# Scientific Computing Group

David Bindel

Doug James

Charlie Van Loan

# Connections

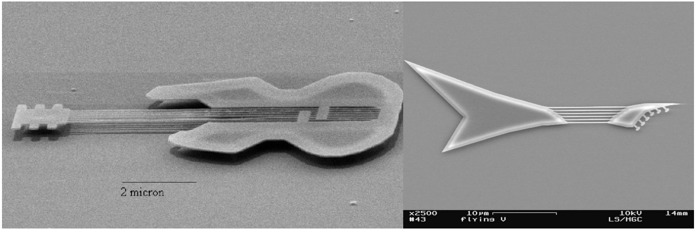
What we do:

- Matrix computations
- Fast transforms
- Model reduction
- Physical simulations
- Network modeling

Who we talk to:

- Graphics and vision
- Machine learning
- Computer systems
- Engineering
- Physical sciences

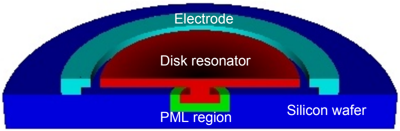
# Example: Radio-Frequency MEMS

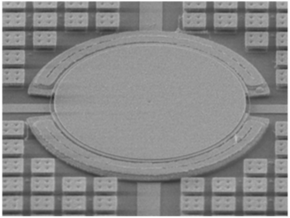



Microguitars from Cornell University (1997 and 2003)

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

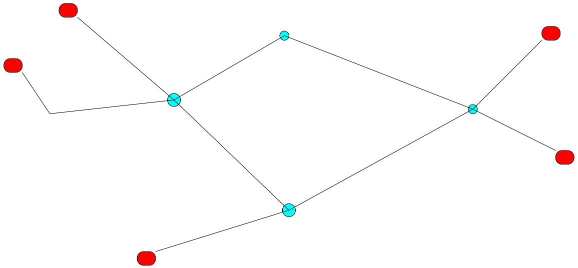
# Modeling a Ringing Disk



- 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*?

# Example: Network Tomography



Measure few paths between $n$ hosts in network to infer

- Path properties
- Link properties
- Network topology?

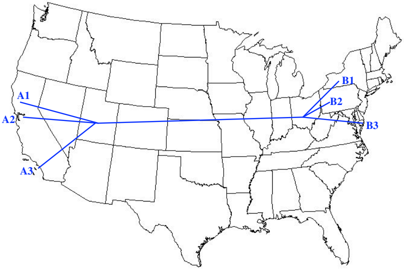
# Network Tomography as Linear Algebra

- 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 $i \rightarrow j$.
- Write in matrix form (one path per row): $y = Gx$

# Path Dependencies and Matrix Structure




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

- Many linear dependencies among paths!
- $k := \mathrm{rank}(G) \ll n^2$
- Ex: Measure only $k$ paths to infer properties for all paths

# Tensor Computations

What is a tensor? Think of it as a higher dimensional matrix.

A fourth-order tensor...

$$A = A(1:n_1, 1:n_2, 1:n_3, 1:n_4)$$

They are typically very large data objects...

$$N = n_1 \cdot n_2 \cdot n_3 \cdots n_d$$

# Why?

Make it possible for scientists to extract information from high-dimensional datasets that arise from modeling...

$$\mathcal{A}(i,j,k,\ell) \;=\; \textit{a measurement that results by setting the value of four independent variables}$$

or discretization...

$$\mathcal{A}(i,j,k,\ell) \;=\; f(w_i, x_j, y_k, z_\ell)$$

# How?

A tensor

$$\mathcal{A} = \mathcal{A}(1\!:\!4, 1\!:\!3, 1\!:\!n_3, 1\!:\!n_4)$$

can be “flattened” into a block matrix:

$$A = \begin{bmatrix} \mathcal{A}(1,1,:,:) & \mathcal{A}(1,2,:,:) & \mathcal{A}(1,3,:,:) \\ \mathcal{A}(2,1,:,:) & \mathcal{A}(2,2,:,:) & \mathcal{A}(2,3,:,:) \\ \mathcal{A}(3,1,:,:) & \mathcal{A}(3,2,:,:) & \mathcal{A}(3,3,:,:) \\ \mathcal{A}(4,1,:,:) & \mathcal{A}(4,2,:,:) & \mathcal{A}(4,3,:,:) \end{bmatrix}$$

**Methodology:** Extract information from $A$ using “classical” matrix computations. Then draw conclusions about tensor $\mathcal{A}$.

# Focus: Low Rank Approximation

**Given:** $\mathcal{A}(1:n, 1:n, 1:n, 1:n, 1:n, 1:n)$.

**Find:** $n$-by-$n$ matrices $B_1, \ldots, B_p$, $C_1, \ldots, C_p$, and $D_1, \ldots, D_p$ so that

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5, i_6) \;\approx\; \sum_{s=1}^{p} B_s(i_1, i_2) C_s(i_3, i_4) D_s(i_5, i_6)$$

Approximating an $O(n^6)$ data object with $3pn^2$ numbers.

**Vehicle:** Multilinear optimization

**Goal:** Make intractable problems tractable through approximation.