Scientific Computing Group

David Bindel

Doug James

Charlie Van Loan

Connections

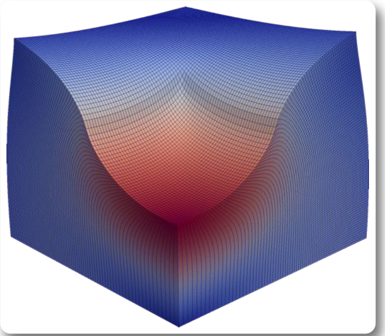
What we do:

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

Who we talk to:

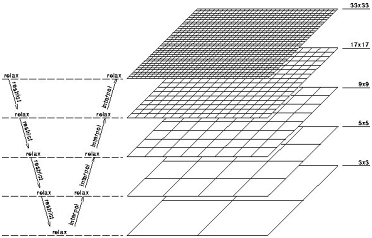
- Graphics and vision
- Machine learning
- Theory
- Computer systems
- Engineering
- Physical sciences

Example: Rank-Structured Factorization



How do we solve $Ax = b$ for big N?

Direct or iterative?

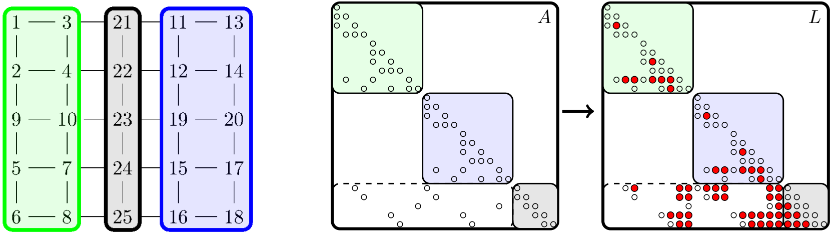


CW: Gaussian elimination scales poorly. Iterate instead!

- **Pro:** Less memory, potentially better complexity
- **Con:** Less robust, potentially worse memory patterns

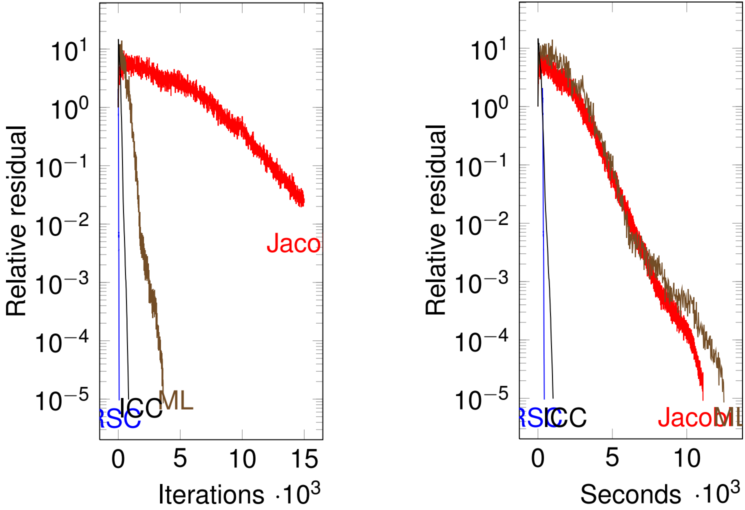
Industry codes still use (out-of-core) Cholesky!

Factorization with nested dissection

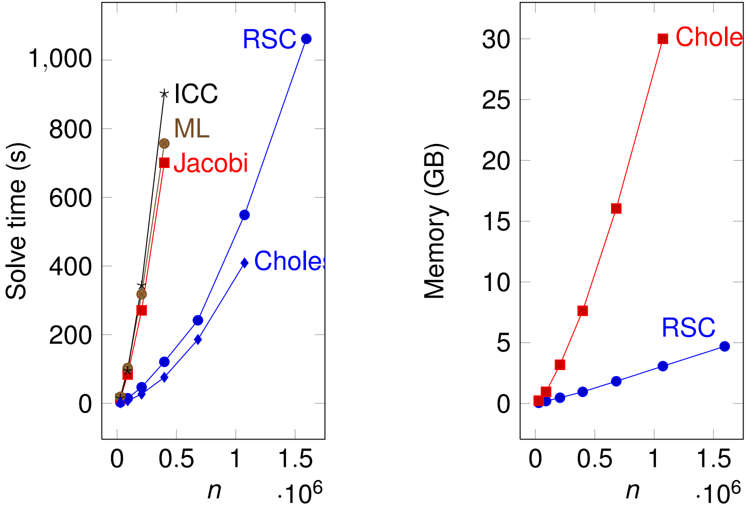


- Cholesky: $A = LL^T$, L is lower triangular.
- L may have more nonzeros than A (fill)
- Order controls fill – still a *lot* of memory in 3D
- But in physical problems, fill has structure!

Direct solves triumph



Direct solves triumph



Example: 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...

$\mathcal{A}(i,j,k,\ell)$ = *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.