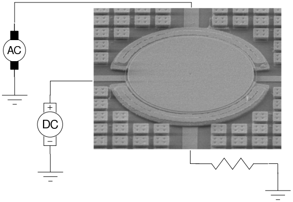
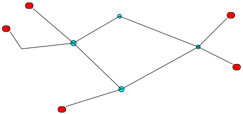
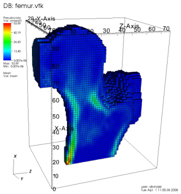
Computational Science and Engineering







Application modeling

Analysis ⟷ Algorithms and software

$\rho \ddot{u} = \nabla \cdot \sigma$

$Ax = b$

$Ax = \lambda x$

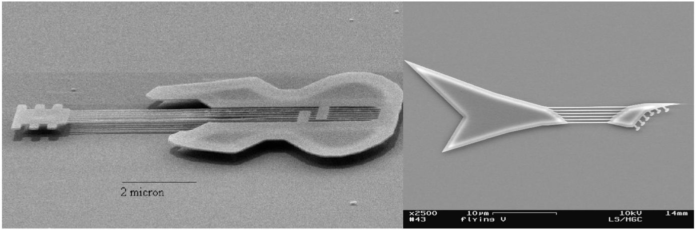
HiQLab, Matscat, ...

Model reduction methods

Structured linear solvers

Today: Two applications and an algorithm idea.

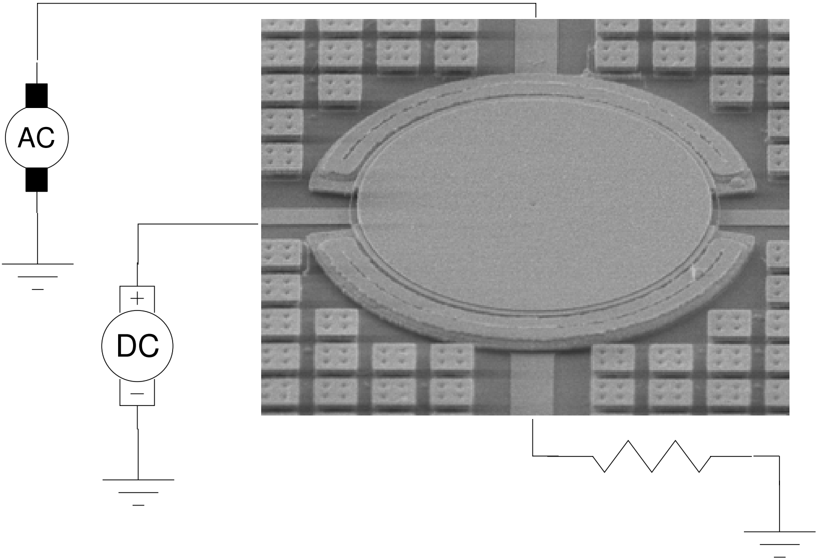
Application: Radio-Frequency MEMS



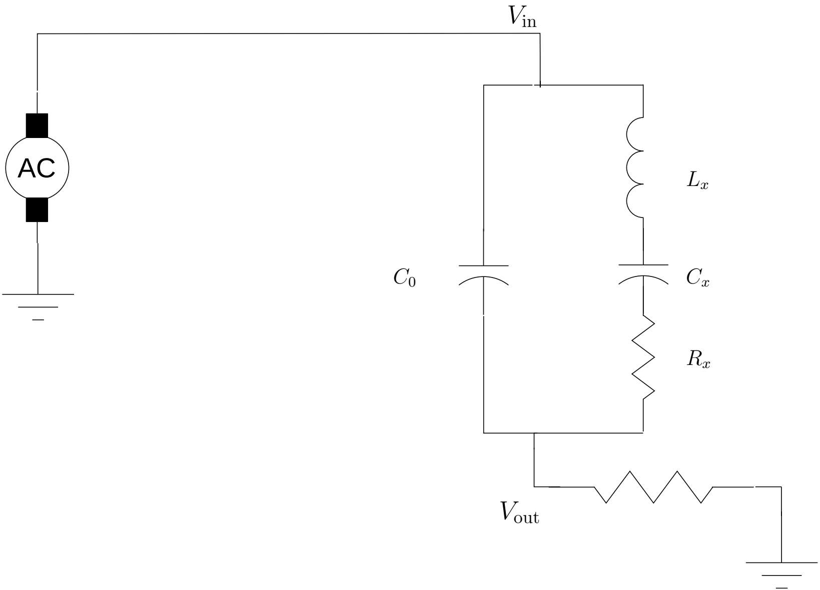
Microguitars from Cornell University (1997 and 2003)

- MHz-GHz mechanical resonators
- Impact: smaller, lower-power cell phones
- Also useful for sensors (inertial, chemical)
- ... and really high-pitch guitars

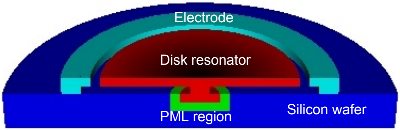
Modeling a Disk Resonator

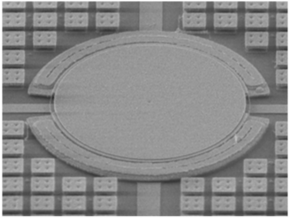


Modeling a Disk Resonator



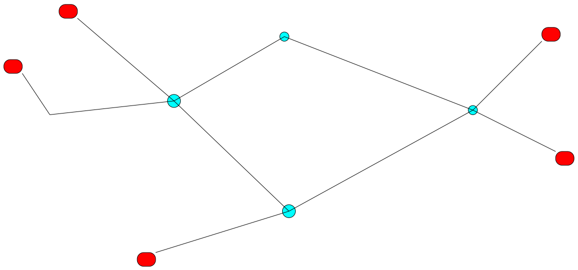
Modeling a Ringing Disk





- At what frequencies does this vibrate?
- How quickly is the ringing damped?
- What about errors (in numerics or fabrication)?
- How do we answer these questions *fast*?

Application: Network Tomography



Measure few paths between n hosts in network to infer

- Path properties
- Link properties
- Network topology?

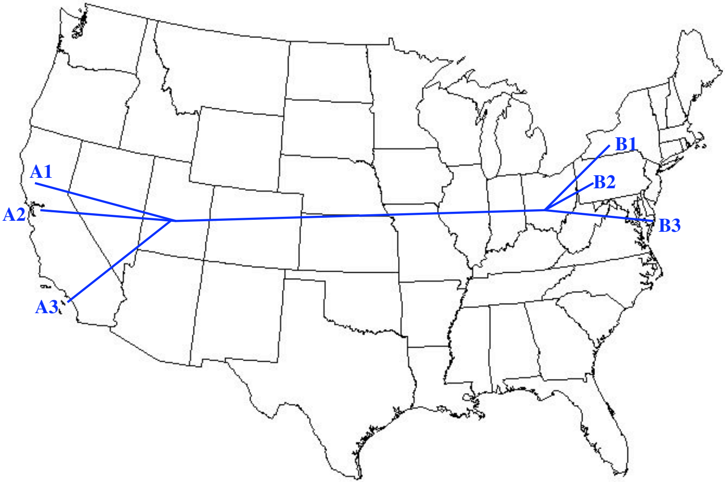
Network Tomography as Linear Algebra

- Example path properties: latency, loss rate, jitter
- Relate path properties y_{ij} to link property x_l:

$$y_{ij} = \sum_{l \in \text{path}} x_l = \sum_l g_{ijl} x_l$$

  where g_{ijl} indicates if link l is used on path $i \to j$.
- Write in matrix form (one path per row): $y = Gx$

Path Dependencies and Matrix Structure



$$(A_i \rightarrow B_i) + (A_1 \rightarrow B_1) = (A_1 \rightarrow B_i) + (A_i \rightarrow B_1)$$

Sparsification: a Motivating Example

Gravitational potential at mass j from other masses is

$$\phi_j(x) = \sum_{i \neq j} \frac{G m_i}{|x_i - x_j|}.$$

In cluster A, don't *really* need everything about B. Just summarize.

A motivating example

Gravitational potential is a linear function of masses

$$\begin{bmatrix} \phi_A \\ \phi_B \end{bmatrix} = \begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix} \begin{bmatrix} m_A \\ m_B \end{bmatrix}.$$

In cluster A, don't *really* need everything about B. Just summarize.

That is, represent P_{AB} (and P_{BA}) compactly.

Low-rank interactions

Summarize masses in B with a few variables:

$$z_B = V_B^T m_B, \quad m_B \in \mathbb{R}^{n_B}, z_B \in \mathbb{R}^p.$$

Then contribution to potential in cluster A is $U_A z_B$. Have

$$\phi_A \approx P_{AA} m_A + U_A V_B^T m_B.$$

Do the same with potential in cluster B; get system

$$\begin{bmatrix} \phi_A \\ \phi_B \end{bmatrix} = \begin{bmatrix} P_{AA} & U_A V_B^T \\ U_B V_A^T & P_{BB} \end{bmatrix} \begin{bmatrix} m_A \\ m_B \end{bmatrix}.$$

Idea is the basis of fast *n*-body methods (e.g. fast multipole method).

Sparsification

Want to solve $Ax = b$ where $A = S + UV^T$ is sparse plus low rank.

If we knew x, we could quickly compute b:

$$
\begin{aligned}
z &= V^T x \\
b &= Sx + Uz.
\end{aligned}
$$

Use the same idea to write $Ax = b$ as a bordered system[1]:

$$
\begin{bmatrix} S & U \\ V^T & -I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.
$$

Solve this using standard sparse solver package (e.g. UMFPACK).

[1] This is Sherman-Morrison in disguise

Sparsification in gravity example

Suppose we have ϕ, want to compute m in

$$\begin{bmatrix} \phi_A \\ \phi_B \end{bmatrix} = \begin{bmatrix} P_{AA} & U_A V_B^T \\ U_B V_A^T & P_{BB} \end{bmatrix} \begin{bmatrix} m_A \\ m_B \end{bmatrix}.$$

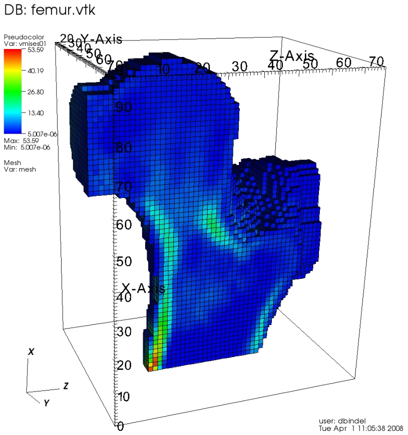
Add auxiliary variables to get

$$\begin{bmatrix} \phi_A \\ \phi_B \\ 0 \\ 0 \end{bmatrix} = \left[\begin{array}{cc|cc} P_{AA} & 0 & 0 & U_A \\ 0 & P_{BB} & U_B & 0 \\ \hline V_A^T & 0 & -I & 0 \\ 0 & V_B^T & 0 & -I \end{array}\right] \begin{bmatrix} m_A \\ m_B \\ z_A \\ z_B \end{bmatrix}.$$

Preliminary work

- Parallel sparsification routine (with Tim Mitchell)
  - User identifies low-rank blocks
  - Code factors the blocks and forms a sparse matrix as above
- Works pretty well on an example problem (charge on a capacitor)
- My goal state: Sparsification of separators for fast PDE solver

Goal state



I want a direct solver for this!

Many cheerful things

- MEMS-ish
  - Frequency-response characterization of AFM tip sharpness
  - Opto-mechanical MEMS gyros
- Network-ish
  - Fast factorization methods for path matrices
  - Algebraic properties of path matrices
  - Network topology inference from end-to-end measurements
- Rank-structured matrices
  - Frameworks for high-performance parallel implementations
  - Robust direct solvers for 3D PDE discretizations
  - Connections to domain decomposition, etc.
- Eigenstuff
  - Algorithms and error analysis for resonance problems
  - Provable bounds for 2D soft obstacle scattering poles
  - Resonances for more general open Riemannian manifolds