refinement”, the software identifies which elements have large relative errors and only refines those elements. Such a mesh refinement effect can also be cheaply replicated using heuristics based on the variation in normals of adjacent BEM elements. This can efficiently be used to identify panels on edges or corners for refinement. This results in a very efficient method to generate well meshed model as can be seen in Figure 4 which will ensure high-accuracy capacitance simulations.

To reduce the geometric complexity of the model, a 3D field inside a tunnel enclosing the selected elements is solved. The elements enclosed by a tunnel surrounding the driving comb-

fingers are shown in Figure 5. The charge density of the electrostatic solution is shown in Figure 6. Creating the model, specifying the boundary conditions, and running the electrostatic simulation takes approximately 30 seconds on a 500 MHz PentiumIII processor.

6 COUPLED ELECTRO-MECHANICAL SIMULATIONS

By creating both electrostatic and mechanical boundary conditions, it is possible to easily create coupled electromechanical MEMS simulations. Typically an electrostatic simulation is first solved, and the solved electrostatic gradient is used to calculate a mechanical traction. Using this inherited traction, the mechanical model is solved to obtain the new mechanical state (i.e. displacement). The electrostatic model typically inherits the new displacement and the cycle iterates until a suitable successful convergence criteria (e.g. small change in displacement, small change in capacitance) or failure criteria (e.g. collision, number of iterations, amount of cputime) are satisfied. The AutoMEMS GUI uses “wizards” to help create these coupling scenarios and convergence criteria.

Figure 7 depicts an example of a series of coupled simulation steps of a torsional MEMS micromirror. The driving electrodes create an electrostatic torque deflecting the paddle. A typical use of numerical simulations is to calculate the electrostatic torque as a function of voltage and deflection as well as the pull-in voltage. Creating the model, specifying the initial boundary conditions, specifying the convergence criteria and running the coupled simulations near the pull-in voltage takes approximately 1 hour on a 500 MHz PentiumIII processor.

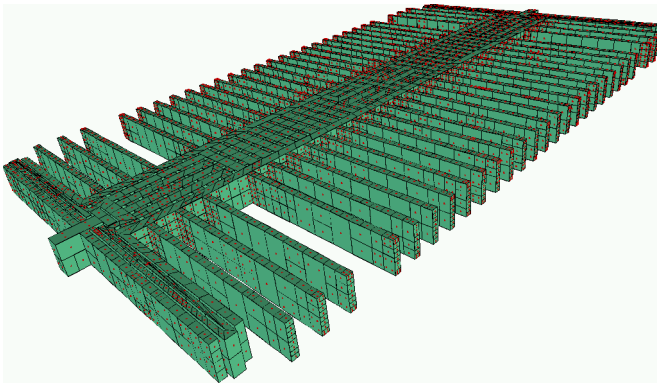

Figure 4: Adaptively refined ADXL50 moving proof mass

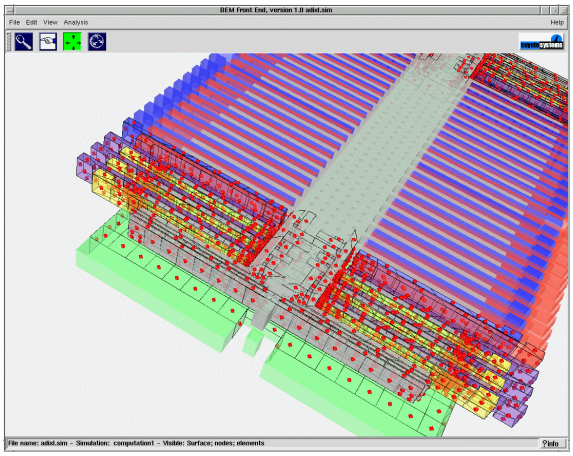

Figure 5: Tunnel around driving comb-fingers

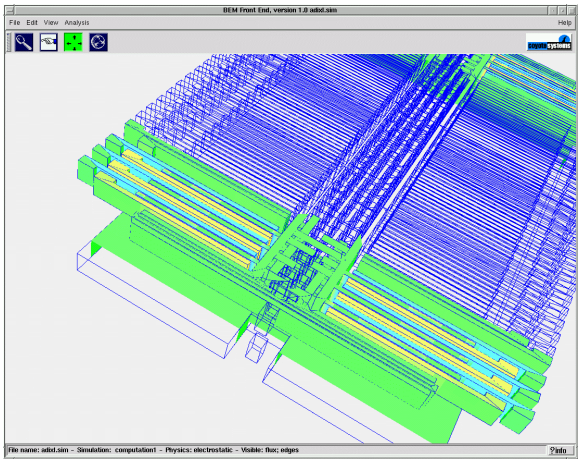

Figure 6: Calculated charge on the drive comb-fingers

REFERENCES

[1] P. Ljung, “Sequential solutions of field equations using a single BEM model”, Solid State Sensor and Actuator Workshop, Hilton Head Island, SC, June 3-6, pp. 117-122, 1996

[2] M. Bächtold, Efficient 3D Computation of Electrostatic Fields and Forces in Microsystems, ETH Zurich, ISBN 3-907574-17-6, Zurich, Switzerland, 1997

[3] P. Ljung, M. Bachtold, M. Spasojevic, “Analysis of Realistic Large MEMS Devices”, Computer Modeling in Engineering & Sciences, vol. 1, no. 1, pp. 21-30, Jan. 2000

[4] Coyote Systems Inc., Users Guide reference manuals, http://www.coyotesystems.com/pdf/autobem.pdf, 1999

[5] C.A. Brebbia, J.C.F. Telles, L.C. Wrobel, Boundary Element Techniques, Theory and Application in Engineering, Springer Verlag Berlin, 1984

[6] J. Gilbert, R. Legtenberg, S. Senturia, “3D coupled electro-mechanics for MEMS: Applications of CoSolve-EM”, Proc. MEMS 95, Amsterdam, NL, Feb., pp. 122-127, 1995

[7] Intellisense Corp., “IntelliCad” and “IntelliFab” reference manuals, Wilmington, MA, 1997

[8] Integrated Systems Engineering AG, “Modeling of Semiconductor Technology, Devices and Systems” reference manuals, Zurich, 1997

[9] J.P. Elliot, G.A. Allan, A.J. Walton, “The Automatic Generation of Conformal 3D Data for Interconnect Capacitance Simulation”, Proc. VLSI Multi-Level Interconnect Conf, pp. 664-666, 1995

[10] E.W. Scheckler, A.R Neureuther, “Coupling Model and Algorithms for 3D Topography Simulation: Plasma Etching, Ion Milling and Deposition in SAMPLE-3D”, Proc. Workshop on Num. Mod. of Processes and Devices for Integrated Circuits, pp. 9-14, 1992

[11] J. Pelka, “Three-Dimensional Simulation of Ion-Enhanced Dry-Etch Processes”, Microelectronic Engineering. , vol. 14, pp. 269-281, 1991

[12] E. Strasser, S. Selberherr, “Algorithms and Models for Cellular Based Topography Simulation”, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 14(9), pp.1104-1114, 1995

[13] P. Ljung, M. Bächtold, “Automatic Model Generation and Analysis for MEMS”, Proc. ASME’98

[14] M. Spasojevic, P. Ljung, M. Bachtold, “Automated MEMS Model Generation from Layout”, Proc. MSM, San Diego, March 2000

[15] M. Bächtold, J.G. Korvink, H.Baltes, “An Error Indicator and Automatic Adaptive Meshing for 3D Electrostatic Boundary Element Simulations”, Boundary Elements XIX, Southampton, UK, pp. 709-718, 1997

[16] Analog Devices Inc., 1999

[17] M. Bächtold, “Efficient 3D computation of electrostatic fields and forces in microsystems”, Ph.D. thesis, ETH Zurich, 1997

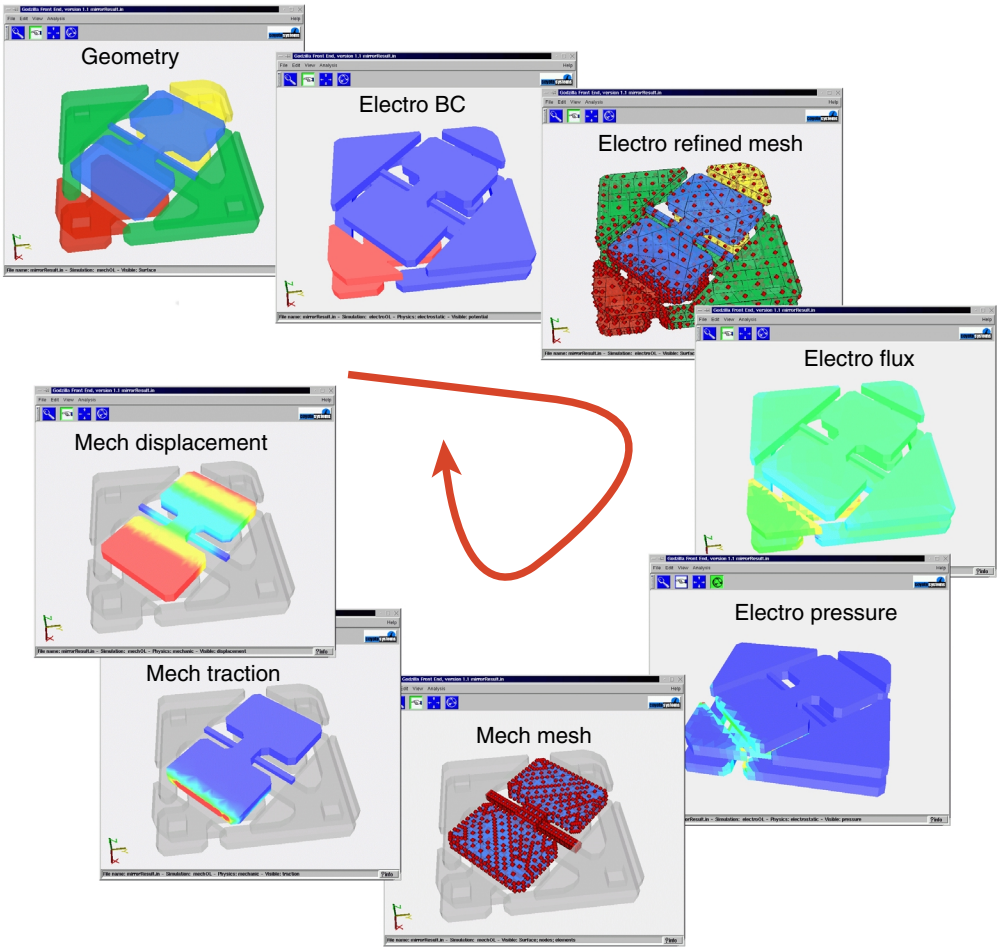

Figure 7: Coupled MEMS micromirror simulations