Computer Aided Design of Micro-Electro-Mechanical Systems

From Energy Losses to Dick Tracy Watches

D. Bindel

Courant Institute for Mathematical Sciences
New York University

Rice University, 14 Jan 2008

The Computational Science Picture

- Application modeling
  - Disk resonator
  - Beam resonator
  - Shear ring resonator, checkerboard, ...
- Mathematical analysis
  - Physical modeling and finite element technology
  - Structured eigenproblems and reduced-order models
  - Parameter-dependent eigenproblems
- Software engineering
  - HiQLab
  - SUGAR
  - FEAPMEX / MATFEAP

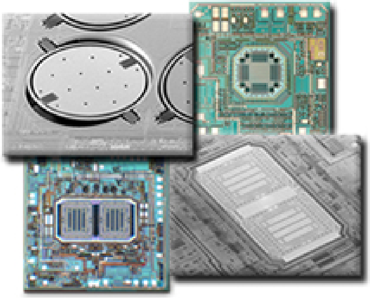
The Computational Science Picture

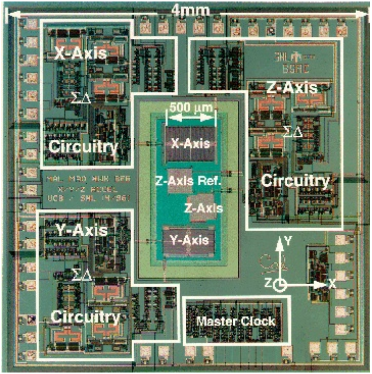
- Application modeling
  - Disk resonator
  - Beam resonator
  - Shear ring resonator, checkerboard, ...
- Mathematical analysis
  - Physical modeling and finite element technology
  - Structured eigenproblems and reduced-order models
  - Parameter-dependent eigenproblems
- Software engineering
  - HiQLab
  - SUGAR
  - FEAPMEX / MATFEAP

Outline

Outline

What are MEMS?

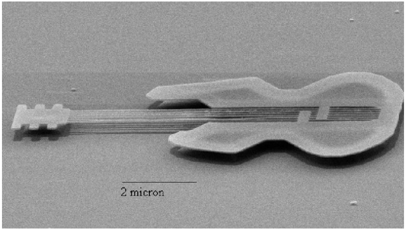


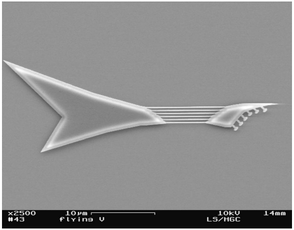


MEMS Basics

- Micro-Electro-Mechanical Systems
  - Chemical, fluid, thermal, optical (MECFTOMS?)
- Applications:
  - Sensors (inertial, chemical, pressure)
  - Ink jet printers, biolab chips
  - Radio devices: cell phones, inventory tags, pico radio
- Use integrated circuit (IC) fabrication technology
- Tiny, but still classical physics

Resonant RF MEMS

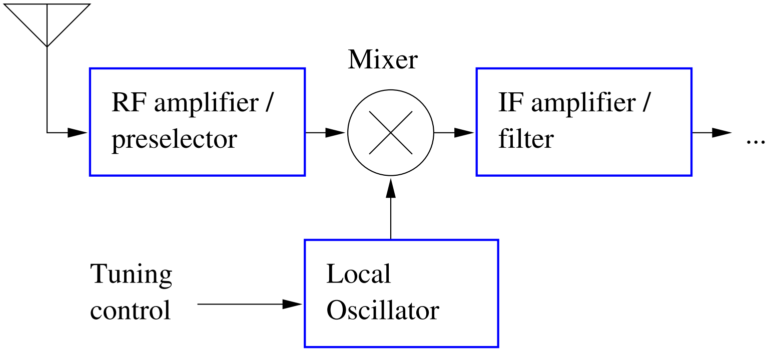




Microguitars from Cornell University (1997 and 2003)

- MHz-GHz mechanical resonators
- Favorite application: radio on chip
- Close second: really high-pitch guitars

The Mechanical Cell Phone

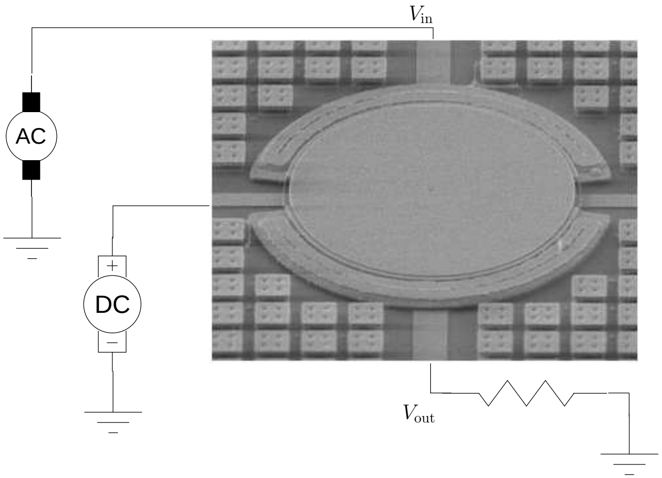


- Your cell phone has many moving parts!
- What if we replace them with integrated MEMS?

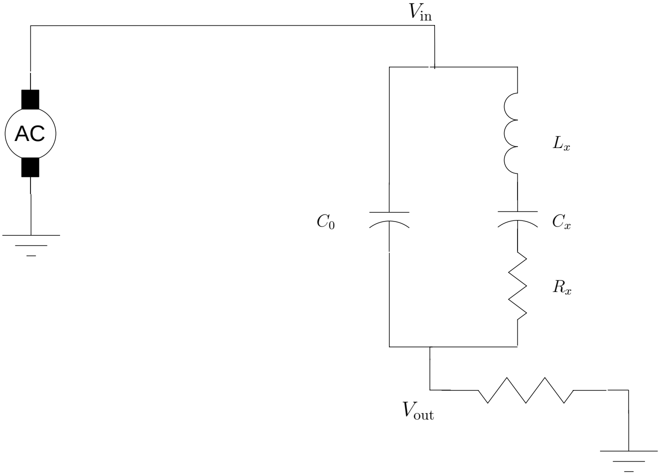
Ultimate Success

"Calling Dick Tracy!"

Disk Resonator



Disk Resonator



Electromechanical Model

Assume time-harmonic steady state, no external forces:

$$\begin{bmatrix} i\omega C + G & i\omega B \\ -B^T & \tilde{K} - \omega^2 M \end{bmatrix} \begin{bmatrix} \delta \hat{V} \\ \delta \hat{u} \end{bmatrix} = \begin{bmatrix} \delta \hat{I}_{\mathrm{external}} \\ 0 \end{bmatrix}$$

Eliminate the mechanical terms:

$$\begin{aligned} Y(\omega)\,\delta\hat{V} &= \delta\hat{I}_{\mathrm{external}} \\ Y(\omega) &= i\omega C + G + i\omega H(\omega) \\ H(\omega) &= B^T(\tilde{K} - \omega^2 M)^{-1} B \end{aligned}$$

Goal: Understand electromechanical piece ($i\omega H(\omega)$).

- As a function of geometry and operating point
- Preferably as a simple circuit

Damping and Q

Designers want high *quality of resonance* (Q)

- Dimensionless damping in a one-dof system

$$\frac{d^2 u}{dt^2} + Q^{-1} \frac{du}{dt} + u = F(t)$$

- For a resonant mode with frequency $\omega \in \mathbb{C}$:

$$Q := \frac{|\omega|}{2 \operatorname{Im}(\omega)} = \frac{\text{Stored energy}}{\text{Energy loss per radian}}$$

To understand Q, we need damping models!

The Designer's Dream

Ideally, would like

- Simple models for behavioral simulation
- Parameterized for design optimization
- Including all relevant physics
- With reasonably fast and accurate set-up

We aren't there yet.

Outline

Enter HiQLab

- Existing codes do not compute quality factors
- ... and awkward to prototype new solvers
- ... and awkward to programmatically define meshes
- So I wrote a new finite element code: HiQLab

Heritage of HiQLab

SUGAR: SPICE for the MEMS world

- System-level simulation using modified nodal analysis
- Flexible device description language
- C core with MATLAB interfaces and numerical routines

FEAPMEX: MATLAB + a finite element code

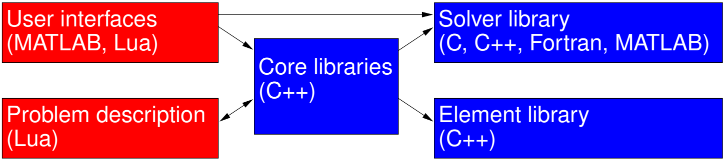
- MATLAB interfaces for steering, testing solvers, running parameter studies
- Time-tested finite element architecture
- But old F77, brittle in places

Other Ingredients

"Lesser artists borrow. Great artists steal."
– Picasso, Dali, Stravinsky?

- Lua: www.lua.org
  - Evolved from simulator data languages (DEL and SOL)
  - Pascal-like syntax fits on one page; complete language description is 21 pages
  - Fast, freely available, widely used in game design
- MATLAB: www.mathworks.com
  - "The Language of Technical Computing"
  - Good sparse matrix support
  - Star-P: http://www.interactivesupercomputing.com/
- Standard numerical libraries: ARPACK, UMFPACK

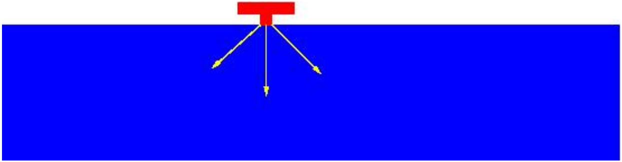
HiQLab Structure



- Standard finite element structures + some new ideas
- Full scripting language for mesh input
- Callbacks for boundary conditions, material properties
- MATLAB interface for quick algorithm prototyping
- Cross-language bindings are automatically generated

Outline

Damping Mechanisms



Possible loss mechanisms:

- Fluid damping
- Material losses
- Thermoelastic damping
- Anchor loss

Model substrate as semi-infinite with a

Perfectly Matched Layer (PML).

Perfectly Matched Layers

- Complex coordinate transformation
- Generates a “perfectly matched” absorbing layer
- Idea works with general linear wave equations
  - Electromagnetics (Berengér, 1994)
  - Quantum mechanics – *exterior complex scaling* (Simon, 1979)
  - Elasticity in standard finite element framework (Basu and Chopra, 2003)

Model Problem

- Domain: $x \in [0, \infty)$
- Governing eq:

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0$$

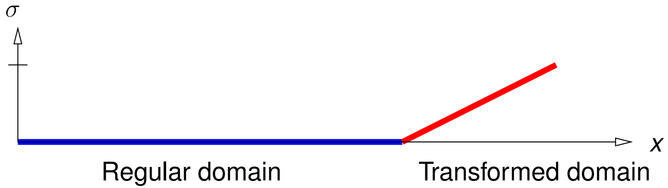
- Fourier transform:

$$\frac{d^2 \hat{u}}{dx^2} + k^2 \hat{u} = 0$$

- Solution:

$$\hat{u} = c_{\mathrm{out}} e^{-ikx} + c_{\mathrm{in}} e^{ikx}$$

Model with Perfectly Matched Layer

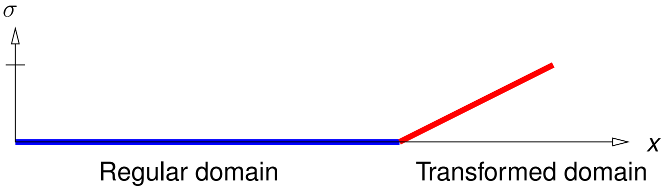


$$\frac{d\tilde{x}}{dx} = \lambda(x) \text{ where } \lambda(s) = 1 - i\sigma(s)$$

$$\frac{d^2\hat{u}}{d\tilde{x}^2} + k^2\hat{u} = 0$$

$$\hat{u} = c_{\text{out}} e^{-ik\tilde{x}} + c_{\text{in}} e^{ik\tilde{x}}$$

Model with Perfectly Matched Layer



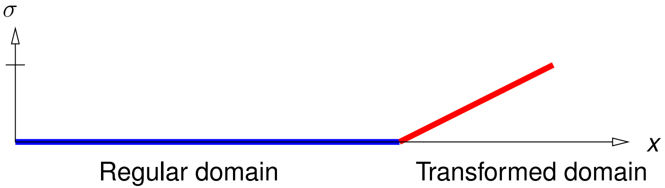
$$\frac{d\tilde{x}}{dx} = \lambda(x) \text{ where } \lambda(s) = 1 - i\sigma(s),$$

$$\frac{1}{\lambda}\frac{d}{dx}\left(\frac{1}{\lambda}\frac{d\hat{u}}{dx}\right) + k^2\hat{u} = 0$$

$$\hat{u} = c_{\mathrm{out}} e^{-ikx - k\Sigma(x)} + c_{\mathrm{in}} e^{ikx + k\Sigma(x)}$$

$$\Sigma(x) = \int_0^x \sigma(s)\, ds$$

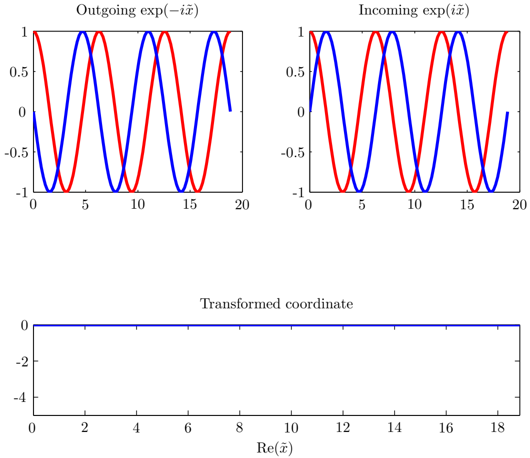
Model with Perfectly Matched Layer



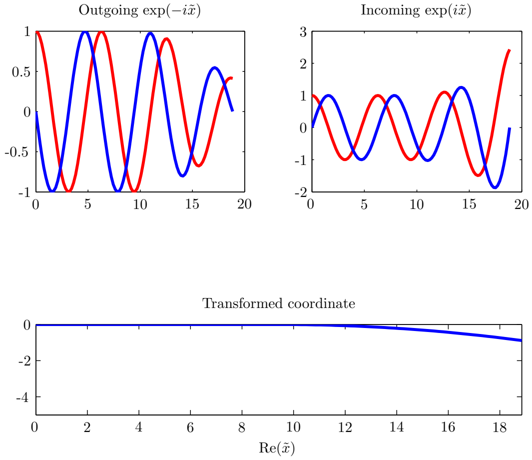
If solution clamped at $x = L$ then

$$\frac{c_{\text{in}}}{c_{\text{out}}} = O(e^{-k\gamma}) \text{ where } \gamma = \Sigma(L) = \int_0^L \sigma(s)\, ds$$

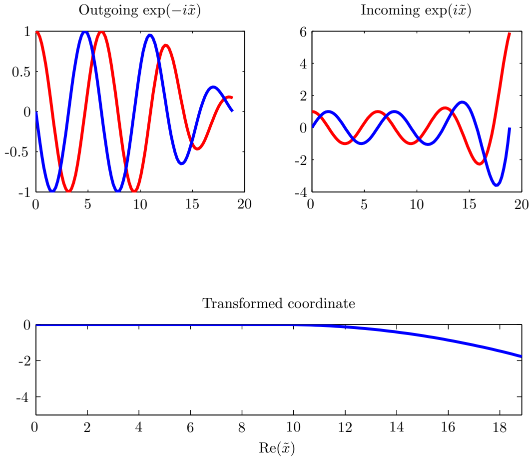
Model Problem Illustrated



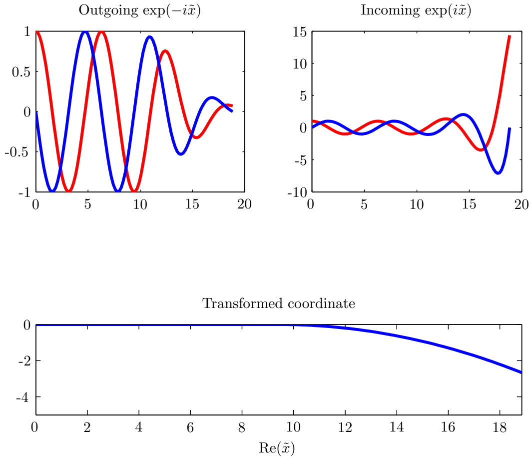
Model Problem Illustrated



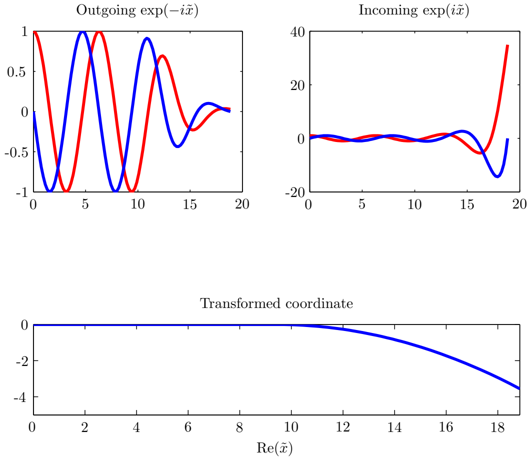
Model Problem Illustrated



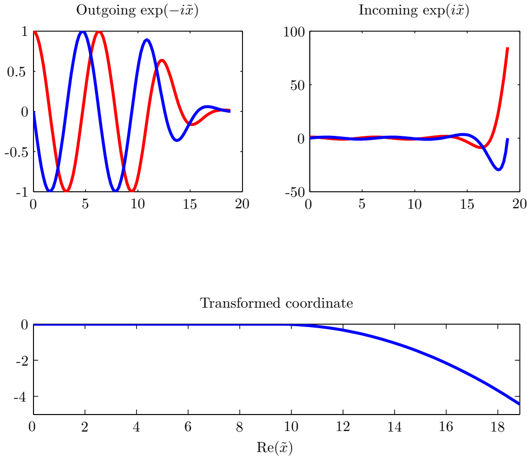
Model Problem Illustrated



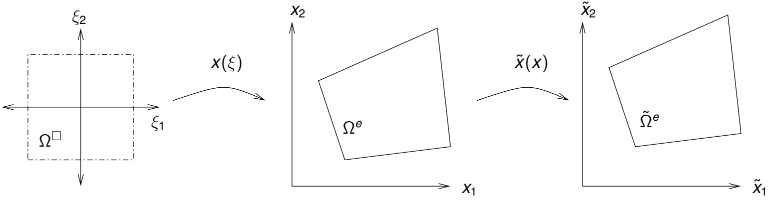
Model Problem Illustrated



Model Problem Illustrated



Finite Element Implementation



- Combine PML and isoparametric mappings

$$\mathbf{k}^e = \int_{\Omega^\square} \tilde{\mathbf{B}}^T \mathbf{D} \tilde{\mathbf{B}} \tilde{J} \, d\Omega^\square$$

$$\mathbf{m}^e = \int_{\Omega^\square} \rho \mathbf{N}^T \mathbf{N} \tilde{J} \, d\Omega^\square$$

- Matrices are *complex symmetric*

Eigenvalues and Model Reduction

Want to know about the transfer function $H(\omega)$:

$$H(\omega) = B^T (K - \omega^2 M)^{-1} B$$

Can either

- Locate poles of H (eigenvalues of (K, M))
- Plot H in a frequency range (Bode plot)

Usual tactic: subspace projection

- Build an Arnoldi basis V for a Krylov subspace $\mathcal{K}_n$
- Compute with much smaller $V^* K V$ and $V^* M V$

Can we do better?

Variational Principles

- Variational form for complex symmetric eigenproblems:
  - Hermitian (Rayleigh quotient):

$$\rho(v) = \frac{v^* K v}{v^* M v}$$

  - Complex symmetric (modified Rayleigh quotient):

$$\theta(v) = \frac{v^T K v}{v^T M v}$$

- First-order accurate eigenvectors $\implies$ Second-order accurate eigenvalues.
- Key: relation between left and right eigenvectors.

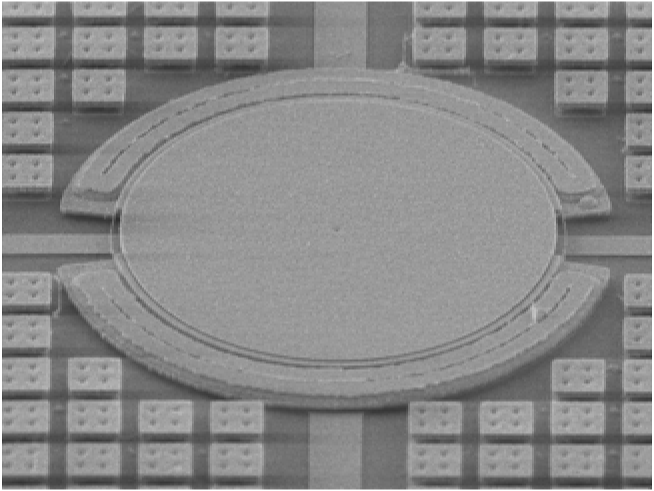
Accurate Model Reduction

- Build new projection basis from V:

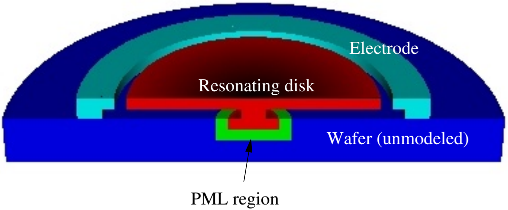
$$W = \text{orth}[\text{Re}(V), \text{Im}(V)]$$

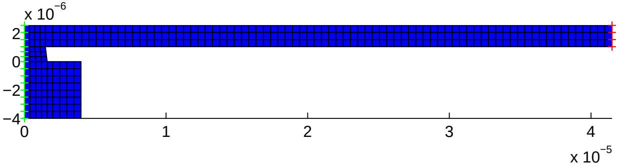
- span(W) contains both $\mathcal{K}_n$ and $\bar{\mathcal{K}}_n$
  $\implies$ double digits correct vs. projection with V
- W is a real-valued basis
  $\implies$ projected system is complex symmetric

Disk Resonator Simulations



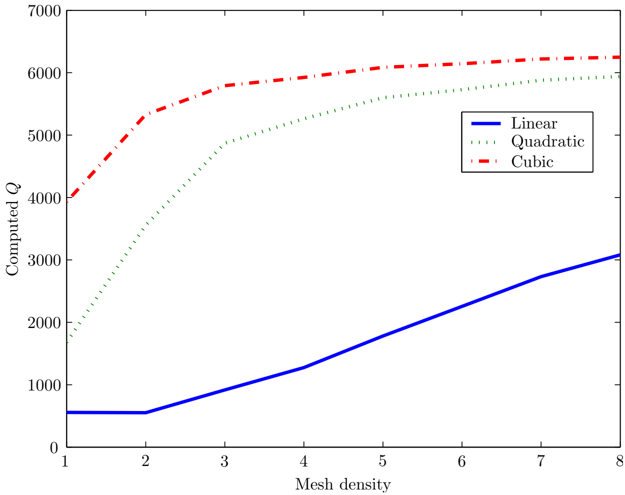
Disk Resonator Mesh





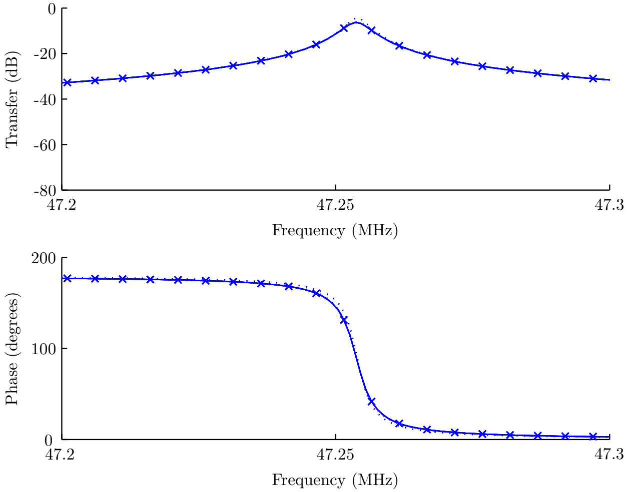
- Axisymmetric model with bicubic mesh
- About 10K nodal points in converged calculation

Mesh Convergence



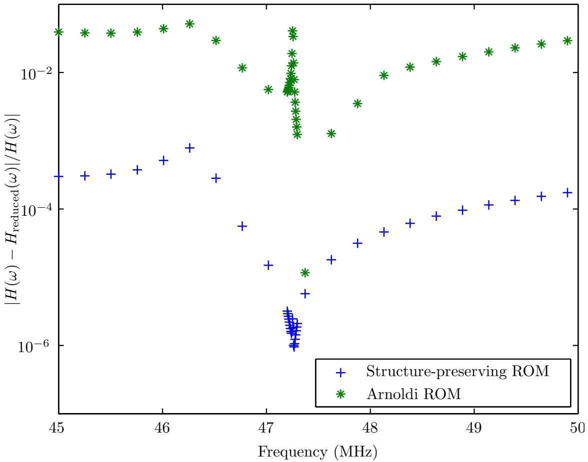
Cubic elements converge with reasonable mesh density

Model Reduction Accuracy



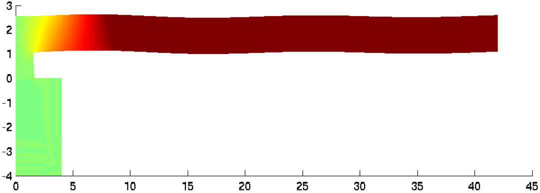
Results from ROM (solid and dotted lines) nearly indistinguishable from full model (crosses)

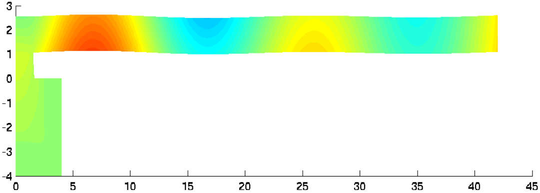
Model Reduction Accuracy



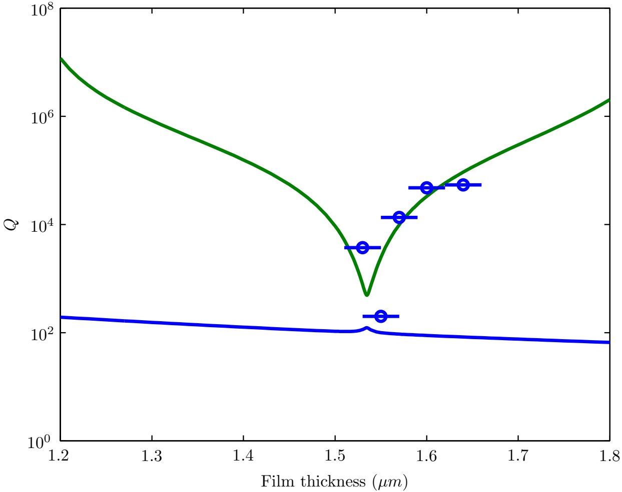
Preserve structure $\Longrightarrow$
get twice the correct digits

Response of the Disk Resonator



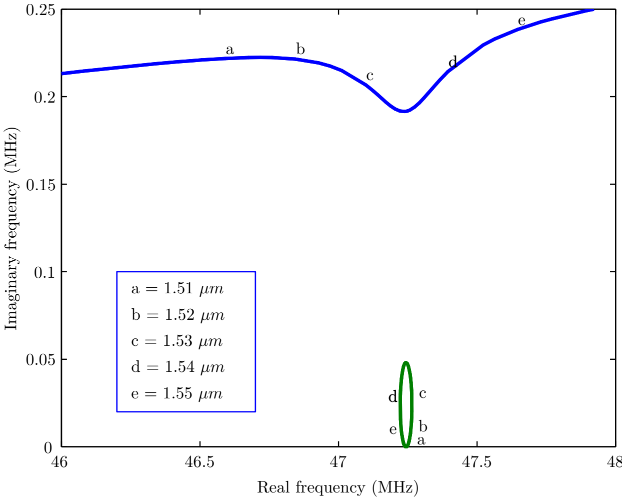


Variation in Quality of Resonance



Simulation and lab measurements vs. disk thickness

Explanation of Q Variation

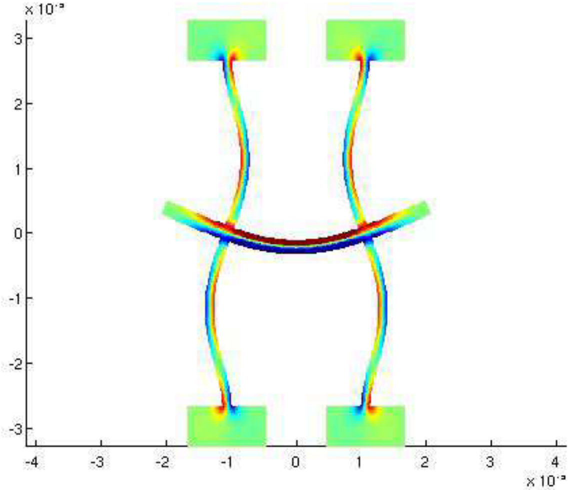


Interaction of two nearby eigenmodes

Outline

1. Resonant MEMS and models
2. HiQLab
3. Anchor losses and disk resonators
4. **Thermoelastic losses and beam resonators**
5. Conclusion

Thermoelastic Damping (TED)



Thermoelastic Damping (TED)

u is displacement and $T = T_0 + \theta$ is temperature

$$\begin{aligned} \sigma &= C\epsilon - \beta\theta 1 \\ \rho\ddot{u} &= \nabla \cdot \sigma \\ \rho c_v \dot{\theta} &= \nabla \cdot (\kappa \nabla \theta) - \beta T_0 \operatorname{tr}(\dot{\epsilon}) \end{aligned}$$

- Coupling between temperature and volumetric strain:
  - Compression and expansion $\implies$ heating and cooling
  - Heat diffusion $\implies$ mechanical damping
  - Not often an important factor at the macro scale
  - Recognized source of damping in microresonators
- Zener: semi-analytical approximation for TED in beams
- We consider the fully coupled system

Nondimensionalized Equations

Continuum equations:

$$\begin{aligned} \sigma &= \hat{C}\epsilon - \xi\theta 1 \\ \ddot{u} &= \nabla \cdot \sigma \\ \dot{\theta} &= \eta\nabla^2\theta - \mathrm{tr}(\dot{\epsilon}) \end{aligned}$$

Discrete equations:

$$\begin{aligned} M_{uu}\ddot{u} + K_{uu}u &= \xi K_{u\theta}\theta + f \\ C_{\theta\theta}\ddot{\theta} + \eta K_{\theta\theta}\theta &= -C_{\theta u}\dot{u} \end{aligned}$$

- Micron-scale poly-Si devices: ξ and η are $\sim 10^{-4}$.
- Linearize about $\xi = 0$

Perturbative Mode Calculation

Discretized mode equation:

$$\begin{aligned}(-\omega^2 M_{uu} + K_{uu})u &= \xi K_{u\theta}\theta \\ (i\omega C_{\theta\theta} + \eta K_{\theta\theta})\theta &= -i\omega C_{\theta u} u\end{aligned}$$

First approximation about $\xi = 0$:

$$\begin{aligned}(-\omega_0^2 M_{uu} + K_{uu})u_0 &= 0 \\ (i\omega_0 C_{\theta\theta} + \eta K_{\theta\theta})\theta_0 &= -i\omega_0 C_{\theta u} u_0\end{aligned}$$

First-order correction in ξ:

$$-\delta(\omega^2) M_{uu} u_0 + (-\omega_0^2 M_{uu} + K_{uu})\delta u = \xi K_{u\theta}\theta_0$$

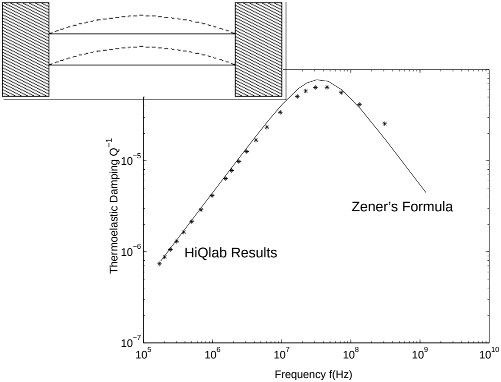
Multiply by u_0^T:

$$\delta(\omega^2) = -\xi \left(\frac{u_0^T K_{u\theta}\theta_0}{u_0^T M_{uu} u_0} \right)$$

Zener's Model

1. Clarence Zener investigated TED in late 30s-early 40s.
2. Model for beams common in MEMS literature.
3. "Method of orthogonal thermodynamic potentials" == perturbation method + a variational method.

Comparison to Zener's Model



- Comparison of fully coupled simulation to Zener approximation over a range of frequencies
- Real and imaginary parts after first-order correction agree to about three digits with Arnoldi

Outline

Onward!

What about:

- Modeling more geometrically complex devices?
- Modeling general dependence on geometry?
- Modeling general dependence on operating point?
- Computing nonlinear dynamics?
- Digesting all this to help designers?

Future Work

- Code development
  - Structural elements and elements for different physics
  - Design and implementation of parallelized version
- Theoretical analysis
  - More damping mechanisms
  - Sensitivity analysis and variational model reduction
- Application collaborations
  - Use of nonlinear effects (quasi-static and dynamic)
  - New designs (e.g. internal dielectric drives)
  - Continued experimental comparisons

Conclusions

- RF MEMS are a great source of problems
  - Interesting applications
  - Interesting physics (and not altogether understood)
  - Interesting computing challenges

`http://www.cims.nyu.edu/~dbindel`

Concluding Thoughts

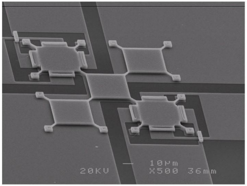
The difference between art and science is that science is what we understand well enough to explain to a computer. Art is everything else.

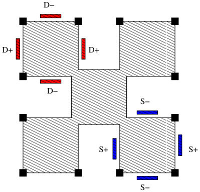
Donald Knuth

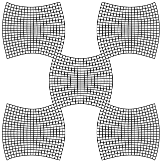
The purpose of computing is insight, not numbers.

Richard Hamming

Checkerboard Resonator





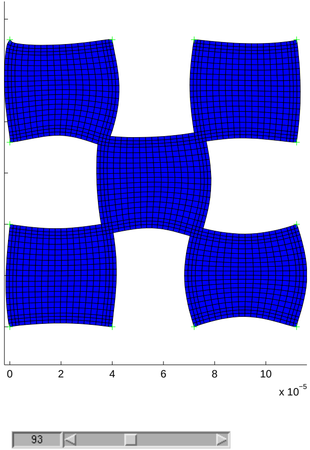


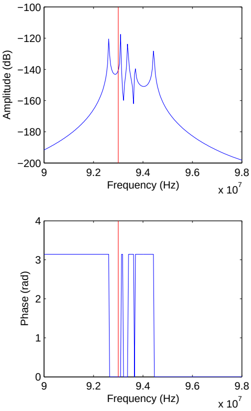
- Anchored at outside corners
- Excited at northwest corner
- Sensed at southeast corner
- Surfaces move only a few nanometers

Checkerboard Model Reduction

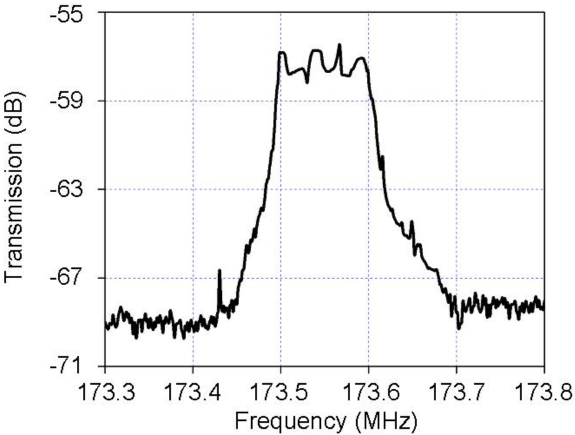
- Finite element model: $N = 2154$
  - Expensive to solve for every $H(\omega)$ evaluation!
- Build a reduced-order model to approximate behavior
  - Reduced system of 80 to 100 vectors
  - Evaluate $H(\omega)$ in milliseconds instead of seconds
  - Without damping: standard Arnoldi projection
  - With damping: Second-Order ARnoldi (SOAR)

Checkerboard Simulation





Checkerboard Measurement



S. Bhave, MEMS 05

Contributions

- Built predictive model used to design checkerboard
- Used model reduction to get thousand-fold speedup – fast enough for interactive use

General Picture

If $w^*A = 0$ and $Av = 0$ then

$$\delta(w^*Av) = w^*(\delta A)v$$

This implies

- If $A = A(\lambda)$ and $w = w(v)$, have

$$w^*(v)A(\rho(v))v = 0.$$

  ρ stationary when $(\rho(v), v)$ is a nonlinear eigenpair.
- If $A(\lambda, \xi)$ and w_0^* and v_0 are null vectors for $A(\lambda_0, \xi_0)$,

$$w_0^*(A_\lambda \delta\lambda + A_\xi \delta\xi)v_0 = 0.$$

Electromechanical Model

Kirchoff's current law and balance of linear momentum:

$$\begin{aligned} \frac{d}{dt}\left(C(u)V\right) + GV &= I_{\text{external}} \\ Mu_{tt} + Ku - \nabla_u \left(\frac{1}{2} V^* C(u) V\right) &= F_{\text{external}} \end{aligned}$$

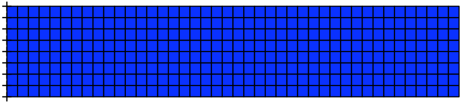
Linearize about static equilibium (V_0, u_0):

$$\begin{aligned} C(u_0)\,\delta V_t + G\,\delta V + \left(\nabla_u C(u_0) \cdot \delta u_t\right) V_0 &= \delta I_{\text{external}} \\ M\,\delta u_{tt} + \tilde{K}\,\delta u + \nabla_u \left(V_0^* C(u_0)\,\delta V\right) &= \delta F_{\text{external}} \end{aligned}$$

where

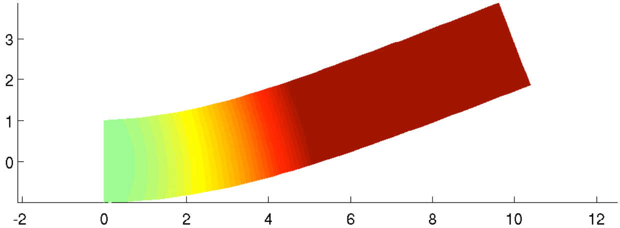
$$\tilde{K} = K - \frac{1}{2}\frac{\partial^2}{\partial u^2}\left(V_0^* C(u_0) V_0\right)$$

HiQLab's Hello World



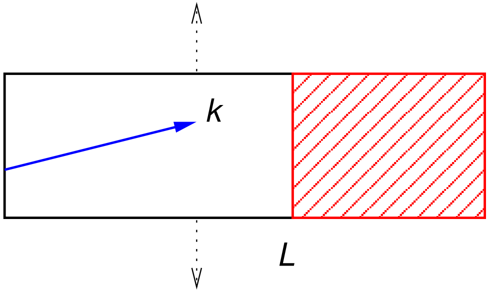
```
mesh  = Mesh:new(2)
mat = make_material('silicon2', 'planestrain')
mesh:blocks2d( { 0, l }, { -w/2.0, w/2.0 },
               mat )

mesh:set_bc(function(x,y)
  if x == 0 then return 'uu', 0, 0; end
end)
```

HiQLab's Hello World




```
>> mesh = Mesh_load('beammesh.lua');
>> [M,K] = Mesh_assemble_mk(mesh);
>> [V,D] = eigs(K,M, 5, 'sm');
>> opt.axequal = 1; opt.deform = 1;
>> Mesh_scale_u(mesh, V(:,1), 2, 1e-6);
>> plotfield2d(mesh, opt);
```

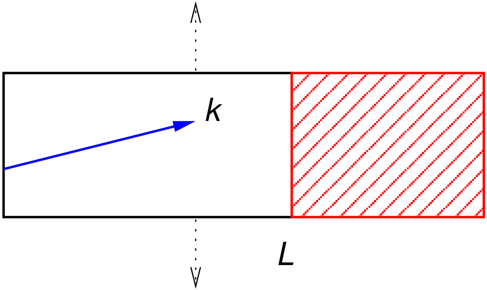
Continuum 2D model problem



$$\lambda(x) = \begin{cases} 1 - i\beta|x - L|^p, & x > L \\ 1 & x \leq L. \end{cases}$$

$$\frac{1}{\lambda}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\frac{\partial u}{\partial x}\right) + \frac{\partial^2 u}{\partial y^2} + k^2 u = 0$$

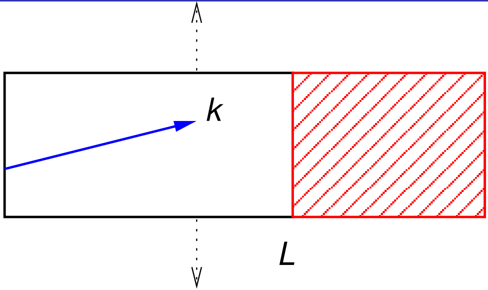
Continuum 2D model problem



$$\lambda(x) = \begin{cases} 1 - i\beta|x-L|^p, & x > L \\ 1 & x \leq L. \end{cases}$$

$$\frac{1}{\lambda}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\frac{\partial u}{\partial x}\right) - k_y^2 u + k^2 u = 0$$

Continuum 2D model problem

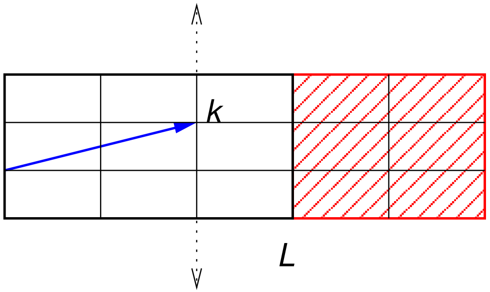


$$\lambda(x) = \begin{cases} 1 - i\beta|x-L|^p, & x > L \\ 1 & x \leq L. \end{cases}$$

$$\frac{1}{\lambda}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\frac{\partial u}{\partial x}\right) + k_x^2 u = 0$$

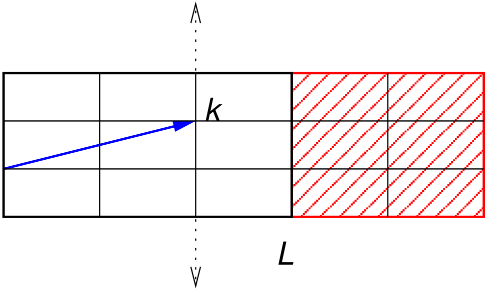
1D problem, reflection of $O(e^{-k_x \gamma})$

Discrete 2D model problem



- Discrete Fourier transform in y
- Solve numerically in x
- Project solution onto infinite space traveling modes
- Extension of Collino and Monk (1998)

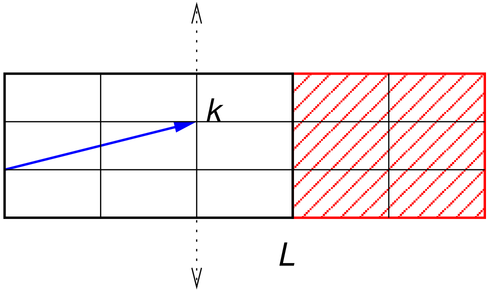
Nondimensionalization



$$\lambda(x) = \begin{cases} 1 - i\beta|x - L|^p, & x > L \\ 1 & x \le L. \end{cases}$$

Rate of stretching:	βh^p
Elements per wave:	$(k_x h)^{-1}$ and $(k_y h)^{-1}$
Elements through the PML:	N

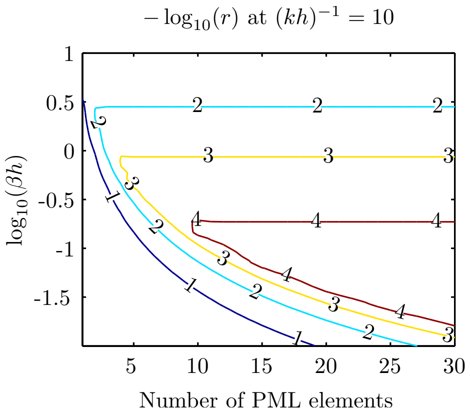
Nondimensionalization



$$\lambda(x) = \begin{cases} 1 - i\beta|x-L|^p, & x > L \\ 1 & x \leq L. \end{cases}$$

Rate of stretching:	βh^p
Elements per wave:	$(k_x h)^{-1}$ and $(k_y h)^{-1}$
Elements through the PML:	N

Discrete reflection behavior



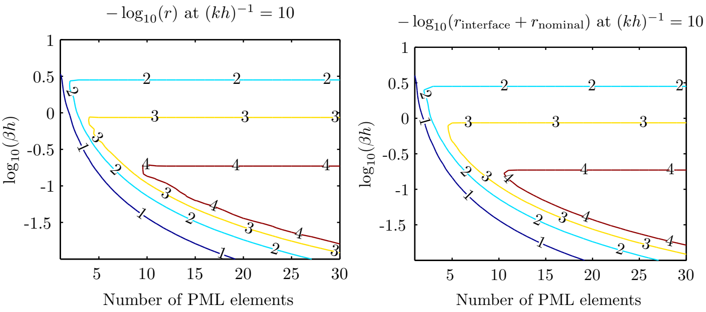
Quadratic elements, $p = 1$, $(k_x h)^{-1} = 10$

Discrete reflection decomposition

Model discrete reflection as two parts:

- Far-end reflection (clamping reflection)
  - Approximated well by continuum calculation
  - Grows as $(k_x h)^{-1}$ grows
- Interface reflection
  - Discrete effect: mesh does not resolve decay
  - Does not depend on N
  - Grows as $(k_x h)^{-1}$ shrinks

Discrete reflection behavior



Quadratic elements, $p = 1$, $(k_x h)^{-1} = 10$

- Model does well at predicting actual reflection
- Similar picture for other wavelengths, element types, stretch functions

Choosing PML parameters

- Discrete reflection dominated by
  - Interface reflection when k_x large
  - Far-end reflection when k_x small
- Heuristic for PML parameter choice
  - Choose an acceptable reflection level
  - Choose β based on interface reflection at k_x^{max}
  - Choose length based on far-end reflection at k_x^{min}